



**NEAR EAST UNIVERSITY**

**Faculty of Engineering**

**Department of Computer Engineering**

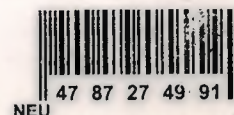
**STUDENT TRACKING SYSTEM USING VISUAL  
BASIC PROGRAMMING**

**Graduating Project  
COM 400**

**Student: Ilker DAMAR (20010760)**

**Supervisor : Assoc.Prof.Dr. Rahib ABIYEV**

**Lefkosa 2006**



## ACKNOWLEDGEMENTS

*"Firstly, I would like to thank to my supervisor Assoc.Prof.Dr Rahib Abiyev for his great advise and recommendation for finishing of this my graduating project properly also, teaching and guiding me in others lectures*

*I am greatly indepted to my family for their endless support from my starting day in my educational life until today. I will never forget the things that my sister Mrs. Gülbeyaz Türkoglu and her husband Mr. Ahmet Türkoglu did for me during my educational life, also I want to say thanks to my big brother Mr. Duran Damar and other brothers( Ali, Raşit,Muhammed Damar), my sisters(Tekin, Hüsne Damar) for supporting me as material and moral after my father's death. And of course, to forget is impossible my father(Tahsin Damar) and mother(Gürcü Damar). I dedicate my project to them.*

*Althought, I encountered many problems in writing program. Fortunately my best friend Metin Ulaş, Who helped me in realising applied visual basic programming for completing my project .I want to say a thanks to Halil Kirisci for helping me about designing visual appereance of my project. My sincere thanks to him and others my house friends*

*I thank all the staff of the faculty of engineering for giving facilities to practise, teaching and solving problem in my comlete undergraduation program*

*Finally, I promise to do my best in my life as an bachelor of science(engineer) after finishing my undergraduate program"*



## ABSTRACT

This aim of this project is to prepare a suitable student tracking program. That contains registration, all educational and financial problem of student. The program was prepared by using visual basic programming and using access database together. Visual basic application provides a complete integrated development environment(IDE). That features the same elements familiar to developers using Microsoft Visual Basic, including aProject Window, aProperties Window, and debugging tools. VBA also includes support for Microsoft Forms for creating custom dialog boxes and ActiveX controls for rapidly building user interface, integrated directly into a host application. To gain access to a remote ODBC data source. To join access is so simple. Because visual basic supports to microsoft access exactly. Access is very useful and using is very simple.

This project consist of so many forms and menus. The main menu of the program is designed for login of five different groups. Which are listed under the its own table title and authorised to reach to own form and student information through that forms. These are student, advisor, teacher, rector and secretary. An individual who is working in any of these predefined type can login to the program by using a predefined password. After logging there will be a form. Which belongs to authorised person. The authority of the user to reach, do changes and update the information in this program is limited with respect to the position according to letting users. For instance, secretary works and responsible for entering grade, has nothing to do with changing the any grades of the student till it is not approved by administration. These are simply expressing how the program was designed to use in proper and secure way. The program provides the main personal details such as name, photo, the admission date and more about students. Additionally the disciplinary situation, academic semester, which they were enrolled, courses they have taken and their payments are available in different forms of the program

To show results show the efficiency of the program of student tracking system in program of the using in other chapters



# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT.....</b>	<b>i</b>
<b>ABSTRACT.....</b>	<b>ii</b>
<b>TABLE OF CONTENTS.....</b>	<b>iii</b>
<b>LIST OF TABLES.....</b>	<b>v</b>
<b>LIST OF FIGURES.....</b>	<b>vi</b>
<b>INTRODUCTION .....</b>	<b>vii</b>
<b>CHAPTER ONE: VISUAL BASIC.....</b>	<b>1</b>
1.1.Introducing Visual Basic.....	02
1.2.Visual Basic Environment.....	02
1.3.Visual Basic for application .....	04
1.4.Step in building a Visual Basic Application.....	06
1.5.Working with controls and writing code.....	07
1.6.Managing data types.....	08
1.6.1.Numeric data types.....	08
1.6.2.Non-Numeric data types.....	08
1.6.3.DDDeclering variables.....	09
1.7.Introduction to Visual Basic Function.....	10
1.7.1.Creating functions.....	10
1.8.Declaring array.....	13
1.9.Database application.....	13
1.9.1.UsingADOcontrol.....	14
<b>CHAPTER TWO: DATABASE STRUCTURE.....</b>	<b>15</b>
2.1.Brief information.....	15
2.2.Table structure in database.....	15
2.3.Relation between table.....	19
2.4.Microsoft Access Database.....	21
2.4.1.Microsoft access database fundamentals.....	21
2.4.2.Creating table.....	22
2.5.Introdusing database.....	22
2.6.Database Keys.....	23
2.7.Working with SQL.....	24
2.7.1.Data manipulation language.....	24
2.7.2.Insert statement.....	25
2.7.3 Select statement.....	25
<b>CHAPTER THREE: FLOWCHART OF PROGRAM.....</b>	<b>26</b>
3.1.Flowchart of main menu.....	26
3.2.Flowchart of student.....	27
3.3.Flowchart of teacher.....	28
3.3.1.Flowchart of student entry from teacher(user).....	29
3.3.2. Course entry's flowchart.....	32
3.4.Flowchart of secretary.....	33
3.4.1.Adding advisor's flowchart.....	34
3.4.2.Grade entry's flowchart.....	35
3.4.3.Flowchart of term opening.....	35
3.4.4.Flowchart of disact.....	36
3.5.Flowchart of rectory Entry.....	37

<b>CHAPTER FOUR: STUDENT TRACKING SYSTEM.....</b>	<b>39</b>
4.1.Main Entrence.....	39
4.2.Teacher Form.....	39
4.2.1.Student detail(from teacher entry).....	40
4.2.2.Course register for student(from teacher entry).....	40
4.2.3.Student disact(from teacher entry).....	41
4.2.4.Searching form(from teacher entry).....	42
4.2.5.All courses with opened term.....	42
4.2.6.Course Register Updating(from teacher entry).....	43
4.3.Student Entry.....	44
4.4.Secretary Form.....	44
4.4.1.Initiating advisor(from secretary).....	45
4.4.2.Term Opening(from secretary).....	46
4.4.3.Disact entry(from secretary).....	47
4.4.4.Course opening(from secretary).....	48
4.5.Rectory Entry.....	48
4.5.1.Student Payments(from rectory).....	49
<b>CONCLUSION.....</b>	<b>51</b>
<b>REFERANCE.....</b>	<b>52</b>
<b>APPENDIX.....</b>	<b>53</b>

## LIST OF TABLES

Table 1.1: Numeric Data Types.....	08
Table 1.2: Nonnumeric Data Types.....	08
Table 2.1. Student Database Table.....	15
Table 2.2. Personal Database Table.....	16
Table 2.3. Term Database Table.....	16
Table 2.4. Advisor Database Table.....	16
Table 2.5. Teacher Database Table.....	17
Table 2.6. Department Database Table.....	17
Table 2.7. Grade Database Table.....	17
Table 2.8. Courseinfo Database Table.....	18
Table 2.9. Course Database Table.....	18
Table 2.10. Disact Database Table.....	18
Table 2.11. Payment Database Table.....	19
Table 2.12. Loan Database Table.....	19



## LIST OF FIGURES

Figure 1.1 The Visual Basic Start-up Dialog Box.....	03
Figure 1.2: The Visual Basic Enviroment.....	04
Figure1.3. Visual Basic Code Window.....	05
Figure 1.4 : The output of example 1.1.....	06
Figure1.5. Example1.4. output.....	10
Figure1.6. Examples1.5'Output.....	11
Figure 1.7. Showing MsgBox().....	12
Figure1.8. Using Ado Control.....	14
Figure 2.1. Relation Between Tables.....	20
Figure2.2. Sample Table.....	21
Figure 3.1. Main Menu Flowchart.....	26
Figure 3.2. Flowchart of Student Entry.....	27
Figure 3.3. Flowchart of Teacher Entry.....	28
Figure 3.4. Teacher Process For Student.....	31
Figure 3.5. Course Process From Teacher Entry.....	32
Figure 3.6. Secretery Process.....	33
Figure 3.7. Initiate Advisor.....	34
Figure 3.8.Enter Grade.....	35
Figure 3.9. Term Process From Secreterary Entry.....	36
Figure 3.9.Disact Process.....	37
Figure 3.9. Rectory Entry.....	38
Figure 4.1. Main Entrence.....	39
Figure 4.2. Teacher Main Form.....	39
Figure 4.3.Student Detail(From Teacher).....	40
Figure 4.5. Course Registration.....	41
Figure 4.6. Student Disact(from teacher).....	41
Figure 4.7.Student Searching Form(teacher).....	42
Figure 4.8. All Course Form(from teacher).....	42
Figure 4.8.Course Updating(from teacher).....	43
Figure 4.9. Student Form.....	44
Figure 4.10. Secretary Main Form.....	44
Figure 4.11. Grade Entry(from secretary).....	45
Figure 4.12.Initiating Advisor(from secretary).....	46
Figure 4.13.Term Opening.....	46
Figure 4.14.Student Disact(from secretary).....	47
Figure 4.15.Opening Course.....	48
Figure 4.16. New Student Register.....	49
Figure 4.17. Student Payments.....	50

## INTRODUCTION

A student tracking program, which includes registration, all financial and educational problems of students is important problem for all education establishment. In the project, it was aimed to write a program considering these problems, that we encountered until today in our life, which was elapsed in education institute. The main structure of the program was designed to apply to the any process in all facilities and not only the university. The program is user friendly and very simply adaptable to other education institute, also can be suitable other companies with simple changes. Which is worked like a education establishment. Using the enormous of visual basic programming gives the chance to update this code in future due to academic needs.

In first chapter the visual basic and its applications are described. Visual basic for applications delivers a competitive advantage for ISV seeking to provide full customization and integration capabilities to customer. The enterprise edition allows professional to create robust distributed applications in a team settings. It includes all the features of the professional edition, back office tools such as SQL server, visual source safe and more.

The next chapters begins with the access and SQL database. Advantages of using access , it provides exactly the same options for the problems you write as it does for the problems you selected from a database. Secondly, the process of writing or selecting problems is almost independent of page layout decisions. Also you can see more details about access and SQL when you see, question in your mind will be destroyed. You will understand that, it is useful and simple. The next is about flowchart of the programs. Its aim is to provide facilities to you. When you are using the programs, also understanding using of program effectively. The last one is about using program. When you see remarkable this information in that chapter. You will see using of program and learnt to use it.

Consequently, the age is scientific. And software is very popular in nowadays. It provide gain speed in our life, so to work a computer program will make easy our life and it gained speed our life.



# CHAPTER ONE: VISUAL BASIC

## 1.1. Introducing Visual Basic

**VISUAL BASIC** is a high level programming language evolved from the earlier DOS version called BASIC. BASIC means **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. It is a fairly easy programming language to learn. The codes look a bit like English Language. Different software companies produced different version of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA and so on.

Firstly, I want to say to you about Visual Basic on the screen. When you apply to it, you will see, VB menu and tools, form, toolbox, properties window, project explorer, form layout and source code window. You can see below clearly.

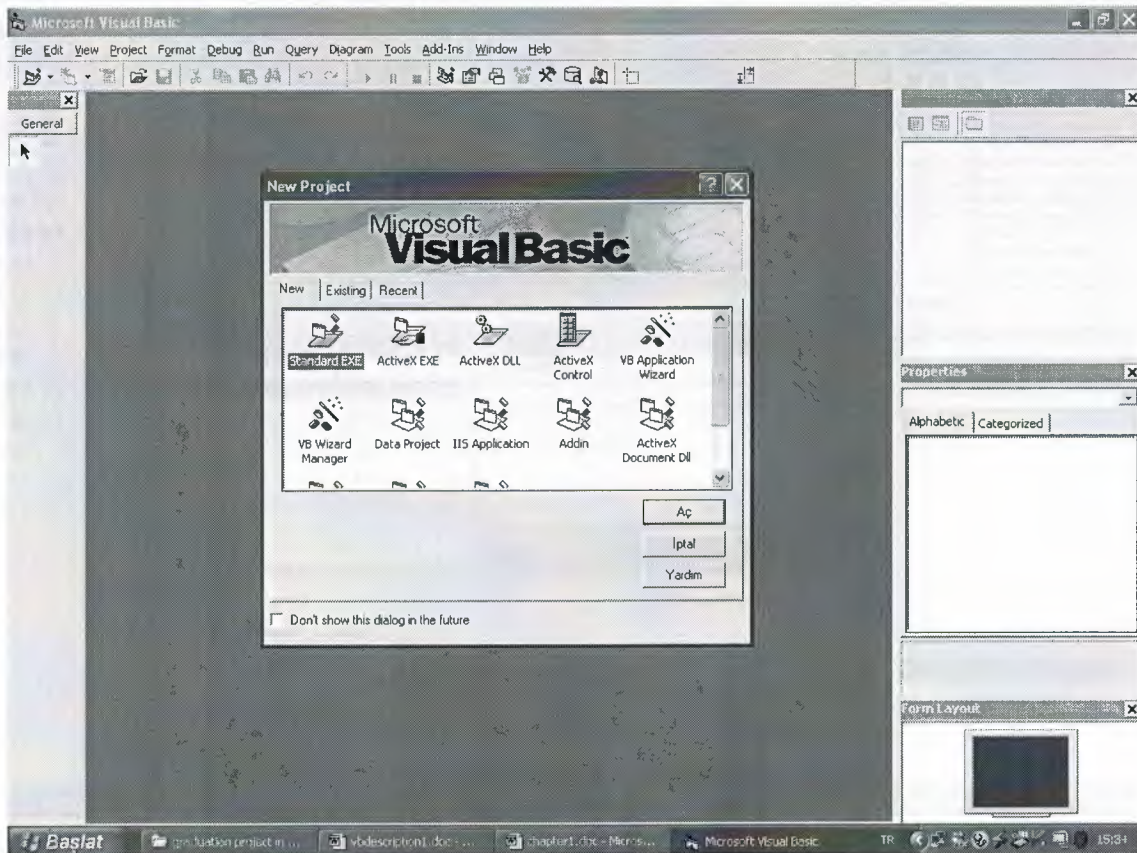
Nowadays, software is very popular. Many programmers do computer programming. If you are a computer programmer, you can consider visual basic as your language of choice for learning how to develop computer programs. Of course there are other programming language is popular. But, Visual Basic is simple as possible. Because, to learning visual basic is so easy, also to do visual interface, has involved into a very powerful development tools and fun to use. But programming in Visual Basic is a challenging. That's basic reason is easyness of Visual Basic. Everybody can do program with visual basic easily. All that facilities shows that, someone can learn Visual Basic itself and train it.

Learning Visual Basic is based on only practise. As I said before, it is visual. Directly, you are in an interaction with your graphical user interface. You must not hesitate to try your simple programs, your imagination, try it in Visual Basic programming. As you know, best way to learn Visual Basic is by examples. Also you can find information to learn about visual basic on the internet and can use a best book. Look from there and try what you learn.

Let me remind to you one more things about best programmer. As you estimate, which is to enjoy programming, just be relax and program for fun and treat programming as a hobby. You will be amazed at the progress you make and success.

## 1.2 The Visual Basic Environment

On start up, Visual Basic 6.0 will display the following dialog box as shown in figure 1.1. You can choose to start a new project, open an existing project or select a list of recently opened programs. A project is a collection of files that make up your application. There are various types of applications we could create, however, we shall concentrate on creating Standard EXE programs (EXE means executable program). Now, click on the Standard EXE icon to go into the actual VB programming environment.



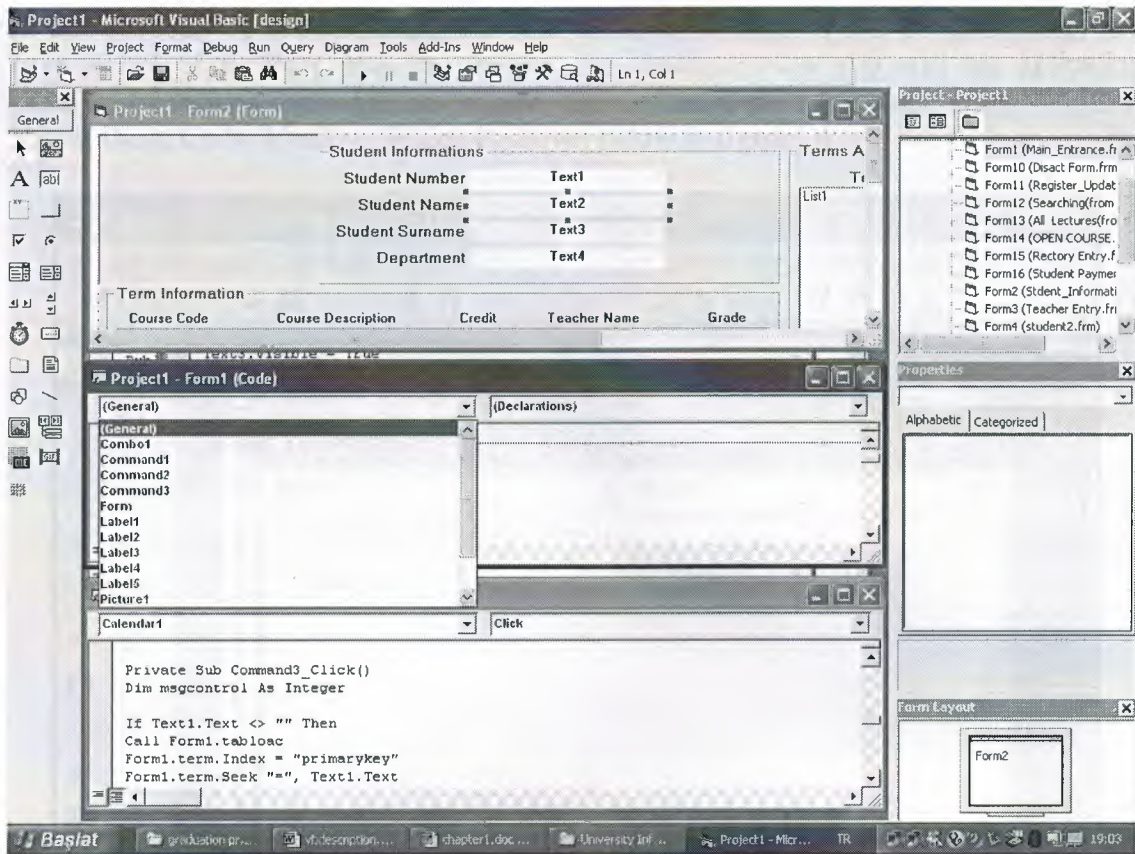
**Figure 1.1 The Visual Basic Start-up Dialog Box**

In figure 1.2, the Visual Basic Environment consists of the:

- A Blank Form for you to design your application's interface.
- The Project window which displays the files that are created in your application.
- The Properties window which displays the properties of various controls and objects that are created in your applications.

It also includes a Toolbox that consists of all the controls essential for developing a VB Application. Controls are tools such as text box, command button, label, combo box, picture box, image box, timer and other objects that can be dragged and drawn on a form to perform certain tasks according to the events associated with them. Additional objects can be added by clicking on the project item on the menu and click on components on the drop-down list, then select those controls you need to use in your program





**Figure 1.2: The Visual Basic Enviroment**

### 1.3. Visual Basic For Application

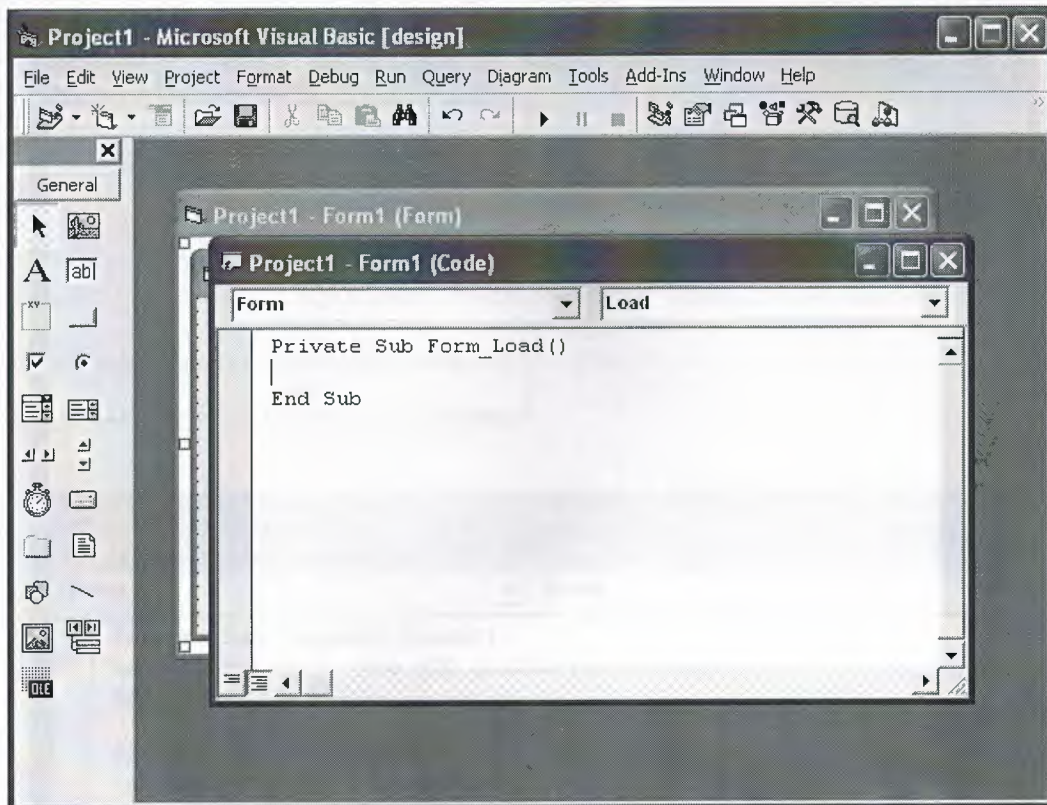
Visual Basic for application provides a complete integrated development environment. It is a powerful development technology for rapidly customizing rich client desktop packaged application and integrating them with existing data and system. VBA offers a sophisticated set of programming tools based on the Microsoft Visual Basic Development system, the worlds most popular rapid applicationdevelopment system, which developpers can use to harness the power of packaged applications.

As mentioned before, Microsoft Visual Basic includes many vindow to fasilitate programmers for their application. Which are VB menu and tools, form, toolbox, properties window, project explorer, form layout, debugging tools, and source code window. VBA also include supporting for Microsoft Form, for creating custom dialog boxes, and ActiveX controls for quickly building user interfaces

Software programs, which is include VBA, it is called customizable applications, that can be tailored to fit specific business need. This class of application enables developpers to rapidly built solution that require less and user trainig. For MIS and business managers, customization means that solution can be developped quickly and deployed easily with minimal maintenance. In an industry familiar with two years backlogs for new applications and high end user training cost. This solution provide a tremen dous business benefit in terms of retern on investment and timelines.



In this section, we are not going into the technical aspects of VB programming; just have a feel of it. Now, you can try out the examples. Now you will see application example by writing code. Of course, you can examine by writing code in to Microsoft Visual Basic Code Window. Which is shown below in figure1.4.



**Figure1.3. Visual Basic Code Window**

Now you can drop onto form command button, list box, text box, option....etc and select an object and procedure then can write code whatever you want. You can see example below.

#### **Example 1.1.**

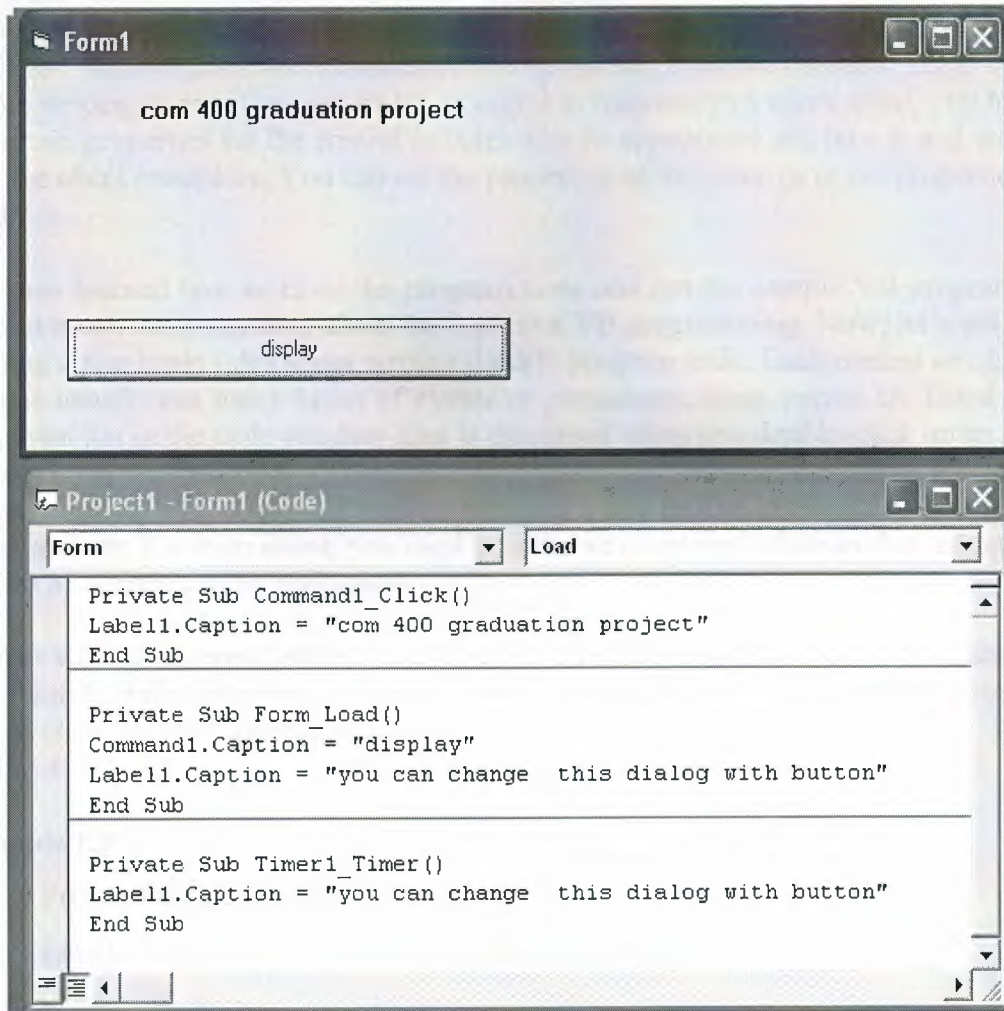
```
Private Sub Command1_Click()
    Label1.Caption = "Com 400 graduation project"
End Sub

Private Sub Form_Load()
    Command1.Caption = "display"
    Label1.Caption = "you can change this dialog with command button"
End Sub
```

```

Private Sub Timer1_Timer()
Label1.Caption = "you can change this dialog with command button"
End Sub

```



**Figure 1.4 : The output of example 1.1.**

## 1.4. Steps in Building a Visual Basic Application

Generally, there are three basic steps in building a VB application. The steps are as follows:

- Step 1 : Design the interface
- Step 2 : Set Properties of the controls (Objects)
- Step 3 : Write the events' procedures



When you will write program. If you can follow that steps whichs are benefit for you. You can improve your project by using other opportunities. Lets talk about other opportunities. Which are you can design your project witout using code. You can change your objects on to your form from properties window, tools, menu editor, format..etc.

## 1.5. Working With Controls And Writing Code

Before writing an event procedure for a control to response to a user's input, you have to set certain properties for the control to determine its appearance and how it will work with the event procedure. You can set the properties of the controls in the properties windows.

you have learned how to enter the program code and run the sample VB programs but without much understanding about the logics of VB programming. Now, let's get down learning a few basic rules about writing the VB program code. Each control or object in VB can usually run many kinds of events or procedures; these events are listed in the dropdown list in the code window that is displayed when you double-click on an object and click on the procedures' box(refer to Figure 2.3). Among the events are loading a form, clicking of a command button, pressing a key on the keyboard or dragging an object and etc. For each event, you need to write an event procedure so that an action or a series of actions can be performed.

To start writing an event procedure, you need to double-click an object. For example, if you want to write an event procedure when a user clicks a command button, you double-click on the command button and an event procedure will appear like in figure2.4

### Example 1.2.

```
Private Sub Command1_Click()  
    Label1.Caption = "Com 400 graduation project"  
End Sub
```

Syntaxes that do not involve setting of properties are also English-like, some of the commands are **Print**, **If...Then....Else....End If**, **For...Next**, **Select Case.....End Select**, **End** and **Exit Sub**. For example, **Print " Visual Basic"** is to display the text Visual Basic on screen and **End** is to end the program. Other commands will be explained in details in the coming lessons.

Program codes that involve calculations is very easy to write, you need to write them almost liket what you do in mathematics. However, in order to write an event procedure that involves calculations, you need to know the basic arithmetic operators in VB as they are not exactly the same as the normal operators we use, except for + and - . For multiplication, we use \*, for division we use /, for raising a number x to the power of n, we use **x ^n** and for square root, we use **Sqr(x)**. More advanced mathematical functions such as **Sin**, **Cos**, **Tan**, **Log** and etc. There are also two important functions that are related to arithmetic operations, i.e. the functions **Val** and **Str\$** where **Val** is to convert text entered into a textbox to numerical value and **Str\$** is to display a numerical value in a textbox as a string (text). While the function **Str\$** is as important as VB can display a



numeric values as string implicitly, failure to use Val will results in wrong calculation. Let's examine Example1.1.

## 1.6. Managing Data Types

There are many types of data we come across in our daily life. For example, we need to handle data such as names, addresses, money, date, stock quotes, statistics and etc everyday. Similarly in Visual Basic, we are also going to deal with these kinds of data. However, to be more systematic, VB divides data into different types.

### 1.6.1 Numeric Data

Numeric data are data that consists of numbers, which can be computed mathematically with various standard operators such as add, minus, multiply, divide and so on. In Visual Basic, the numeric data are divided into 7 types, they are summarized in Table1.1

Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values 1.401298E-45 to 3.402823E+38 for positive values.
Double	8 bytes	-1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.

**Table 1.1: Numeric Data Types**

### 1.6.2 Non-numeric Data Types

The nonnumeric data types are summarized in Table1.2

Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False

**Table 1.2: Nonnumeric Data Types**

### 1.6.3. Declaring Variables

In Visual Basic, one needs to declare the variables before using them by assigning names and data types. They are normally declared in the general section of the codes' windows using the **Dim** statement.

The format is as follows:

Dim variableName as DataType

### **Example 1.3**

```
Dim password As String
Dim yourName As String
Dim firstnum As Integer
Dim secondnum As Integer
Dim total As Integer
Dim doDate As Date
```

You may also combine them in one line, separating each variable with a comma, as follows: Dim password As String, yourName As String, firstnum As Integer,.....

### **If.....Then.....Else Statements with Operators**

To effectively control the VB program flow, we shall use If...Then...Else statement together with the conditional operators and logical operators.

The general format for the if...then...else statement is

**If** conditions **Then**

VB expressions

**Else**

VB expressions

**End If**

### **Select Case**

If you have a lot of conditional statements, using If..Then..Else could be very messy. For multiple conditional statements, it is better to use Select Case. The format is :

#### **Select Case expression**

Case value1

Block of one or more VB statements

Case value2

Block of one or more VB Statements

End Select

### **Looping**

Visual Basic allows a procedure to be repeated as many times as long as the processor could support. This is generally called looping.

#### **1. Do**

Block of one or more VB statements

**Loop While condition**



## 2. Do

Block of one or more VB statements

**Loop Until condition**

## 3. For....Next Loop

For counter=startNumber to endNumber (Step increment)

One or more VB statements

**Next**

## 1.7. Introduction to VB Functions

### 1.7.1 Creating Functions

The general format of a function is as follows:

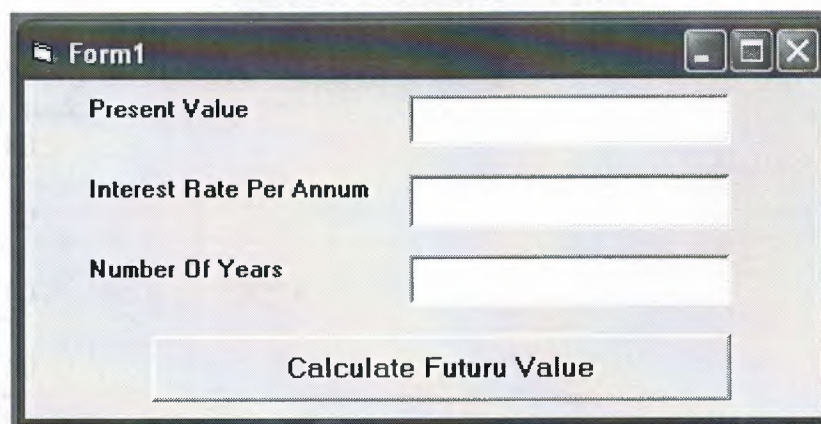
Public Function **functionName** (Arg As dataType,.....) As dataType  
or

Private Function **functionName** (Arg As dataType,.....) As dataType

\*Public indicates that the function is applicable to the whole program and  
Private indicates that the function is only applicable to a certain module or procedure.

### Example 1.4

In this example, a user can calculate future value of a certain amount of money he has today based on the interest rate and the number of years from now supposing he will invest this amount of money somewhere). The calculation is based on the compound interest rate.

A screenshot of a Visual Basic form titled 'Form1'. The form has a light blue background and a dark border. It contains three input fields stacked vertically, each with a label to its left: 'Present Value', 'Interest Rate Per Annum', and 'Number Of Years'. Below these fields is a button labeled 'Calculate Futuru Value' (note the typo 'Futuru'). The form has standard Windows window controls (minimize, maximize, close) in the top right corner.

**Figure1.5. Example1.4. output**

```
Public Function FV(PV As Variant, i As Variant, n As Variant) As Variant
'Formula to calculate Future Value(FV)
'PV denotes Present Value
FV = PV * (1 + i / 100) ^ n
End Function
```



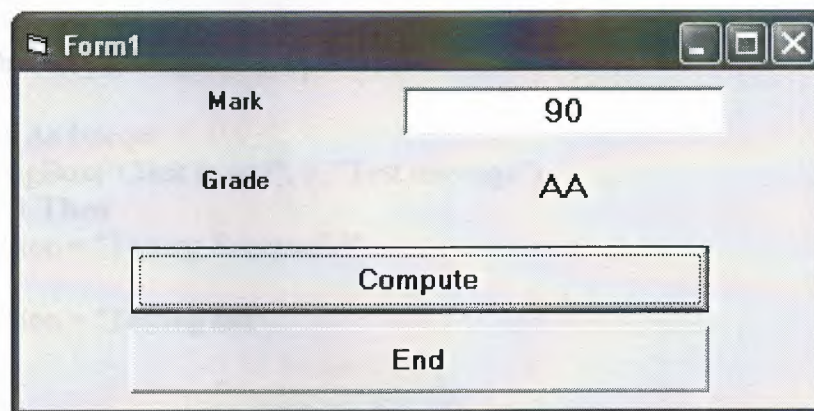
```

Private Sub compute_Click()
'This procedure will calculate Future Value
Dim FutureVal As Variant
Dim PresentVal As Variant
Dim interest As Variant
Dim period As Variant
PresentVal = PV.Text
interest = rate.Text
period = years.Text
FutureVal = FV(PresentVal, interest, period)
MsgBox ("The Future Value is " & FutureVal)
End Sub

```

### Example1.5.

The following program will automatically compute examination grades based on the marks that a student obtained.



**Figure1.6. Examples1.5' output**

```

Public Function grade(mark As Variant) As String
Select Case mark
Case Is >= 80
grade = "A"
Case Is >= 70
grade = "B"
Case Is >= 60
grade = "C"
Case Is >= 50
grade = "D"
Case Is >= 40
grade = "E"
Case Else
grade = "F"
End Select
End Function

```

```
Private Sub compute_Click()
grading.Caption = grade(mark)
End Sub
```

```
Private Sub End_Click()
End
End Sub
```

I want to remind you something about functions in Visual Basic. Also message boxes are functions inside of Visual Basic. When you write **msgbox()** programs automatically call functions from inside of Visual Basic. I will explain little with by example. In this explanation, we are going to learn two very basic but useful internal functions, i.e. the **MsgBox()** and **InputBox()** functions.

The objective of **MsgBox** is to produce a pop-up message box and prompt the user to click on a command button before he /she can continue. This message box format is as follows:

#### Example 1.6.

```
Private Sub Test_Click()
```

```
Dim testmsg As Integer
testmsg = MsgBox("Click to test", 1, "Test message")
If testmsg = 1 Then
Display.Caption = "Testing Successful"
Else
Display.Caption = "Testing fail"
End If
```

```
End Sub
```

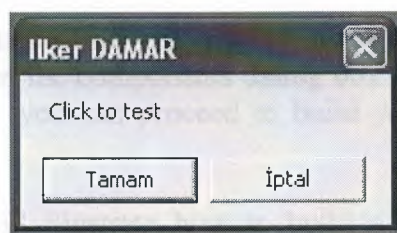


Figure 1.7. Showing **Msgbox()**

### 1.8.Declaring Array

We could use **Public** or **Dim** statement to declare an array just as the way we declare a single variable. The **Public** statement declares an array that can be used throughout an



application while the Dim statement declare an array that could be used only in a local procedure. The general format to declare an array is as follow:

**Dim arrayName(subs) as dataType**

where subs indicates the last subscript in the array.

### **Example 13.1**

**Dim FindName(10) as String**

## **1.9.Database Application**

Finally, I want to put end point in that part, with telling about database application. Of course, All topics is not restricted what I emphasize in that part.

Visual basic allows us to manage databases created with different database program such as MS Access, Dbase, Paradox and etc. In this lesson, we are not dealing with how to create database files but we will see how we can access database files in the VB environment

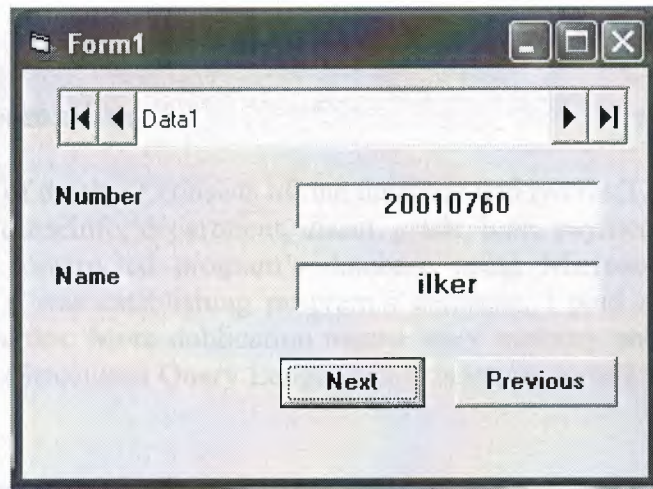
### **1.9.1. Using ADO Control**

We have learned to build VB database applications using data control. However, data control is not a very flexible tool as it could work only with limited kinds of data and must work strictly in the Visual Basic environment. To overcome these limitations, we can use a much more powerful data control in VB known as ADO control. ADO stands for ActiveX data objects. As ADO is ActiveX-based, it could work in different platforms (different computer systems) and different programming languages. Besides, it could access many different kinds of data such as data displayed in the Internet browsers, email text and even graphics other than the usual relational and non relational database information.

To be able to use ADO data control, you need to insert it into the toolbox. To do this, simply press Ctrl+T to open the components dialog box and select Microsoft ActiveX Data Control 6. After this, you can proceed to build your ADO-based VB database applications.

The following example will illustrate how to build a relatively powerful database application using ADO data control. First of all, name the new form as Student and change its caption to Book Tiles- ADO Application. Secondly, insert the ADO data control and name it as Course and change its caption to book. Next, insert the necessary labels, text boxes and command buttons. The runtime interface of this program is shown in the diagram below, it allows adding and deletion as well as updating and browsing of data.





**Figure1.8. Using ADO Control**

Consequently, As mentioned before. All things are not limeted that, 'what I told in that project'. As you gues. Many things can be done using Visual Basic. You can animate, file manage, advance database managing...etc. I summarized here, whichs was used in that project. I hope that, What I told this part. Which will give opinion to you 'how i did that project'.

## CHAPTER TWO: DATABASE STRUCTURE

### 2.1. Brief Information

General structure of database consists of one database and twelve(12) tables. Which are advisor, course, courseinfo, department, disact, grade, loan, payment, personel, student, teacher, term. I constructed program's database using Microsoft Acces Database. Aspecially when I was establishing program's database, i paid my attention do not dublicate in my tables. More dublication means more memory and less speed. Also I studied with SQL(Structured Query Language). It is much usefull to combine program with database.

### 2.2. Tables Structure In Database

As I mentioned before I used twelve table with a Access Database. All tables are given below.

Student database table includes information about student

STUDENT.DB			
Field Name	Type	Size	Key
Stno	String	8	*
Stname	String	20	
Stsurname	String	20	
Stfathername	String	20	
Stmothername	String	20	
Stplaceofbirth	String	15	
Stdateofbirth	Date	10	
Country	String	15	
Nationality	String	10	
Sthighschool	String	15	
Ststartdate	Date	10	
Stfinishdate	Date	10	
Staddress	String	70	
Sttel	String	18	
Stmail	String	25	
Stgender	Boolean	1	
Deptid	String	3	ForeingKey
Stusername	String	10	
Stpassword	String	16	

**Table 2.1. Student Database Table**



PERSONAL.DB			
Field Name	Type	Size	Key
Pid	String	8	*
Pplace	String	20	
Pname	String	20	
Psname	String	20	
Paddress	String	70	
Ptel	String	18	
Pmail	String	25	
Pcountry	String	15	
Ptype	String	10	
Pstartdate	Date	10	
Penddate	Date	10	
Pgender	Boolean	1	
Pbirthdate	Date	10	
Pbirthplace	String	15	
Pnationality	String	10	
Pusername	String	10	
Ppassword	String	16	

**Table 2.2. Personal Database Table**

TERM.DB			
Field Name	Type	Size	Key
Termid	String	3	*
Termname	String	11	
Termstart	Date	10	
Termend	Date	10	

**Table 2.3. Term Database Table**

ADVISOR.DB			
Field Name	Type	Size	Key
Tid	String	8	*
Stno	String	8	*
Termid	String	3	*

**Table 2.4. Advisor Database Table**

TEACHER.DB			
Field Name	Type	Size	Key
Tid	String	8	*
Tname	String	20	
Tsurname	String	20	
Taddress	String	70	
Ttell	String	18	
Tmail	String	25	
Tcountry	String	15	
Tdegree	String	10	
Deptid	String	3	ForeingKey
Tstartdate	Date	10	
Tenddate	Date	10	
Tbirthdate	Date	10	
Tbirthplace	String	15	
Tnationality	String	10	
Tgender	Boolean	1	
Texpert	String	10	
Tusername	String	20	
Tpassword	String	16	

**Table 2.5. Teacher Database Table**

DEPARTMENT.DB			
Field Name	Type	Size	Key
Deptid	String	3	*
Deptname	String	25	
Faculty	String	20	
Tenumber	String	2	
NTEnumber	String	2	

**Table 2.6. Department Database Table**

GRADE.DB			
Field Name	Type	Size	Key
Courseid	String	8	*
Stno	String	8	*
Grade	String	2	
Termid	String	3	*
Tid	String	8	

**Table 2.7. Grade Database Table**



COURSEINFO.DB			
Field Name	Type	Size	Key
Courseid	String	13	*
Cousecode	String	10	
Deptid	String	3	ForeingKey
Coursename	String	40	
Credit	Number	2	
Coursetype	String	4	
Prereq	String	20	
Termno	String	2	
Teacherid	String	10	ForeingKey

**Table 2.8. Courseinfo Database Table**

COURSE.DB			
Field Name	Type	Size	Key
Couseid	String	13	
Teacherid	String	8	*
Termid	String	3	*

**Table 2.9. Course Database Table**

DISACT.DB			
Field Name	Type	Size	Key
Disactid	String	8	*
Stno	String	8	ForeingKey
Disactinfo	String	40	
Disactdate	Date	10	
Termid	String	3	ForeingKey
Disactresult	String	70	

**Table 2.10. Disact Database Table**

PAYMENT.DB			
Field Name	Type	Size	Key
Payid	String	10	*
Paybillno	String	10	
Paydate	Date	10	
Payamount	Number	4	
Stno	String	8	ForeingKey
Paytype	String	2	
Termid	String	3	ForeingKey

**Table 2.11. Payment Database Table**

LOAN.DB			
Field Name	Type	Size	Key
Loanid	String	10	*
Stno	String	8	ForeingKey
Loaninfo	String	20	
Termid	String	3	ForeingKey
LastOfLoan	Date	10	
Taxrate	Number	2	
Loanamount	Number	5	

**Table 2.12. Loan Database Table**

### 2.3. Relation Between Tables

All tables was created in Microsoft Access Database. All relations was done from relation menu. There are twelve(12) tables. As you know, that tables includes many key to make relation easyly. Which keys are primary key, foreing key, komposit key. Some times,to do primary key from one column is not possible.that's why composite key can be selected. All keys are used to make relation between tables. Relations are between that keys. To see relation is little hard, but Anderstand is very easy. Of course you can also see clearly in the figure below.



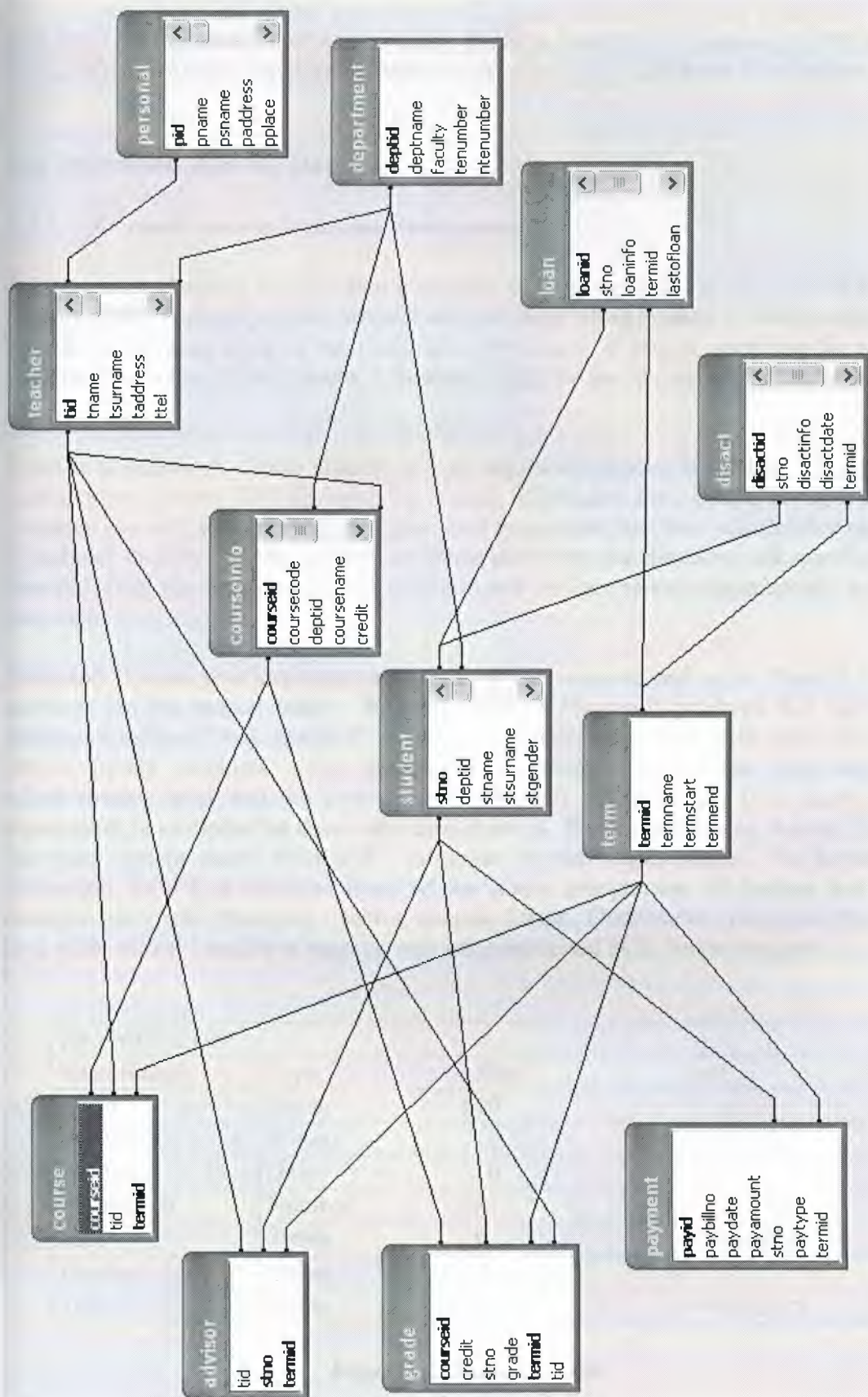


Figure 2.1. Relation Between Tables

As you saw above, all tables was done with using Microsoft Access Database. I will give brief information about Access and SQL(Structured Query Language). Of course, Using SQL in Access Database is possible. Also you will find more information about SQL.

## 2.4. Microsoft Access Database

### 2.4.1. Microsoft Access Database Fundamentals

Are you overwhelmed by the large quantities of data that need to be tracked in your organization? Perhaps you're currently using a paper filing system, text documents or a spreadsheet to keep track of your critical information. If you're searching for a more flexible data management system, a database might be just the salvation you're looking for.

What is a database? Quite simply, it's an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

Microsoft Access provides users with one of the simplest and most flexible DBMS solutions on the market today. Regular users of Microsoft products will enjoy the familiar Windows "look and feel" as well as the tight integration with other Microsoft Office family products. An abundance of wizards lessen the complexity of administrative tasks and the ever-present Microsoft Office Helper (you know... the paper clip!) is available for those who care to use it. Before purchasing Access, be sure that your system meets Microsoft's minimum system requirements. To further our discussion, let's first examine three of the major components of Access that most database users will encounter – tables, queries, forms. Once we've completed that we'll look at the added benefits of reports, web integration and SQL Server integration.

:

PAYMENT.DB			
Field Name	Type	Size	Key
Payid	String	10	*
Paybillno	String	10	
Paydate	Date	10	
Payamount	Number	4	
Stno	String	8	ForeingKey
Paytype	String	2	
Termid	String	3	ForeingKey

**Figure2.2. Sample Table**

The table above contains the employee information for our organization characteristics like name, date of birth and title. Examine the construction of the table and you'll find that each column of the table corresponds to a specific employee characteristic (or



**attribute** in database terms). Each row corresponds to one particular employee and contains his or her information. That's all there is to it! If it helps, think of each one of these tables as a spreadsheet-style listing of information.

In the previous section, we learned how tables allow us to create the framework for storing information in a database. Obviously, a database that only stored information would be useless -- we need methods to retrieve information as well. If you simply want to recall the information stored in a table, Microsoft Access allows you to open the table and scroll through the records contained within it. However, the real power of a database lies in its capabilities to answer more complex requests, or **queries**. Access queries provide the capability to combine data from multiple tables and place specific conditions on the data retrieved.

#### **2.4.2. Creating Table**

Many techniques allow you to create a database, the fastest of which consists of using one of the provided examples. Microsoft Access 97 shipped with 22 sample databases while Microsoft Access 2000 ships with 10. Furthermore, the 97 version allowed to provide sample data into the database. This is not available with the 2000 release. The databases that ship with Microsoft Access can help you in two main ways: they provide a fast means of creating a database and you can learn from their structure.

To create a database using one of the samples, there is a little detail to follow depending on whether you had launched the program already or not. If Microsoft Access is not running, you can start it. When the first dialog box comes up, you can click the second radio button: Access Database Wizard, Pages

The New dialog box displays two property pages labeled General and Databases. If you want to create a database based on one of the samples, you can click the Databases property page. A list of the sample databases appears. You can then choose one and click OK.

When creating a database using one of the samples, depending on the sample you selected, the Database Wizard will display a few objects and suggest some fields for your database. Some fields are already associated with the objects and some other fields can be added. You can examine them, then add some fields you think are important for your database. You will also have the option of selecting a design layout. Some of the sample databases have been configured to require information about the company you are creating the database for.

### **2.5. Introducing Database**

Databases are designed to offer an organized mechanism for storing, managing and retrieving information. They do so through the use of tables. If you're familiar with spreadsheets like Microsoft Excel, you're probably already accustomed to storing data in tabular form. It's not much of a stretch to make the leap from spreadsheets to databases.

Just like Excel tables, database tables consist of columns and rows. Each column contains a different type of attribute and each row corresponds to a single record. For example, imagine that we were building a database table that contained names and

telephone numbers. We'd probably set up columns named "FirstName", "LastName" and "TelephoneNumber." Then we'd simply start adding rows underneath those columns that contained the data we're planning to store.

At this point, you're probably asking yourself an obvious question – if a database is so much like a spreadsheet, why can't I just use a spreadsheet? Databases are actually much more powerful than spreadsheets in the way you're able to manipulate data. Here are just a few of the actions that you can perform on a database that would be difficult if not impossible to perform on a spreadsheet:

- Retrieve all records that match certain criteria

- Update records in bulk

- Cross-reference records in different tables

- Perform complex aggregate calculations

As we walk through this tutorial, you'll learn how you can use databases to achieve each of these objectives. Page 2 of this lesson provides you with an overview of how database keys can be used to uniquely identify records and form relationships between tables. Page 3 describes how the Structured Query Language allows you to interact with your database. On page 4, we examine the different types of databases available on the market today.

## 2.6. Database Keys

On the previous page of this article, you learned how databases use tables to organize data. As you probably recall, each table consists of a number of rows, each of which corresponds to a single database record. So, how do databases keep all of these records straight? It's through the use of keys.

The first type of key we'll discuss is the **primary key**. Every database table should have one or more columns designated as the primary key. The value this key holds should be unique for each record in the database. For example, assume we have a table called Employees that contains personnel information for every employee in our firm. We'd need to select an appropriate primary key that would uniquely identify each employee. Your first thought might be to use the employee's name.

This wouldn't work out very well because it's conceivable that you'd hire two employees with the same name. A better choice might be to use a unique employee ID number that you assign to each employee when they're hired. Some organizations choose to use Social Security Numbers (or similar government identifiers) for this task because each employee already has one and they're guaranteed to be unique. However, the use of Social Security Numbers for this purpose is highly controversial due to privacy concerns.

Once you decide upon a primary key and inform the database of this decision, it will enforce the uniqueness of the key. If you try to insert a record into a table with a primary key that duplicates an existing record, the insert will fail.

Most databases are also capable of generating their own primary keys. Microsoft Access, for example, may be configured to use the AutoNumber data type to assign a



unique ID to each record in the table. While effective, this is a bad design practice because it leaves you with a meaningless value in each record in the table. Why not use that space to store something useful?

The other type of key that we'll discuss in this course is the **foreign key**. These keys are used to create relationships between tables. Natural relationships exist between tables in most database structures. Returning to our employees database, let's imagine that we wanted to add a table containing departmental information to the database. This new table might be called Departments and would contain a large amount of information about the department as a whole. We'd also want to include information about the employees in the department, but it would be redundant to have the same information in two tables (Employees and Departments). Instead, we can create a **relationship** between the two tables.

Let's assume that the Departments table uses the Department Name column as the primary key. To create a relationship between the two tables, we add a new column to the Employees table called Department. We then fill in the name of the department to which each employee belongs. We also inform the database that the Department column in the Employees table is a **foreign key** that references the Departments table. The database will then enforce *referential integrity* by ensuring that all of the values in the Departments column of the Employees table have corresponding entries in the Departments table.

Note that there is no uniqueness constraint for a foreign key. We may (and most likely do!) have more than one employee belonging to a single department. Similarly, there's no requirement that an entry in the Departments table have *any* corresponding entry in the Employees table. It is possible that we'd have a department with no employees.

## 2.7. Working with SQL

SQL (pronounced "ess-que-el") stands for structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingress, etc. although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "select", "insert", "delete", "create" and "drop" can be used to accomplish almost everything that one needs to do with a database.

### 2.7.1. Data Manipulation Language

The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

### 2.7.1.1.INSERT

The INSERT command in SQL is used to add records to an existing table. Returning to the personal\_info example from the previous section, let's imagine that our HR department needs to add a new employee to their database. They could use a command similar to the one shown below:

```
INSERT INTO department  
values('c01','computer','engineering',5,6)
```

Note that there are four values specified for the record. These correspond to the table attributes in the order they were defined: first\_name, last\_name, employee\_id, and salary.

### 2.7.1.2.SELECT

The SELECT command is the most commonly used command in SQL. It allows database users to retrieve the specific information they desire from an operational database. Let's take a look at a few examples, again using the personal\_info table from our employees database.

The command shown below retrieves all of the information contained within the personal\_info table. Note that the asterisk is used as a wildcard in SQL. This literally means "Select everything from the personal\_info table."

```
SELECT *  
FROM student
```

Alternatively, users may want to limit the attributes that are retrieved from the database. For example, the Human Resources department may require a list of the last names of all employees in the company. The following SQL command would retrieve only that information:

```
SELECT stname  
FROM student WHERE stno=20010760
```



## CHAPTER THREE: FLOWCHART OF PROGRAM

### 3.1. Flowchart Of Main Menu

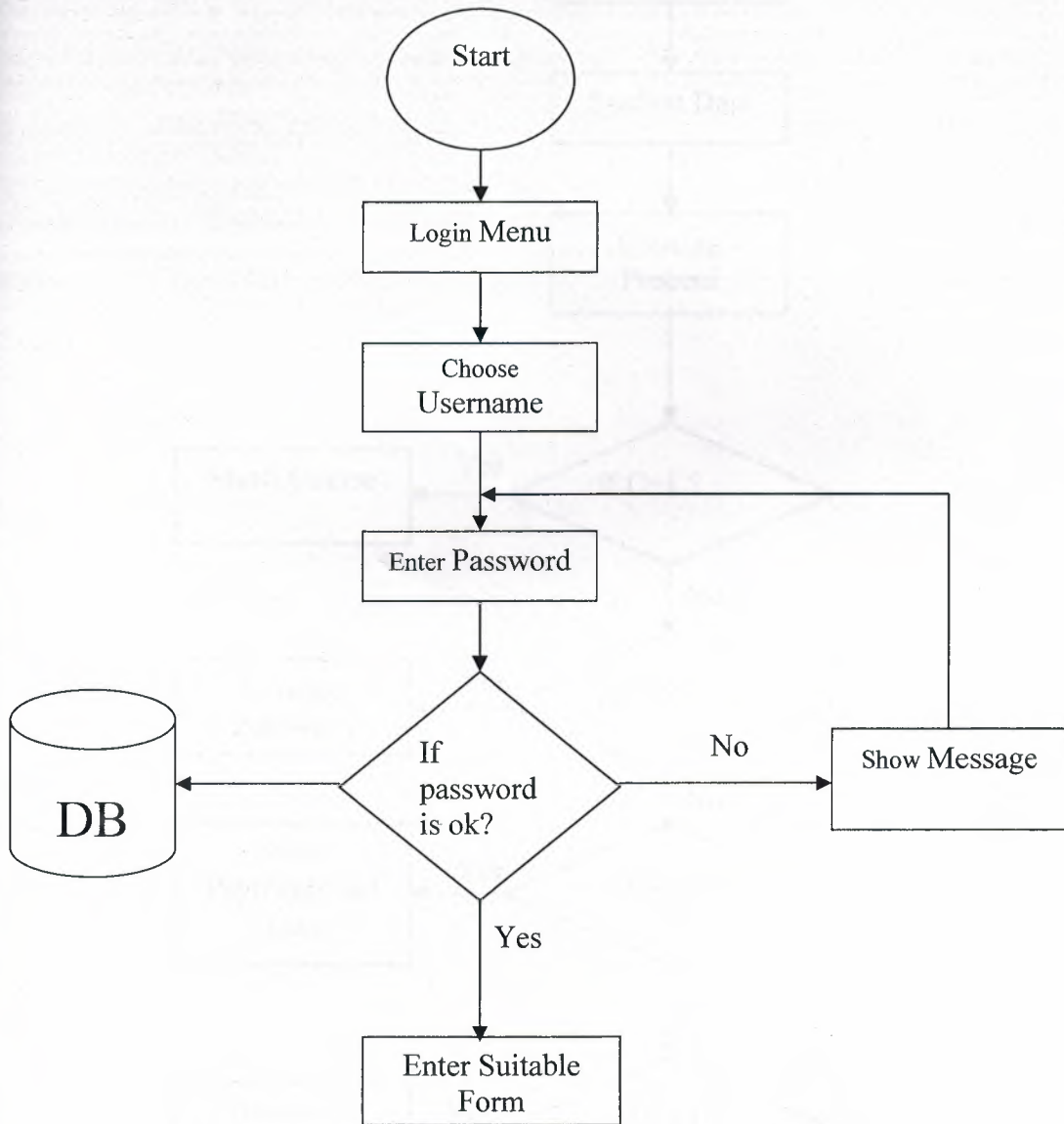


Figure 3.1. Main Menu Flowchart

### 3.2. Flowchart Of Student

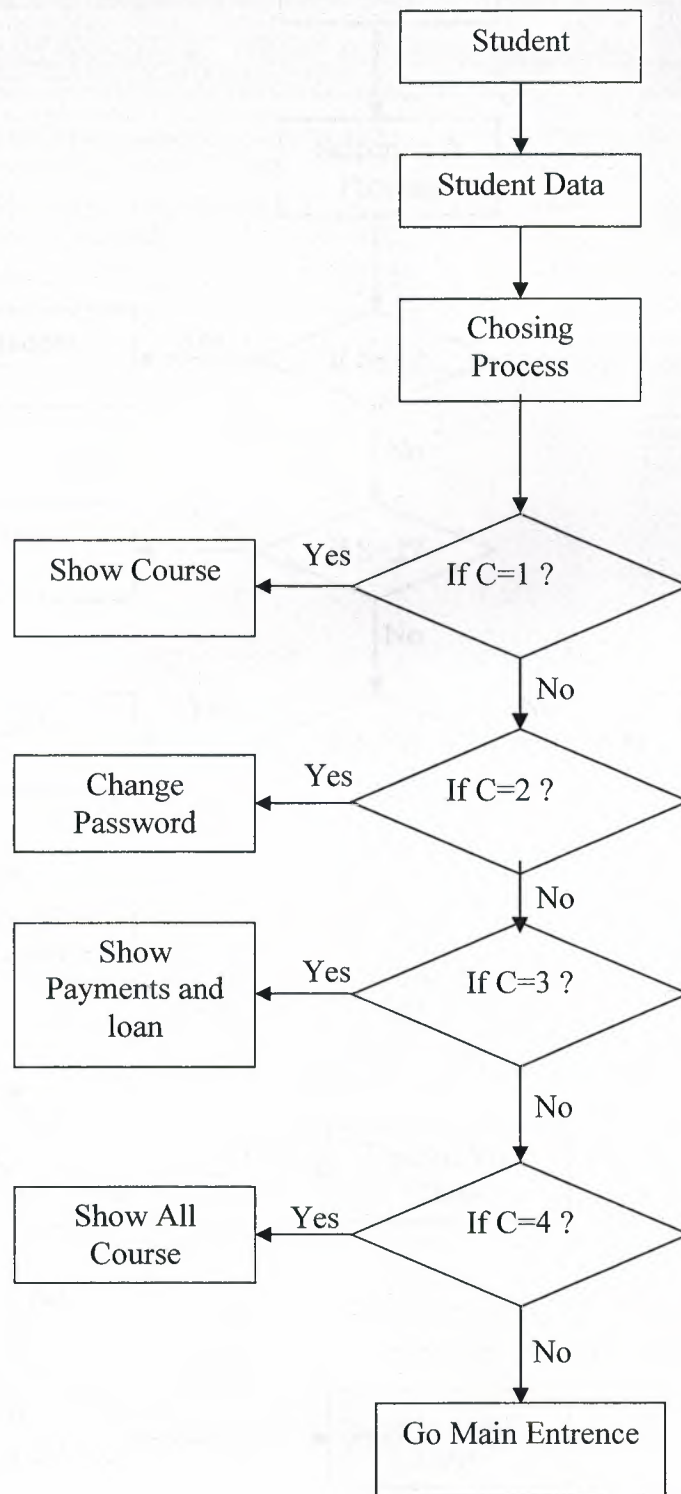


Figure 3.2. Student Entry



### 3.3.Flowchart Of Teacher

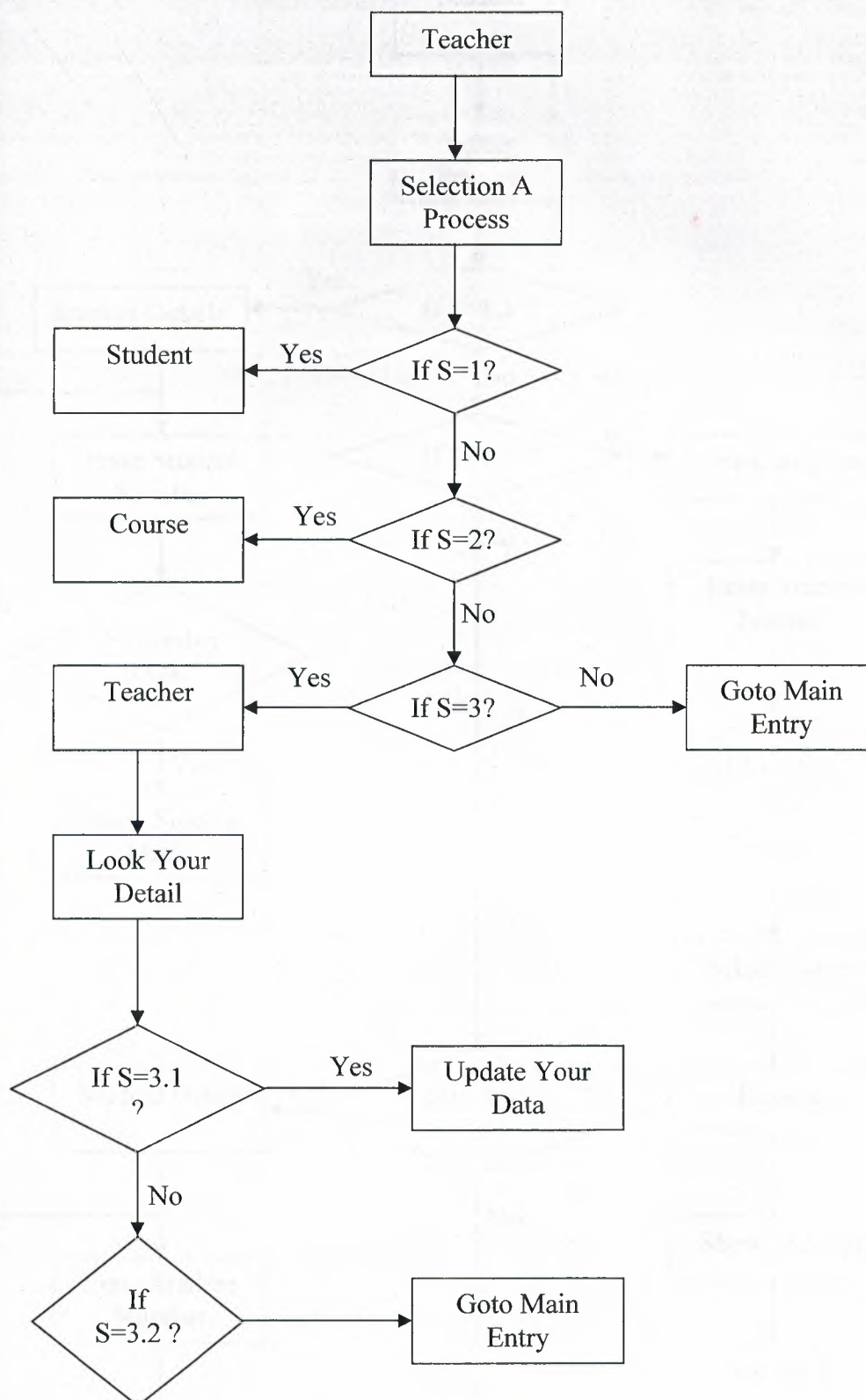
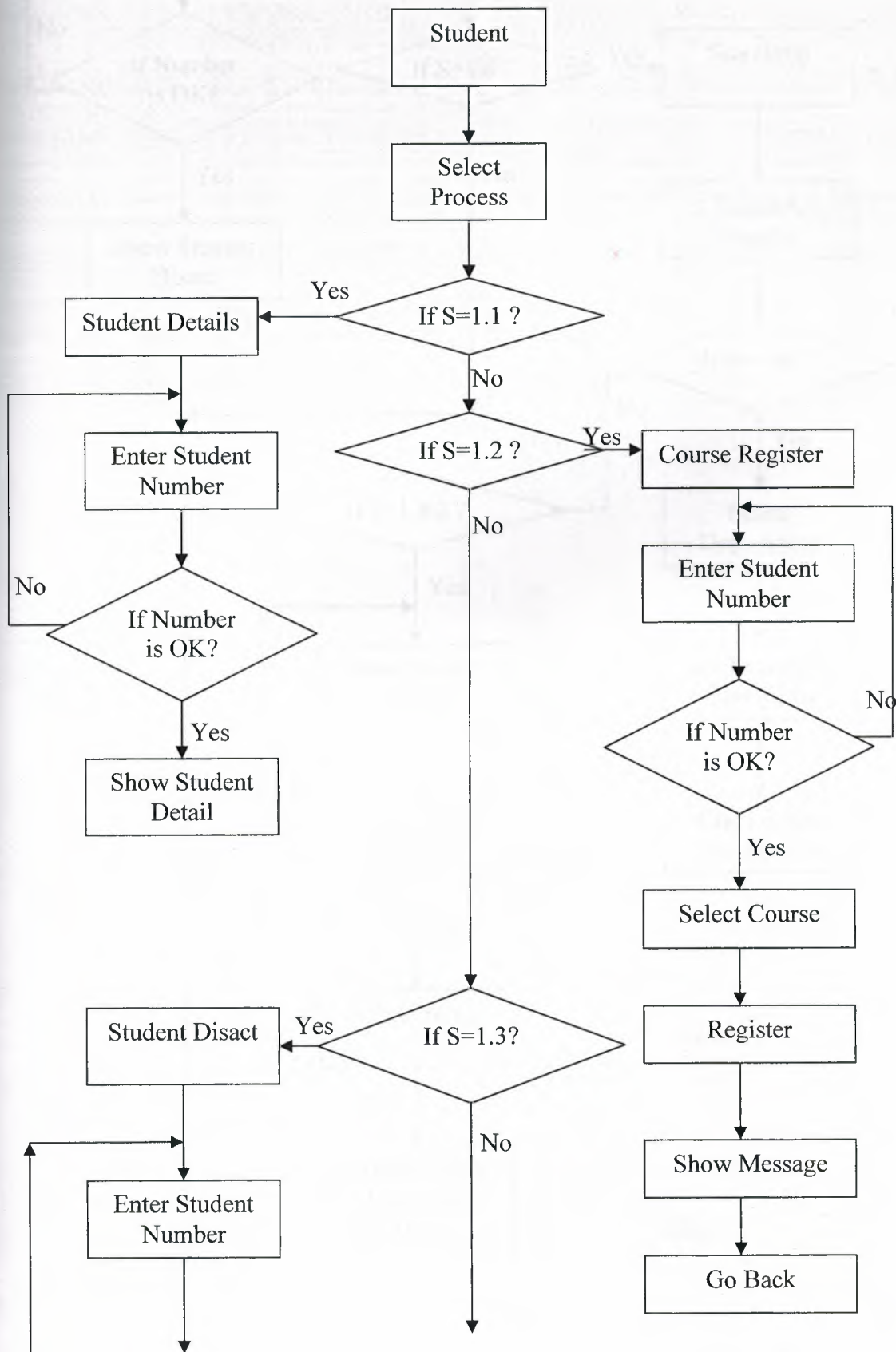
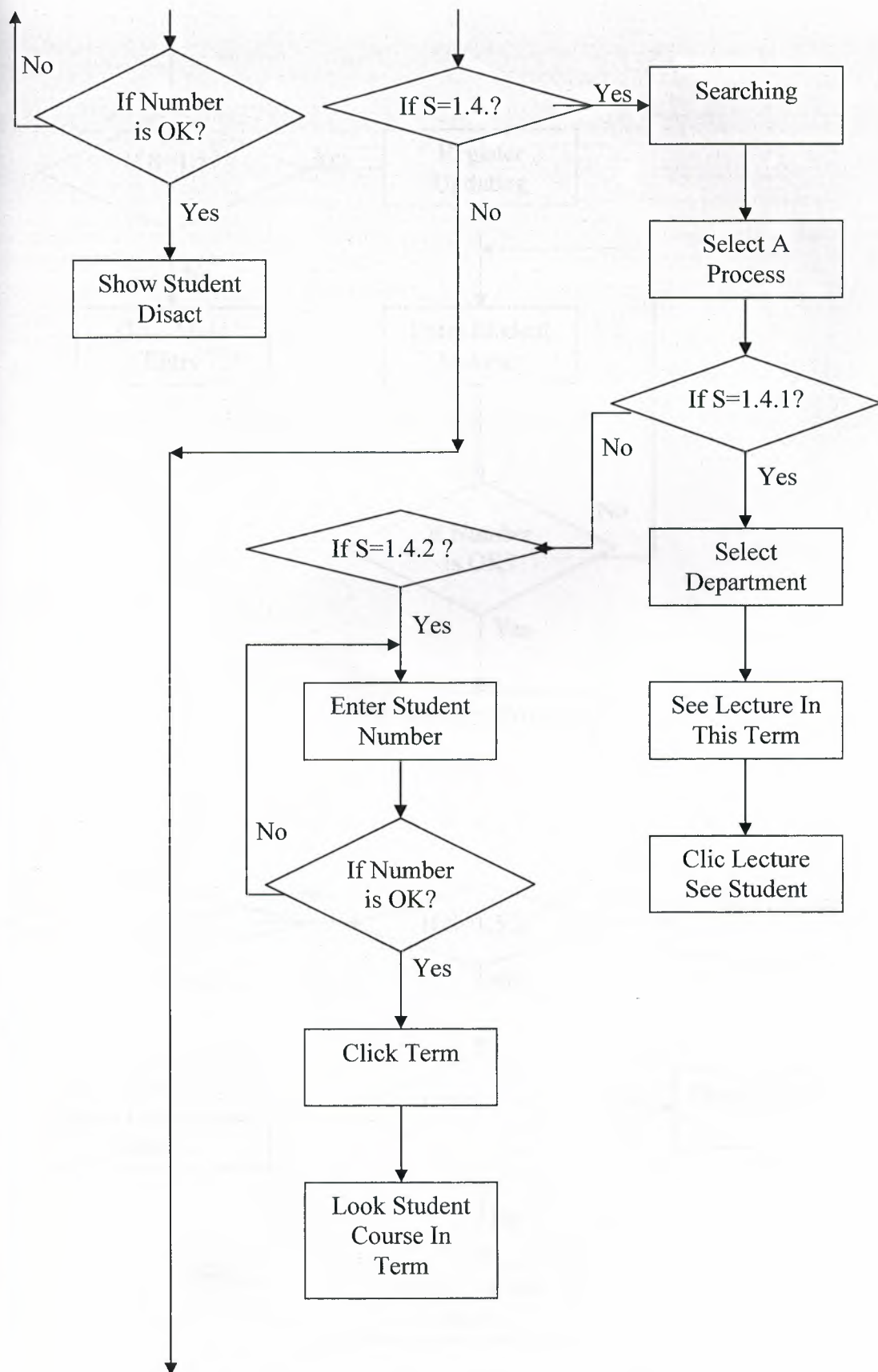


Figure 3.3. Teacher Entry

### 3.3.1.Student Entry From Teacher







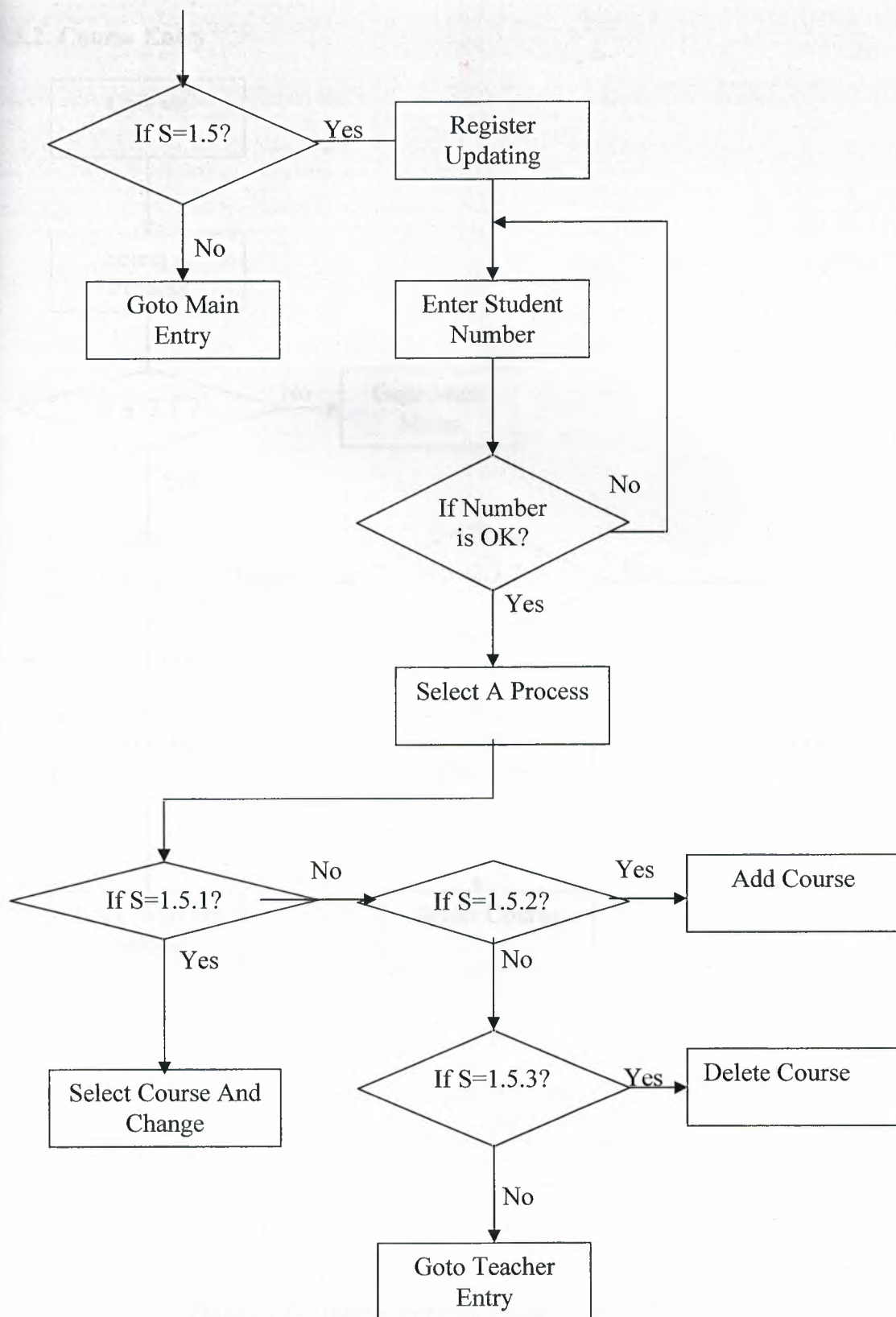


Figure 3.4. Teacher Process For Student



### 3.3.2. Course Entry

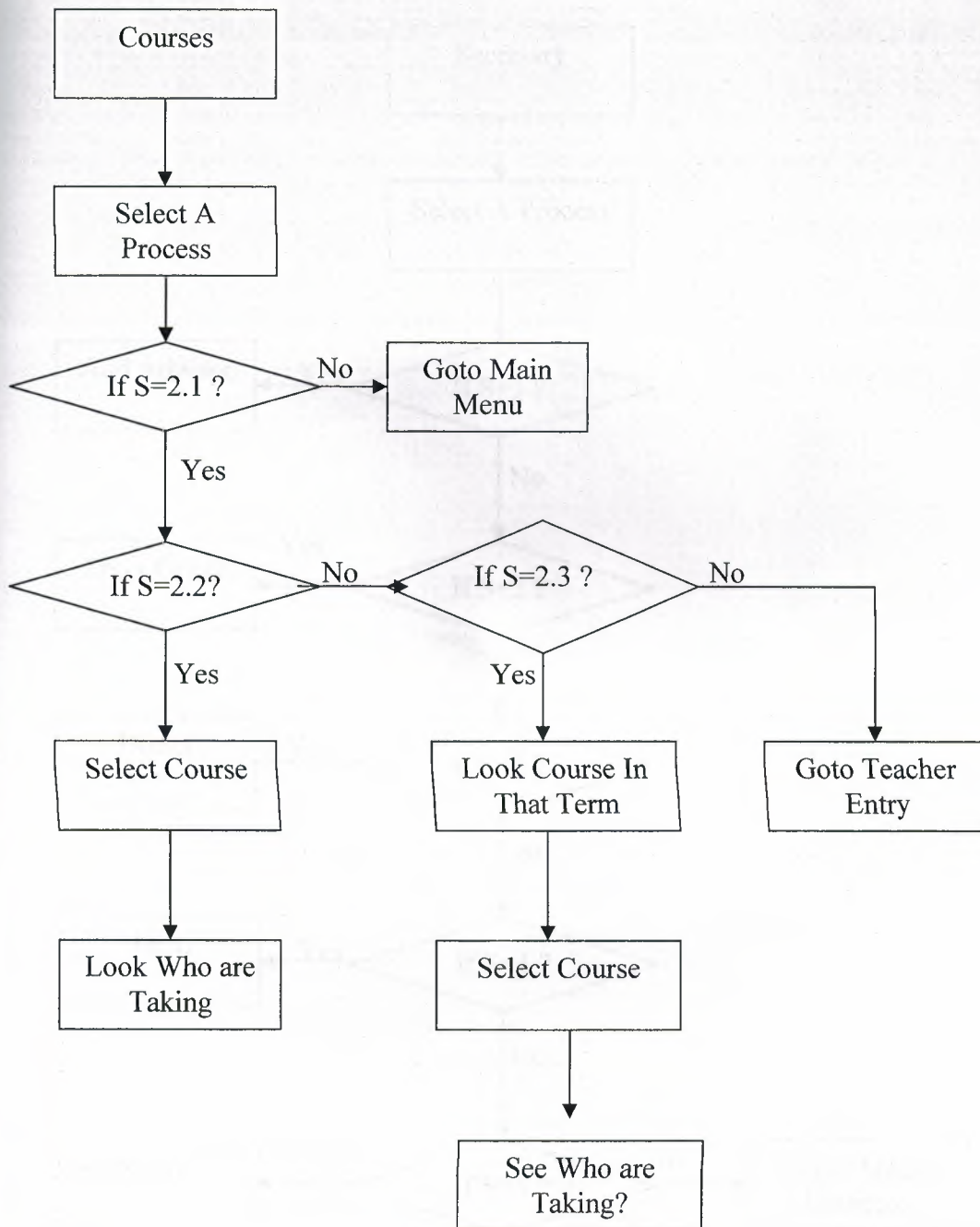


Figure 3.5. Course Process From Teacher Entry

### 3.4. Flowchart Of Secretary

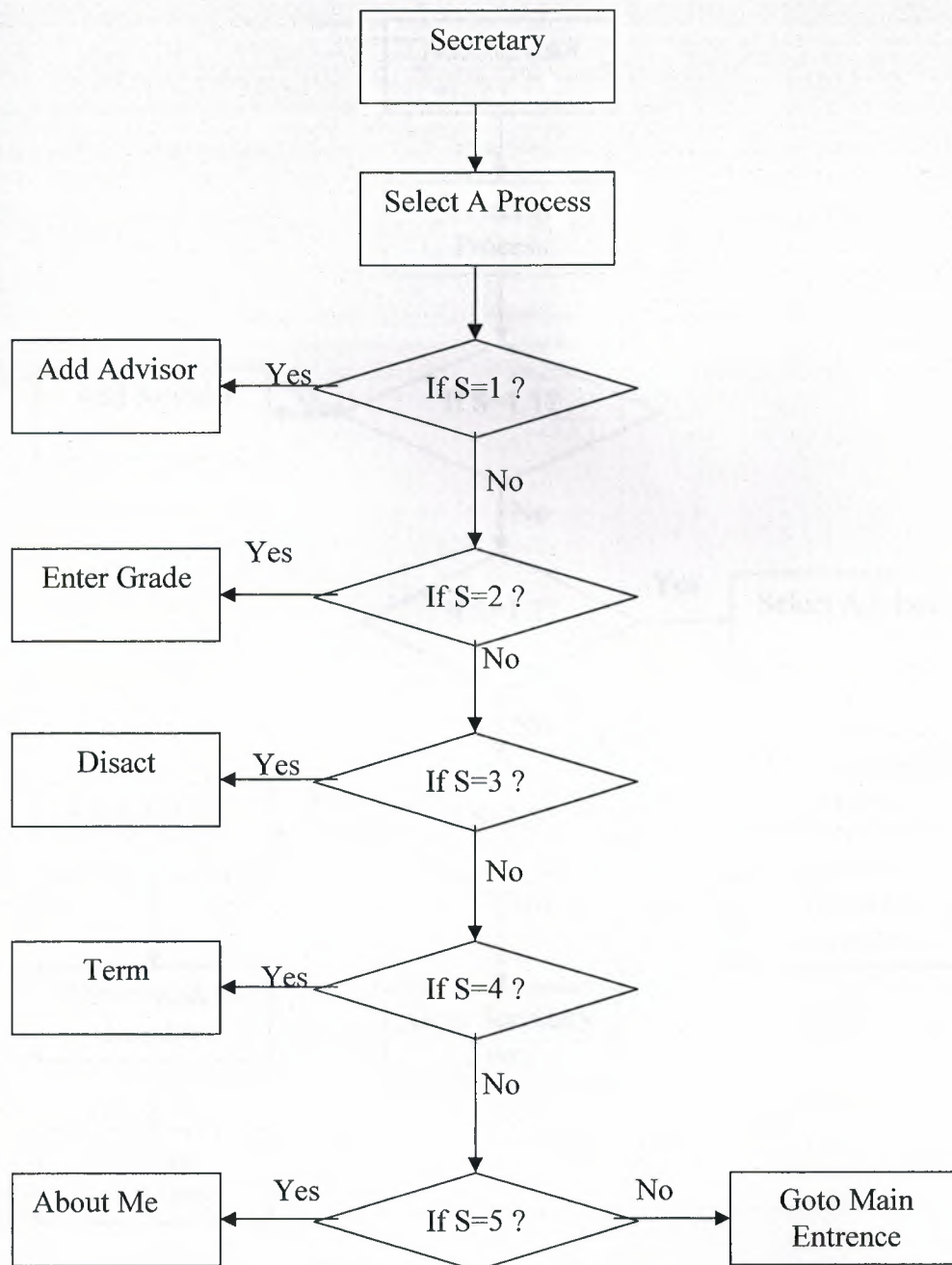


Figure 3.6. Secretery Process



### 3.4.1. Fowchart Of Adding Advisor

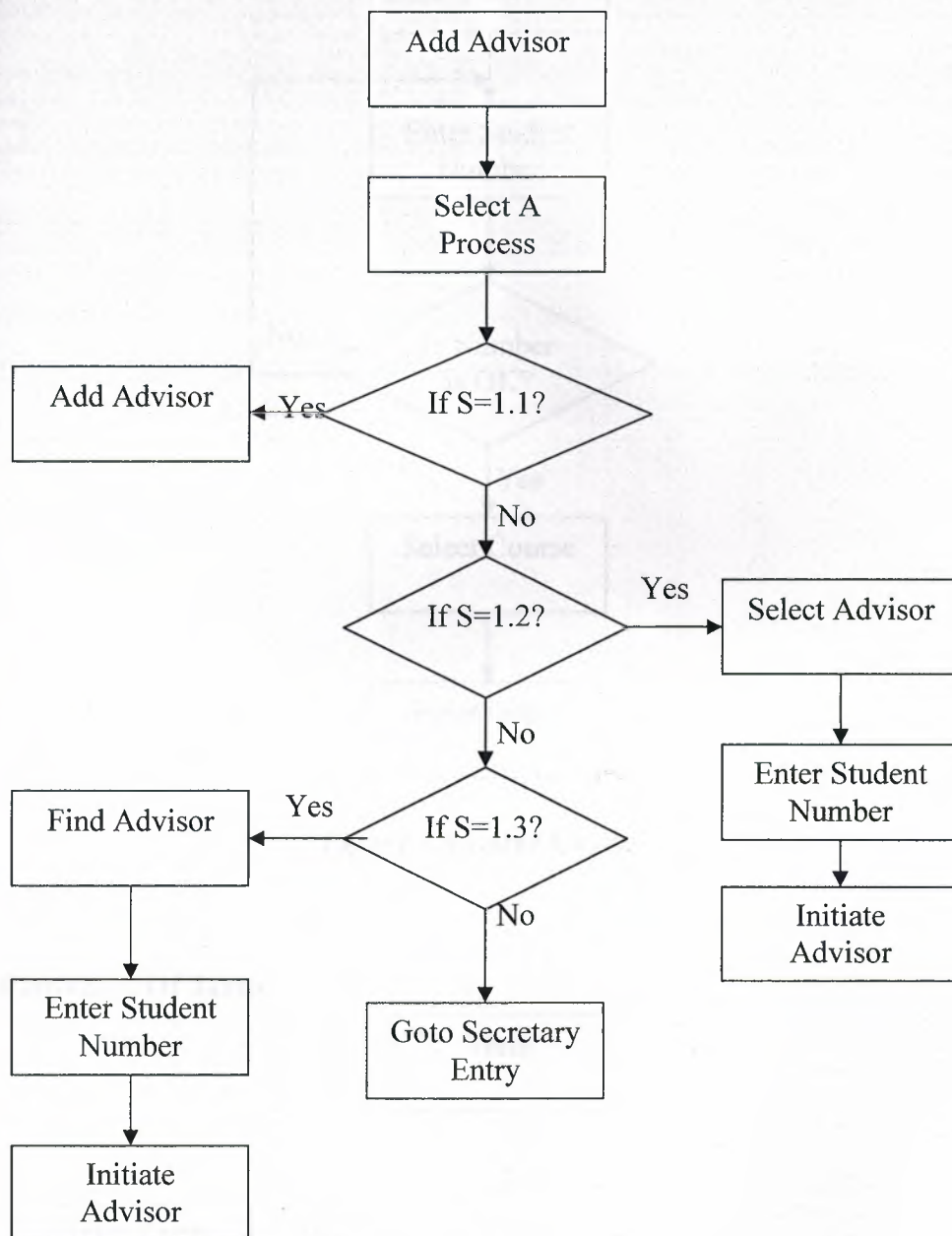


Figure 3.7. Initiate Advisor

### 3.4.2. Grade Entry

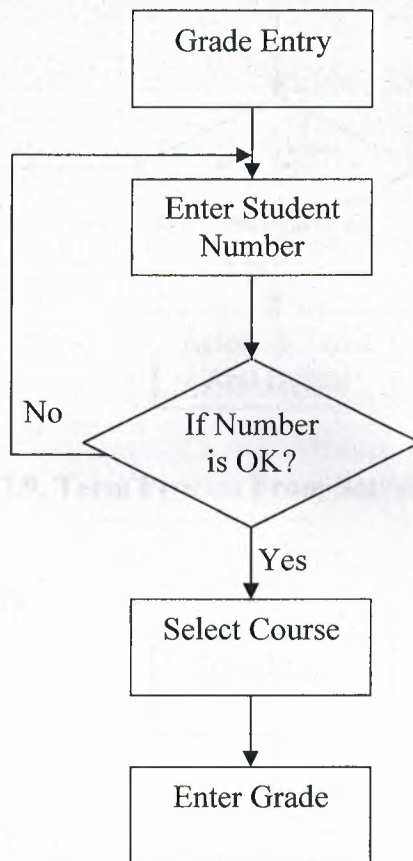
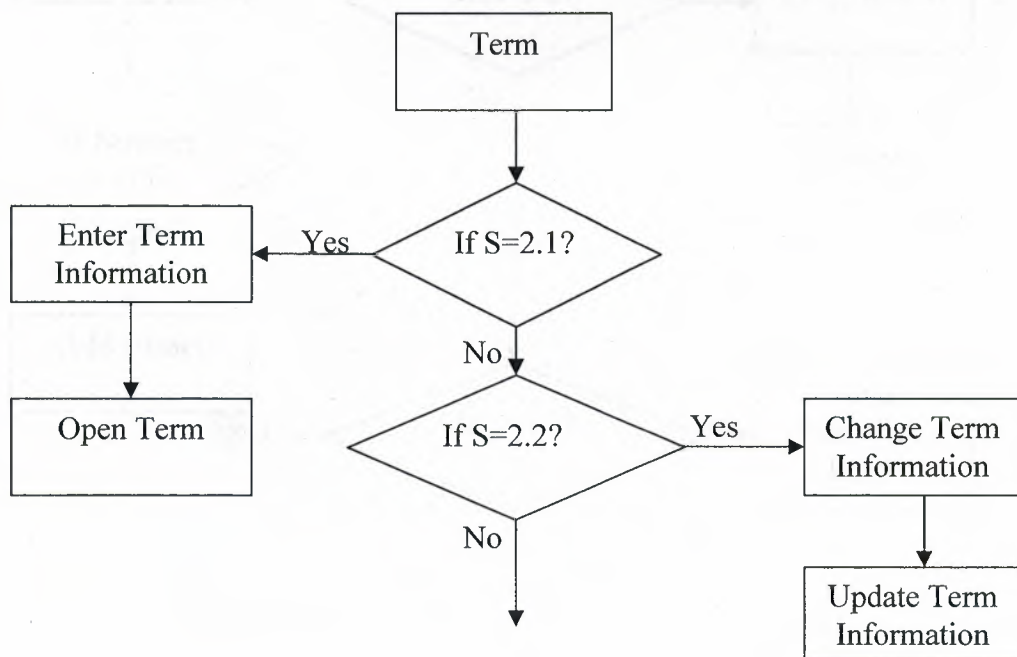


Figure 3.8.Enter Grade

### 3.4.3. Flowchart Of Term





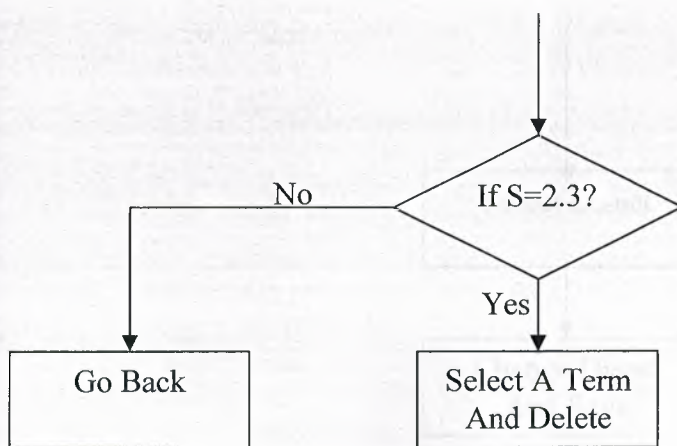
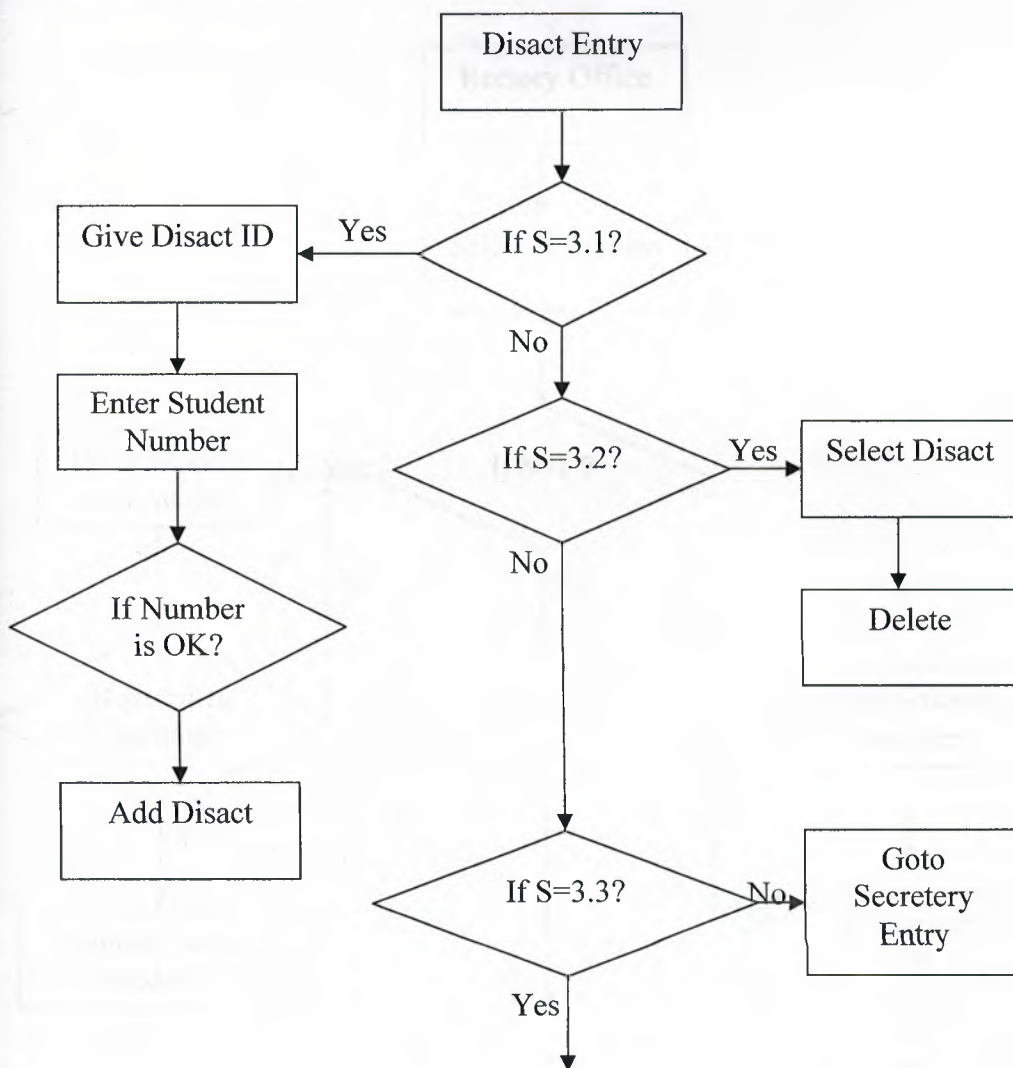
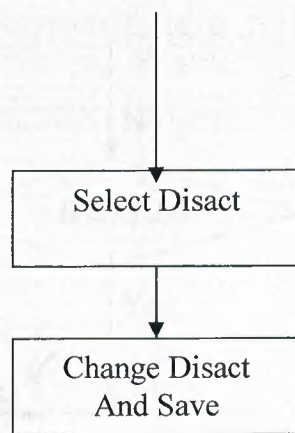


Figure 3.9. Term Process From Secretary Entry

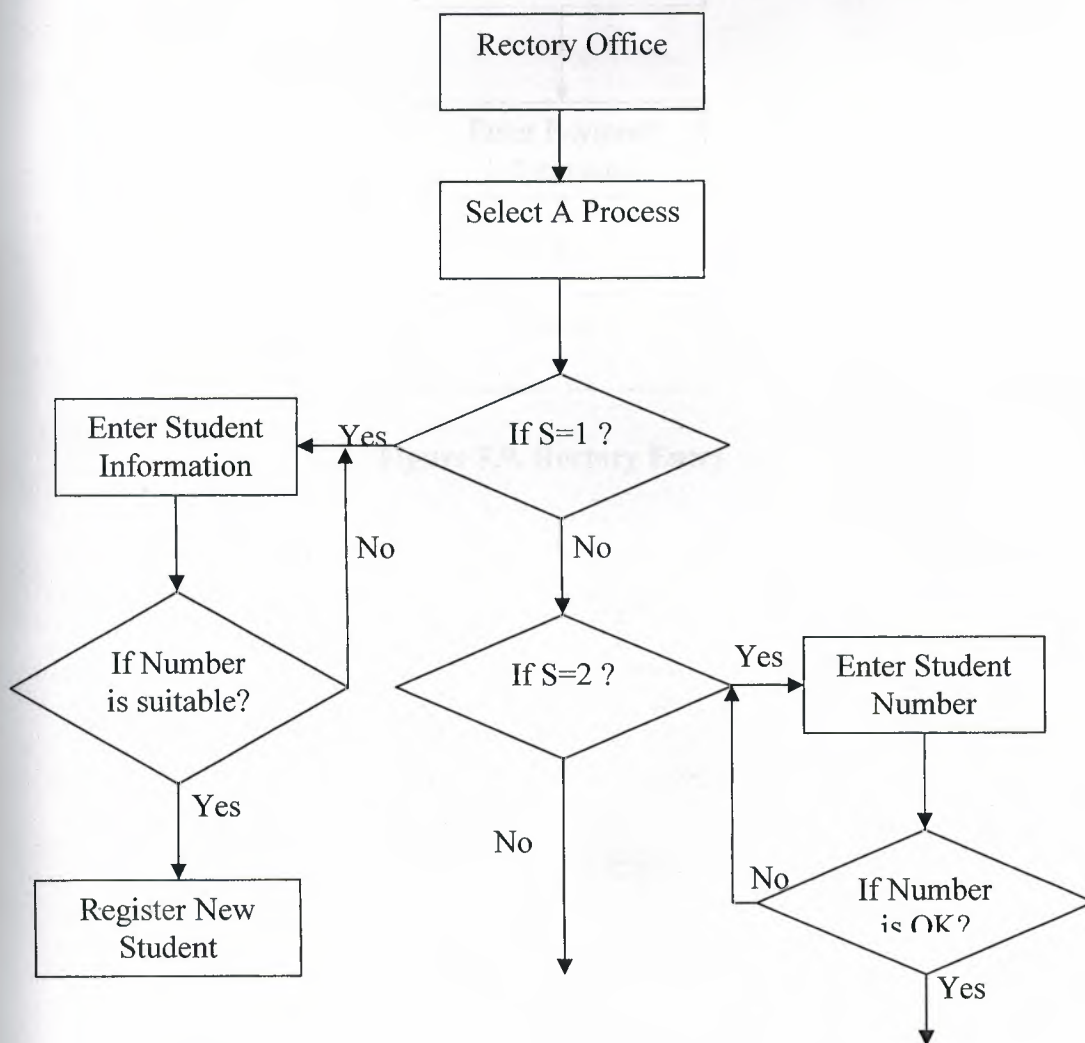
#### 3.4.4.Flowchart Of Disact





**Figure 3.9. Disact Process**

### 3.5. Flowchart Of Rectory Entry





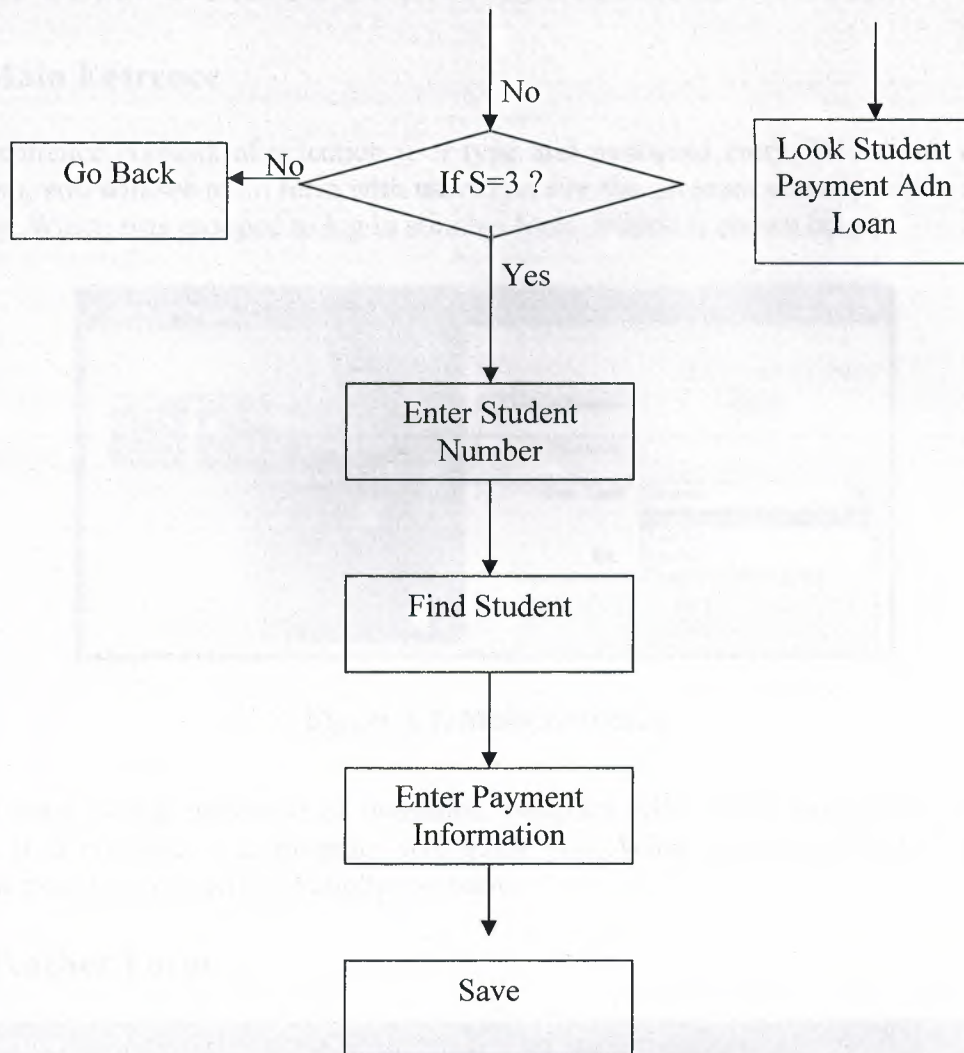


Figure 3.9. Rectory Entry

## CHAPTER FOUR: STUDENT TRACKING SYSTEM

### 4.1. Main Entrence

Main entrence consists of selection user type and password entry. When you run the program, you will see main form with user type. For the program security and to respect all user. Which was grouped to log in suitable form. Which is shown below.

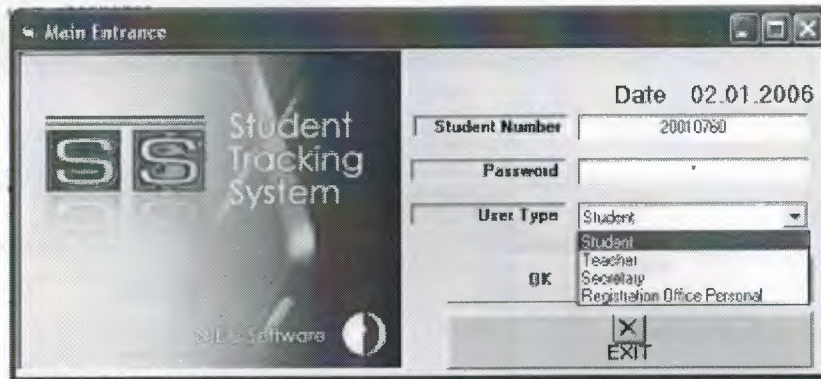


Figure 4.1. Main Entrence

If you enter wrong password or username, program will give a message to you. Of course it is possible. But, program will guide you. What is wrong. When you enter suitable possition you will see another window.

### 4.2. Teacher Form

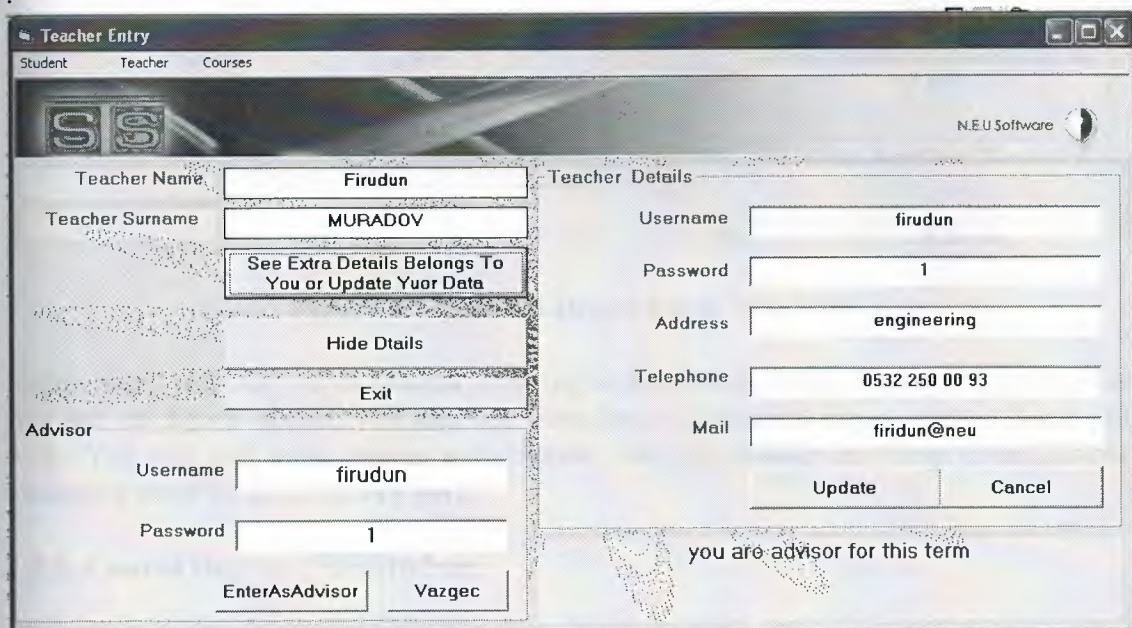


Figure 4.2. Teacher Main Form



When you enter the system as a teacher. If you are advisor program will inform you, as you see in figure 4.2. If you are advisor for that term. Course registration will be activated for you. Additionally you will hide your data by using hide button, also you can update your data as you see on figure4.2.

#### 4.2.1. Student Detail(from teacher entry)

**Figure 4.3.Student Detail(From Teacher)**

In this page, you can search student according to number or name, whatever you know. As you see on figure above, you can see complete information about student what you seek. You can just look student information, can not change anything about his/her. Changing must be from rectory part.

#### 4.2.2. Course Register For Student

As I mentioned before. If you are advisor program will inform you. If you are advisor you can change course register for student or you can add course. Suitable form is shown below.



Advisor Entry(from teacher)

03.01.2006  
Fall 2005-2006

Student Registration Form

Student Details

Student Number: 20010760

Student First Name: ilker

Student Last Name: damar

Department: Computer Engineering

Find Student Courses Will Be Taken

Was Taken

Course	Course Name	Credit	Grade	Type	Teacher Name
com 241	data structures	3	AA	must	Okan DONANGIL
com 313	automatic control	3	CC	must	Doğan İBRAHİM
mat 101	calculus 1	4	AA	must	Deniz ÖNENGÜT
mat 102	calculus 2	4	BA	must	Af DENKER
com300	summer training 2	3		must	Kaan UYAR
com 315	algorithms	3	BB	must	Rahib ABİYEY
com 340	internet programming	3		must	Umit İLHAN
com 320	computer hardware	3	BA	te	Doğan HAKTANIR
com 401	microprocessors 2	3		te	Doğan İBRAHİM
com318	data communications	3		must	Tayseer ASHLANBALEH
com 342	data base management system	3		must	Umit İLHAN
com416	computer network	4		must	Murat TEZER
com 420	nero network	3		te	Adnan KASHMAN

Term Average: Text5

Lecture Will Be Taken

Course Code	Course Name	Credit	Course Type	Pre-Request
com 400	graduation project	4	must	
com 411	software engineering	3	te	com 310, com 342
com 430	visual basic programming	3	te	com 310
com 432	delphi	3	te	com 310
com 442	java programming	3	te	com 310
com 446	object oriented database	3	te	com 342
com 450	oracle	3	te	com 342
com 463	image processing	3	te	com 416
econ 431	economics for engineers	3	must	
ee 207	electrical circuits	3	must	phy 102
ee 208	basic electronic	3	must	ee 207
eng 101	english 1	3	must	
eng 210	english communication skills	3	must	eng 102
erus 201	russian language	3	rite	
man 402	management for engineers	3	must	
mat 205	linear algebra and differential equation	4	must	mat 102
mat 301	numerical analysis	3	must	mat 205
mat 350	probability and statistics	3	must	mat 102
phy 101	general physics	4	must	
phy102	general physics 102	4	must	phy 101

Will Be Taken In This Term

Course Code	Course Description
com 432	delphi
com 463	image processing
ee 207	electrical circuits
com 432	delphi

Register Remove Clear All

Figure 4.5. Course Registration

You can find student just writing student number and can be added course what will be given in suitable form.

#### 4.2.3. Student Disact(from teacher)

Student Disact Form

Searching student Disact

Student Number: 20010760

Student Name: ilker

Student Surname: damar

Department: Computer Engineering

Show Disact

Disact Information

Term Name	Date	Information	Result
Fall 2005-2006	23.12.2005	copy	failed
Fall 2005-2006	23.12.2005	copy from 567	failed

Figure 4.6. Student Disact(from teacher)

In this form you can see student disact. Sometimes, to get like that information for teacher can be better for grading lecture about student .



#### 4.2.4. Searching Form(from teacher)

I hope that, it can be usefull. Sometimes teacher may want to see that 'how many student taking my course and who are taking'. This information can be benefit. Form is shown below.

**Searching(from teacher)**

N.E.U Software

**Course In This Term**  
Computer Engineer

**Student Taking In This Term**

Student Number	Student Name
20010733	metin ulay
20010760	ilker damar

**Course Information**

Course Code	Description	Credit	Grade	Teacher Name
com300	summer training 2	3	BB	kaan uyar
com315	algorithms	3		rahil abiyev
com340	internet programming	3		umit ilhan
com320	computer hardware	3	BA	dogan haktanr
com401	microprocessor 2	3		dogan ibrahim
com318	data communications	3		tayseer ashtenbalch
com342	data base management system	3		umit ilhan
com416	computer network	4		murat tezer
com420	network	3		adnan kashman

**Student Information**

**Student Number**  
20010760

**Student Name**  
ilker

**Student Surname**  
damar

**Department**  
Computer Engineering

**Terms**  
Fall 2002-2003  
Fall 2003-2004  
Fall 2004-2005  
Spring 2001-2002  
Spring 2002-2003  
Spring 2003-2004  
Spring 2004-2005  
Summer 2003  
Summer 2004  
Summer 2005  
Fall 2005-2006

Figure 4.7. Student Searching Form(teacher)

#### 4.2.5. All Course With Opened Term

**All Lectures(from teacher)**

N.E.U Software

**All Course**  
Computer Engineering

**COMPUTER ENGINEERING**

**List All Courses**

**List All Course In That Term**

**Student No**  
20010760

**Student Name**  
ilker damar

Course Code	Course Name	Credit	Type	Prerequisite	Educator
com111	introduction to computer engi	3	must		okan DONANGIL
com121	discrete structures	3	must		firudin MURAD
com122	digital logic fundamentals	4	must	com121	ozgur OZERDEM
com141	introduction to programming	3	must		okan DONANGIL
com142	c programming	4	must	com141	rahil ABIYEV
com200	summer training period	4	must		kaan UYAR
com211	digital logic systems	4	must	com122	kaan UYAR
com241	data structures	3	must	com111	okan DONANGIL
com252	computer architecture	4	must	com211	kaan UYAR
com300	summer training 2	3	must	com200	kaan UYAR
com301	microprocessors	4	must	com252,ee	kaan UYAR
com310	object oriented programming	4	must	com142	rahil ABIYEV
com312	operating system	4	must	com342	umit ILHAN
com313	automatic control	3	must	com310	dogan IBRAHIM
com315	algorithms	3	must	com342	rahil ABIYEV
com318	data communications	3	must	ee 208,com	tayseer ASHLAN
com320	computer hardware	3	te	com318,com	dogan HAKTANI
com340	internet programming	3	must	com241	umit ILHAN
com342	data base management system	3	must		umit ILHAN
com400	graduation project	4	must		

Figure 4.8. All Course Form(from teacher)

#### 4.2.6. Course Register Updating

There is a possibility making wrong things. Sometimes student wants to change course. When you encountered like that situation. You can change or update student course by using another form. For this situation, course updating form is suitable. Which is shown below.

**Register Updating**

Student Number: 20010760

Student Name: ilker

Student Surname: damar

Department: Computer Engineering

COM 411

**Find Student**

**Taking In This Term**

**Fall 2005-2006**

CourseCode	Course Name	Credit	Type	Pre-Request	Educator
com300	summer training 2	3	must	com 200	Kaan UYAR
com 340	internet programing	3	must	com 318,com 310	Umit ILHAN
com 401	microprocessor 2	3	te	com 301	Doğan IBRAHIM
com318	data cominications	3	must	ee 208,com313	Tayseer ASHLANBALEH
com 342	data base management s	3	must	com 241	Umit ILHAN
com416	computer network	4	must	com 318,com 340	Murat TEZEFİ
com 420	nero network	3	te	com 416	Adnan KASHMAN

**Change** **Add Course** **Delete**

**Figure 4.8.Course Updating(from teacher)**

If you are advisor you can change course registration or add course for suitable student. Also you can see student photo onto form. Which reduce wrongs or reduce confusing student each other.



### 4.3. Student Entry

That page is for student. Student, who are run in that part, who can see just information belong to himself/herself. Student can not change anything about 'what he/her has seen' except student password. You can see student form below.

**Student Informations**

Student Number: 20010760  
 Student Name: Ilker  
 Student Surname: DAMAR  
 Department: Computer Engineering

**Terms And SGPA**

Terms: Fall 2002-2003, Spring 2002-2003, Spring 2003-2004, Fall 2004-2005, Fall 2005-2006  
 SGPA: 3.4

**Change Password Menu**

Change Password  
 Enter New Password:   
 Confirm Password:   
 Cancel Save

**Term Information**

Course Code	Course Description	Credit	Teacher Name	Grade
COM300	Summer Training 2	3	Kaan UYAR	BB
COM 315	Algorithms	3	Rahib ABYEV	
COM 340	Internet Programming	3	Umit ILHAN	
COM 401	Computer Hardware	3	Doğan HAKTANIR	BA
COM 401	Microprocessor 2	3	Doğan İBRAHİM	
COM318	Data Communications	3	Tayseer ASHLANBALEH	
COM 342	Data Base Management Systems	3	Umit ILHAN	
COM416	Computer Network	4	Murat TEZER	
COM 420	Nero Network	3	Adnan KASHMAN	

Total Credit : 25      Term Average : 3.38

**Student Loan Information**

Loan Information	Terms	Last Pay Date	Tax	Amount
Free Fee Semester	Fall 2004-2005	23.12.2004	2	750

Total Amount : 750

**Student Payment Information**

Bill No	Terms	Date	Pay Type	Amount
2	Fall 2002-2003	12.10.2002	ch	100
4	Spring 2002-2003	27.04.2003	c	1850
5	Fall 2004-2005	28.12.2004	c	500

Total Amount : 2450

Exit

Figure 4.9. Student Form

As you see onto form, you can see your payments and loan. Also you can see your all course upto now, what you have taken with clicking to 'see your all lectures' button. Additionally, you can see your average with by term.

### 4.4. Secretary Form

In this form, there are secretary working. Firstly, secretary see her data. She can update.

**Secretary Form**

Advisor   Grade   Disact   Term   Aboutme

**Update Your Data**

Name: Ayse      Username: ayse      Password: 1      Update      Cancel

Surname: Yurun

Figure 4.10. Secretary Main Form

In the same form, secretary can enter student grade. Just writing student number, she can obtain student data about courses and can enter easily. If she want to change something(which is authorized), she can do.

**Figure 4.11. Grade Entry(from secretary)**

#### 4.4.1. Initiating Advisor

Before initiating advisors, secretary must open the term firstly. After opening term, secretary can initiate advisors for opened term. Initiating advisor consist of many choices. You can initiate teacherID, with selection department. Also you can see teacher name with his/her department. From there you can select and add for opened term. Which is shown below, you can follow, what shoul do from the figure.



Figure 4.12. Initiating Advisor(from secretary)

#### 4.4.2. Term Opening

Figure 4.13. Term Opening

Secretary must open the term firstly, then can do other things. As you know that, task is procedural one by one. Dong like that will give facilities to secretary. As you see on figure above, term opening is very easy, you have a calendar on the screen and you can select from calendar and open term.



#### 4.4.3. Disact Entry

I think that, everybody knows from their experience. Disact is necessary. Unfortunately, i had to put that form, as you know somebody can not obey rules so that part must be.

**Disact Entry**

Disact ID: example: s0301 03.01.2006

Give A Disact ID:

Student NO:  18:04:51

Disact Info:

Disact Result:

☒ Delete ☐ Update

**Disact Information**

Student	Disact Reason	Disact Date	Term Name	Disact Result
20010733	copy	12.11.2002	Fall 2002-2003	
20010760	copy	23.12.2005	Fall 2005-2006	failed
20010760	copy from 567	23.12.2005	Fall 2005-2006	failed
20010760	copy	02.01.2006	Fall 2005-2006	failed

**Disact Delete OR Update**

Disact ID:

Student NO:

Student Name:

Term Name:

**Updating**

Disact Info:

Disact result:

**Figure 4.14. Student Disact (from secretary)**

Disact adding can be done filling text and pressing the 'add' button. Everybody is a human, who can make mistake or sometimes you can need a changing. Doing this process is possible only selection disact and changing data or adding new or deleting completely.



#### 4.4.4.Course Opening

Course opening is done by secretary. It is not hard. Only choosing course and click open course button. Secretary can see opened course and will be opened course at the same time. It is more facilities for secretary. It reduce wrong things.

**Open Course**

Fall 2005-2006 03.01.2006

c01

**Course In This Term**

**Course Code**

com 320  
com 312  
com 111  
com 121  
com 211  
com 241  
com 313  
com 310  
com 252  
com 450  
com 315  
com 340  
com 301  
com 420  
com 430  
com 442

**Course Code**

com 111  
com 121  
com 122  
com 141  
com 142  
com 200  
com 211  
com 241  
com 252  
com 300  
com 301  
com 310  
com 312  
com 313  
com 315  
com 318  
com 320  
com 340  
com 342  
com 400  
com 401  
com 411  
com 416  
com 420  
com 430  
com 432  
com 442  
com 446  
com 450  
com 463  
econ 431  
ee 207  
ee 208  
eng 101  
eng 210  
erus 201  
man 402  
mat 101  
mat 102  
mat 205  
mat 301  
mat 350  
phy 101

**Educator**

okan donangil  
firudun muradov  
özgür özerdem  
okan donangil  
rahib abiyev  
kaan uyar  
kaan uyar  
okan donangil  
kaan uyar  
kaan uyar  
kaan uyar  
rahib abiyev  
umit ilhan  
doğan ibrahim  
rahib abiyev  
tayseer ashlanbaleh  
dogan haktanır  
umit ilhan  
umit ilhan  
  
doğan ibrahim  
adil amircanov  
murat tezer  
adnan kashman  
umit ilhan  
rahib abiyev  
adil amircanov  
besime erin  
kemal ataman  
adnan kashman  
mustafa gunduz  
kaan uyar  
kaan uyar  
aydin beyzade  
sevilay cangul  
  
mustafa gunduz  
demir onengut  
ali denker  
ali denker  
faik radvan  
faik radvan  
sameer ikhladeer

**List All Courses**

Computer Engineering

**Open Courses**

com 301  
com 411  
com 318  
com 450

**Open Courses**

**Remove Item**

**Clear List**

**EXIT**

Figure 4.15.Opening Course

I hope that, you agree with me about use of the this part. As you see above. Opening course is very easy. Just select course and prees 'open course button'. If you are any fault you can compensate easy by 'remove item' or 'clear list button' as ou see on figure above.

#### 4.5. Rectory Entry

Actually, this part must be comprehensive as much as other part. But I cut it little short. It is not mean that, this part is not important. This part can be expanded more and more. I will tell about that part with figure shortly.

**Rectory Entry**

Student

**Registration**

**New Registration**

firstly select department

**Major Details**

Student Name

Student Surname

Student Sex

Department ID

Department Name

**Department**

Student NO

Student Password

Student Finish

Register Date

**Give A Number**

**More Details**

Father Name

Mother Name

Date Of Birth

Place Of Birth

Country

Nationality

High School

Address

Telephone

Mail

**Register Student**

**Figure 4.16. New Student Register**

As I determined before, that part can be expanded more and more. I tried to do less filling text. In frame Which is above of form, more of the text is filled by choosing department and pressing 'give a number button' automatically. Other part can not be automatically filling. As you gues that part changable for every student.

#### **4.5.1.Student Payment**

program must contain that part whether the university gives education with money or without money. Every student gives money to university. So that part must be. Actually, that part is needed by rectory. In this part, student payment can be seen and If a students pay his/her loan, this payment can save by suitable person. You will see below more clear.



[illegible]

**Figure 4.17. Student Payments**

Suitable person can add payment as you see above. Also you can see student picture. Which gives to you help do not making mistake.

I want to tell about the future of the program. In this program student picture was used frequently. This reason is to reduce faults and confusing information each other belong to student. Also I tried to do best about filling text. More filling text means more wrong. Because you can do wrong with filling text. I tried that, everything could be automatic and selection from boxes or finding with button. More automatic or selection- as you know- means less fault.

Lets come about future, taht program can be runned over internet or over the network. Of course it is possible and also easy, just with IP address.

## CONCLUSION

In this graduation project Visual Basic Programming was used to create a simple student tracking program. That can be used by all faculties. Program is open for improvement. To expand is so simple. The development of student tracking system includes such problem as; student and course registration, educator registration, and student payment following.

For each block the special menu is designed and allows any user to easily realise, update and apply the searching process. The advantage of the Visual Basic Programming particularly allows the programmer to create a perfect view and control the database easily. Reason of easy useage is based on creating a perfect view.

The security of this programme and the authorization for reaching the data was the most important factor considered in all step of creation of this program. Precisely automation in all aspect of our life is perfect, but in any education system the back up units and printed records of old data is absolutely needed. In this programme currently the print commend and its required facilities are not available and shoul be added in future. The other important things was not added here was the time table and excat rectory entry with entrence. Dependent on; It can be possible to make more usefull the program of a faculty. In this project it means that, to say that also it is possible to upload or working over network for more usage. Visual basic allows like that operation.

Finally, putting last point for that programme, this programme can be updated in future and can be added extra parts due to factors what will be needed in future. That may be something was ignored or due to necessities that was imagined after preparing programme.





## REFERENCES

1. Ihsan Karagülle and Zeydin Pala, Microsoft Visual Basic 6.0 Pro, Türkmen Printing House, Istanbul ,2001.
2. H.M.Deitel, P.J.Deitel and T.R.Nieto, Visual Basic 6: How To Program, Prentice Hall,Inc. Upper Saddle River., New Jersey, 1999
3. A research for finding Visual Basic code, Fined November 01, 2005 from the World Wide Web "<http://www.vbtürk.com/allcodes/capture.htm>"
4. A guide for writin about Visual Basic description, Retrieved December 05, 2005 from the World Wide Web <http://www.vbtutor.net/lesson.html>.
5. A guide research for writing program. Retrieved October 10, 2005 from the World Wide Web "<http://www.programlama.com>"

## APPENDICES

```
Public db As Database
Public student As Recordset
Public department As Recordset
Public term As Recordset
Public advisor As Recordset
Public personal As Recordset
Public teacher As Recordset
Public grade As Recordset
Public courseinfo As Recordset
Public course As Recordset
Public disact As Recordset
Public payment As Recordset
Public loan As Recordset
Public secretarycontrol As String
Dim studentnumber As String
Dim teacherid As String
Dim personalid As String
Dim termidcontrol As Integer
Public tidcontrol As String
Dim control1 As Integer
Dim termidcontrol2 As String
Public tidcontrol1 As String
Private Sub Combo1_Click()
If Combo1.Text = "Student" Then
Label1.Caption = "Student Number "
Else
Label1.Caption = "User Name "
Text1.Text = ""
Text2.Text = ""
Text1.SetFocus
End If
End Sub
Private Sub Command1_Click()
Dim control1 As Integer
Dim siralama As Integer
Dim siralama2 As Integer
Dim extra As String
Dim sira As String
Dim sira2 As String
Dim i As Integer
Dim j As Integer
control1 = 0
Call tabloac
termidcontrol = 0
'control of secretary entry
If Combo1.Text = "Secretary" Then
If Text1.Text <> "" And Text2.Text <> "" Then
```



```

personal.MoveFirst
While Not personal.EOF
If personal.Fields("pusername") = Text1.Text Then
control1 = 1
If personal.Fields("ppassword") = Text2.Text Then
Form6.Show
Form6.Text1.Text = StrConv(personal.Fields("pname"), vbProperCase)
Form6.Text2.Text = StrConv(personal.Fields("psname"), vbProperCase)
Form6.Text3.Text = personal.Fields("pusername")
Form6.Text4.Text = personal.Fields("ppassword")
secretarycontrol = personal.Fields("pid")
Form6.Text11.Text = secretarycontrol
Form1.Hide
control1 = 2
End If
If control1 <> 2 Then
MsgBox ("your password is wrong")
Else
Call Form1.tablokapat
Exit Sub
End If
If control1 = 1 Then
MsgBox ("your username is wrong")
End If
End If
personal.MoveNext
Wend
End If
End If
'end of thr secretery entry
'teacher icin giris kontrolu
If Combo1.Text = "Teacher" Then
If Text1.Text <> "" And Text2.Text <> "" Then
teacher.MoveFirst
While Not teacher.EOF
If teacher.Fields("tusername") = Text1.Text Then
control1 = 1
If teacher.Fields("tpassword") = Text2.Text Then
tidcontrol1 = teacher.Fields("tid")
Call Form1.tablokapat
Call Form11.findtermid
Call Form1.tabloac
advisor.MoveFirst
While Not advisor.EOF
If advisor.Fields("tid") = tidcontrol1 And advisor.Fields("termid") =
Form11.termid11 Then
Form3.Label13.Caption = "you are advisor"
Form3.MnStudentRegister.enable=false
End If
advisor.MoveNext

```

```

Wend
Form3.Show
Form1.Hide
Form3.MnStudentRegister.Enabled =false
Form3.Text1.Text = StrConv(teacher.Fields("tname"), vbProperCase)
Form3.Text2.Text = UCase(teacher.Fields("tsurname"))
Form3.Text3.Text = teacher.Fields("tusername")
Form3.Text4.Text = teacher.Fields("tpassword")
Form3.Text5.Text = teacher.Fields("taddress")
Form3.Text6.Text = teacher.Fields("ttel")
Form3.Text7.Text = teacher.Fields("tmail")
Form3.Text8.Text = teacher.Fields("tid")
Form3.Text9.Text = teacher.Fields("tusername")
Form3.Text10.Text = teacher.Fields("tpassword")
control1 = 2
End If
End If
teacher.MoveNext
Wend
If control1 = 2 Then GoTo Form3
If control1 = 1 Then
MsgBox ("password is wrong")
Text2.Text = ""
Text2.SetFocus
Else
MsgBox ("username is wrong")
Text1.Text = ""
Text1.SetFocus
End
Else
MsgBox ("enter user name and password")
End If
End If
Form3:
'teacher için giriş kodu kontrolu sonu
'student için giriş kodu kontrolu
If Combo1.Text = "Student" Then
'textlerin boş olup olmama kontrolü
If Text1.Text <> "" And Text2.Text <> "" Then
student.Index = "primarykey"
student.Seek "=", Text1.Text
'öğrenci no bulunmama kontrolu
If student.NoMatch = 0 Then
'password kontrollü
studentnumber = student.Fields("stno")
If student.Fields("stpassword") = Text2.Text Then
'form2 textlerin doldurulması
Form2.Text1.Text = student.Fields("stno")
Form2.Text2.Text = StrConv(student.Fields("stname"), vbProperCase)
Form2.Text3.Text = UCase(student.Fields("stsurname"))

```



```

department.Index = "primarykey"
department.Seek "=", student.Fields("deptid")
If department.NoMatch = 0 Then
Form2.Text4.Text = StrConv(department.Fields("deptname"), vbProperCase)
End If
'form2 textlerin doldurulması sonu
'termlerin bulunması
grade.MoveFirst
While Not grade.EOF
If grade.Fields("stno") = studentnumber Then
term.Index = "primarykey"
term.Seek "=", grade.Fields("termid")
If term.NoMatch = 0 Then
termidcontrol = 0
For i = 0 To Form2.List7.ListCount - 1
If Form2.List7.List(i) = grade.Fields("termid") Then
termidcontrol = 1
End If
Next
If termidcontrol <> 1 Then
Form2.List7.AddItem grade.Fields("termid")
Form2.List1.AddItem term.Fields("termname")
End If
End If
End If
grade.MoveNext
Wend
'liste sıralam
For i = 0 To Form2.List1.ListCount - 2
For j = i To Form2.List1.ListCount - 2
siralama = Val(Right(Form2.List1.List(i), 4))
siralama2 = Val(Right(Form2.List1.List(j + 1), 4))
If siralama > siralama2 Then
extra = Form2.List1.List(j + 1)
Form2.List1.List(j + 1) = Form2.List1.List(i)
Form2.List1.List(i) = extra
extra2 = Form2.List7.List(j + 1)
Form2.List7.List(j + 1) = Form2.List7.List(i)
Form2.List7.List(i) = extra2
End If
Next
Next
For i = 0 To Form2.List1.ListCount - 2
If Right(Form2.List1.List(i), 4) = Right(Form2.List1.List(i + 1), 4) Then
sira = Left(Form2.List1.List(i), 1)
sira2 = Left(Form2.List1.List(i + 1), 1)
If sira = "S" Then
extra = Form2.List1.List(i + 1)
Form2.List1.List(i + 1) = Form2.List1.List(i)
Form2.List1.List(i) = extra

```

```

extra2 = Form2.List7.List(i + 1)
Form2.List7.List(i + 1) = Form2.List7.List(i)
Form2.List7.List(i) = extra2
End If
End If
Next
'liste sıralama sonu
'termlerin bulunması sonu
Call temizle
Call tablokapat
Form2.Show
Form1.Hide
Call Form2.findgpa(Form2.Text1.Text)
Form2.Text7.Text = Form2.gpa
Form2.Label14.Caption = "To Learn Your Lectures In Term Click Term List..."
Else
'password kontrolü sonu
MsgBox ("Your Password Is Wrong Try Again..!")
End If
'öğrenci no bulunmama kontrolu sonu
Else
'yanlış öğrenci no kontrolü
MsgBox ("Your Number Is Wrong Try Again..!")
'yanlış öğrenci no kontrolü sonu
End If
Else
'textlerin boş olma durumu
MsgBox ("You Must Enter The Student Number And Password..!")
'textlerin boş olma durumu sonu
End If
'textlerin boş olup olmama kontrolu sonu
End If
'student için giriş kodu kontrolu sonu
'rectory entry starting
If Combo1.Text = "Registration Office Personal" Then
Call Form1.tabloac
Form1.personal.MoveFirst
While Not Form1.personal.EOF
If Form1.personal.Fields("pusername") = Text1.Text Then
If Form1.personal.Fields("ppassword") = Text2.Text Then
Call tablokapat
Form15.Show
Form1.Hide
Exit Sub
End If
End If
Form1.personal.MoveNext
Wend
Call tablokapat
End If

```



```

Public Function tabloac()
Set Form1.db = OpenDatabase("\University Information\Database\university.mdb")
Set Form1.student = db.OpenRecordset("student")
Set Form1.department = db.OpenRecordset("department")
Set Form1.personal = db.OpenRecordset("personal")
Set Form1.teacher = db.OpenRecordset("teacher")
Set Form1.term = db.OpenRecordset("term")
Set Form1.grade = db.OpenRecordset("grade")
Set Form1.courseinfo = db.OpenRecordset("courseinfo")
Set Form1.disact = db.OpenRecordset("disact")
Set Form1.course = db.OpenRecordset("course")
Set Form1.payment = db.OpenRecordset("payment")
Set Form1.loan = db.OpenRecordset("loan")
Set Form1.advisor = db.OpenRecordset("advisor")
End Function
Private Sub Timer1_Timer()
Label4.Caption = Date
End Sub
Form2-Source Code
Public totcredit As Integer
Public totproduct As Double
Public gpa As Double
Private Sub Check1_Click()
If Check1.Value = 1 Then
Text8.Visible = True
Text9.Visible = True
Label15.Visible = True
Label16.Visible = True
Command2.Visible = True
Command3.Visible = True
Text8.SetFocus
Else
Text8.Visible = False
Text9.Visible = False
Label15.Visible = False
Label16.Visible = False
Text8.Text = ""
Text9.Text = ""
Command2.Visible = False
Command3.Visible = False
End If
End Sub
Private Sub Command1_Click()
Call temizle2
Form2.Text7.Text = ""
Form1.Show
Form2.Hide
End Sub
Private Sub Command2_Click()
If Text8.Text <> "" And Text9.Text <> "" Then

```

```

If Text8.Text = Text9.Text Then
Call Form1.tabloac
Form1.student.Index = "primarykey"
Form1.student.Seek "=", Text1.Text
If Form1.student.NoMatch = 0 Then
Form1.student.Edit
Form1.student.Fields("stpassword") = Text8.Text
Form1.student.Update
Label14.Caption = "Your Password Has Changed Successfully..! Your New Password
Is " + Text8.Text
Check1.Value = 0
Call Form1.tablokapat
End If
Else
MsgBox ("You Have Wrote Different Password Try Again...!")
Text8.Text = ""
Text9.Text = ""
Text8.SetFocus
End If
Else
MsgBox ("You Must Write Your New Password..!")
Text8.Text = ""
Text9.Text = ""
Text8.SetFocus
End If
End Sub
Private Sub Command3_Click()
Check1.Value = 0
Command1.SetFocus
End Sub
Private Sub List18_Click()
Dim ind As Integer
Dim tind As Integer

Private Sub List1_Click()
Call Form1.tabloac
List7.ListIndex = List1.ListIndex
Call temizle
'form2 listelerin doldurulması
Form1.grade.MoveFirst
While Not Form1.grade.EOF
If Form1.grade.Fields("termid") = List7.Text Then
If Form1.grade.Fields("stno") = Text1.Text Then
If IsNull(Form1.grade.Fields("grade")) Then
List6.AddItem " "
Else
List6.AddItem Form1.grade.Fields("grade")
End If
Form1.courseinfo.Index = "primarykey"
Form1.courseinfo.Seek "=", Form1.grade.Fields("courseid")

```



```

If Form1.courseinfo.NoMatch = 0 Then
List2.AddItem UCase(Form1.courseinfo.Fields("coursecode"))
List3.AddItem StrConv(Form1.courseinfo.Fields("coursename"), vbProperCase)
If IsNull(Form1.courseinfo.Fields("credit")) Then
List4.AddItem " "
Else
List4.AddItem Form1.courseinfo.Fields("credit")
End If
End If
Form1.teacher.Index = "primarykey"
Form1.teacher.Seek "=", Form1.grade.Fields("tid")
If Form1.teacher.NoMatch = 0 Then
List5.AddItem StrConv(Form1.teacher.Fields("tname"), vbProperCase) + " " +
UCase(Form1.teacher.Fields("tsurname"))
End If
End If
End If
Form1.grade.MoveNext
Wend
Call Form1.tablokapat
'total kredi ve average hesaplama
Call average(Text1.Text, List7.Text)
Text5.Text = totproduct
Text6.Text = totcredit
'total kredi ve average hesaplama sonu
End Sub

Public Function average(stnumavg As String, termidavg As String)
Dim grade(0 To 100) As Double
Dim credit(0 To 100) As Integer
Dim gr As Double
Dim i As Integer
Dim grades As String
Dim totalproduct As Double
Dim totalcredit As Integer
Call Form1.tabloac
i = 0
Form1.grade.MoveFirst
While Not Form1.grade.EOF
If Form1.grade.Fields("stno") = stnumavg And Form1.grade.Fields("termid") =
termidavg Then
If IsNull(Form1.grade.Fields("grade")) Then
grades = "0"
Else
grades = Form1.grade.Fields("grade")
End If
Select Case grades
Case "AA": gr = 4
Case "BA": gr = 3.5
Case "BB": gr = 3
Case "CB": gr = 2.5

```

```

Case "CC": gr = 2
Case "DC": gr = 1.5
Case "DD": gr = 1
Case "FD": gr = 0.5
Case "FF": gr = 0
Case Else: credit(i) = 0
End Select
Form1.courseinfo.Index = "primarykey"
Form1.courseinfo.Seek "=", Form1.grade.Fields("courseid")
If Form1.courseinfo.NoMatch = 0 Then
If IsNull(Form1.courseinfo.Fields("credit")) Then
credit(i) = 0
Else
credit(i) = Form1.courseinfo.Fields("credit")
End If
Else
credit(i) = 0
End If
grade(i) = gr
i = i + 1
End If
Form1.grade.MoveNext
Wend
totalproduct = 0
totalcredit = 0
For i = 0 To 100
totalproduct = totalproduct + grade(i) * credit(i)
totalcredit = totalcredit + credit(i)
Next
totproduct = Round(totalproduct / totalcredit, 3)
totcredit = totalcredit
Call Form1.tablokat
End Function

Public Function findgpa(stnumgpa As String)
Dim grade(0 To 100) As Double
Dim credit(0 To 100) As Integer
Dim termid(0 To 100) As String
Dim courseid(0 To 100) As String
Dim gr As Double
Dim i As Integer
Dim grades As String
Dim totalcredit As Integer
Dim tgpa As Double
Dim j As Integer
Call Form1.tabloac
gpa = 0
i = 0
Form1.grade.MoveFirst
While Not Form1.grade.EOF
If Form1.grade.Fields("stno") = stnumgpa Then

```



```

courseid(i) = Form1.grade.Fields("courseid")
If IsNull(Form1.grade.Fields("grade")) Then
grades = "0"
Else
grades = Form1.grade.Fields("grade")
End If
Select Case grades
Case "AA": gr = 4
Case "BA": gr = 3.5
Case "BB": gr = 3
Case "CB": gr = 2.5
Case "CC": gr = 2
Case "DC": gr = 1.5
Case "DD": gr = 1
Case "FD": gr = 0.5
Case "FF": gr = 0
Case Else: credit(i) = 0
End Select
Form1.courseinfo.Index = "primarykey"
Form1.courseinfo.Seek "=", Form1.grade.Fields("courseid")
If Form1.courseinfo.NoMatch = 0 Then
If IsNull(Form1.courseinfo.Fields("credit")) Then
credit(i) = 0
Else
credit(i) = Form1.courseinfo.Fields("credit")
End If
Else
credit(i) = 0
'veri eksikliği mesaj box
End If
grade(i) = gr
termid(i) = Form1.grade.Fields("termid")
End If
i = i + 1
Form1.grade.MoveNext
Wend
If List1.ListCount > 1 Then
For i = 0 To 100
For j = 0 To 100
If courseid(i) = courseid(j) Then
If Val(Right(termid(i), 2)) = Val(Right(termid(j), 2)) And Left(termid(i), 1) = "f" And i
< j Then
grade(j) = 0
credit(j) = 0
End If
If Val(Right(termid(i), 2)) > Val(Right(termid(j), 2)) And i < j Then
grade(j) = 0
credit(j) = 0
End If
End If
End If

```

```

Next
tgpa = tgpa + grade(i) * credit(i)
totalcredit = totalcredit + credit(i)
Next
gpa = tgpa / totalcredit
gpa = Round(gpa, 2)
End If
Call Form1.tablokapat
End Function
Private Sub Option1_Click()
Dim i As Integer
Dim tloanamount As Double
Dim tpayamount As Double
If Option1.Value = True Then
Call Form1.tabloac
'loan listelerinin doldurulması
Form1.loan.MoveFirst
While Not Form1.loan.EOF
If Form1.loan.Fields("stno") = Text1.Text Then
List8.AddItem StrConv(Form1.loan.Fields("loaninfo"), vbProperCase)
Form1.term.Index = "primarykey"
Form1.term.Seek "=", Form1.loan.Fields("termid")
If Form1.term.NoMatch = 0 Then
List9.AddItem Form1.term.Fields("termname")
Else
List9.AddItem " "
End If
List10.AddItem Form1.loan.Fields("lastofloan")
List11.AddItem Form1.loan.Fields("taxrate")
List12.AddItem Form1.loan.Fields("loanamount")
For i = 0 To List8.ListCount - 1
tloanamount = tloanamount + Val(List12.List(i))
Next
Text10.Text = Round(tloanamount, 2)
End If
Form1.loan.MoveNext
Wend
'loan listelerinin doldurulması sonu
'payment listelerinin doldurulması
Form1.payment.MoveFirst
While Not Form1.payment.EOF
If Form1.payment.Fields("stno") = Text1.Text Then
List13.AddItem Form1.payment.Fields("paybillno")
Form1.term.Index = "primarykey"
Form1.term.Seek "=", Form1.payment.Fields("termid")
If Form1.term.NoMatch = 0 Then
List14.AddItem Form1.term.Fields("termname")
Else
List14.AddItem " "
End If

```



```

List15.AddItem Form1.payment.Fields("paydate")
List16.AddItem Form1.payment.Fields("paytype")
List17.AddItem Form1.payment.Fields("payamount")
End If
Form1.payment.MoveNext
Wend
For i = 0 To List14.ListCount - 1
tpayamount = tpayamount + Val(List17.List(i))
Next
Text11.Text = tpayamount
'payment listelerinin doldurulması sonu
Else
Frame5.Visible = False
Frame6.Visible = False
End If
Call Form1.tablokapat
End Sub
Private Sub Option2_Click()
Dim control As Integer
control = 0
Call Form1.tabloac
If Option2.Value = True Then
Frame5.Visible = False
Frame6.Visible = False
Frame8.Visible = True
Frame9.Visible = True
Else
Frame5.Visible = True
Frame6.Visible = True
End If
Form1.tablokapat
'seeng student all lectures
If Text1.Text <> "" Then
Call Form1.tabloac
Form1.grade.MoveFirst
While Not Form1.grade.EOF
If Form1.grade.Fields("stno") = Text1.Text Then
Call Form11.courseinfo(Form1.grade.Fields("courseid"))
If IsNull(Form1.grade.Fields("tid")) Then
List21.AddItem ""
Else
Call Form11.findteacher(Form1.grade.Fields("tid"))
End If
List18.AddItem Form11.lastcoursecode
List19.AddItem Form11.lastcoursedes
If Form11.lastcredit = "" Then
List20.AddItem ""
Else
List20.AddItem Form11.lastcredit
End If

```

```

List21.AddItem Form11.tname11 + " " + Form11.tsurname11
If IsNull(Form1.grade.Fields("grade")) Then
List22.AddItem ""
Else
List22.AddItem Form1.grade.Fields("grade")
End If
End If
Form1.grade.MoveNext
Wend
Call Form1.tablokat
End If
'end of seeng all lecture
'lecture will be taken
Call Form11.findstudent(Text1.Text)
Call Form1.tabloac
Form1.courseinfo.MoveFirst
While Not Form1.courseinfo.EOF
If Form1.courseinfo.Fields("deptid") = Form11.stdeptid Then
Form1.grade.MoveFirst
While Not Form1.grade.EOF
If Form1.courseinfo.Fields("courseid") = Form1.grade.Fields("courseid") Then
GoTo jump
Else
control = control + 1
If control = Form1.grade.RecordCount - 1 Then
List23.AddItem Form1.courseinfo.Fields("coursecode")
List24.AddItem Form1.courseinfo.Fields("coursename")
If IsNull(Form1.courseinfo.Fields("credit")) Then
List25.AddItem ""
Else
List25.AddItem Form1.courseinfo.Fields("credit")
End If
End If
End If
Form1.grade.MoveNext
Wend
End If
jump:
Form1.courseinfo.MoveNext
Wend
Call Form1.tablokat
'end of lecture will be taken
End Sub
Private Sub Timer1_Timer()
Label14.Caption = "To Learn Your Lectures In Term Click Term List..."
End Sub
Form3-Source Code
Dim control5 As Integer
Dim findtermid As String
Dim upcontrol As Integer

```



```

Private Sub Command1_Click()
Call Form1.findtermid
Call Form1.tabloac
Form1.advisor.MoveFirst
While Not Form1.advisor.EOF
If Form1.advisor.Fields("tid") = Text8.Text Then
If Form1.advisor.Fields("termid") = Form11.termid11 Then
Frame2.Visible = True
Form1.teacher.Index = "primarykey"
Form1.teacher.Seek "=", Text8.Text
If Form1.teacher.NoMatch = 0 Then
Form3.Text1.Text = StrConv(Form1.teacher.Fields("tname"), vbProperCase)
Form3.Text2.Text = UCase(Form1.teacher.Fields("tsurname"))
Form3.Text3.Text = Form1.teacher.Fields("tusername")
Label13.Caption = "you are advisor for this term"
End If
End If
End If
Form1.advisor.MoveNext
Wend
Form1.teacher.Index = "primarykey"
Form1.teacher.Seek "=", Text8.Text
If Form1.teacher.NoMatch = 0 Then
Form3.Text1.Text = StrConv(Form1.teacher.Fields("tname"), vbProperCase)
Form3.Text2.Text = UCase(Form1.teacher.Fields("tsurname"))
Form3.Text3.Text = Form1.teacher.Fields("tusername")
Form3.Text4.Text = Form1.teacher.Fields("tpassword")
Form3.Text5.Text = Form1.teacher.Fields("taddress")
Form3.Text6.Text = Form1.teacher.Fields("ttel")
Form3.Text7.Text = Form1.teacher.Fields("tmail")
Form3.Text8.Text = Form1.teacher.Fields("tid")
End If
Call Form1.tablokat
End Sub
Private Sub Command2_Click()
Form1.Show
Form1.Combo1.SetFocus
Unload Me
End Sub
Private Sub Command4_Click()
If Text3.Text <> "" And Text4.Text <> "" Then
Call Form1.tabloac
Form1.teacher.Index = "primarykey"
Form1.teacher.Seek "=", Trim(Text8.Text)
If Form1.teacher.NoMatch = 0 Then
Form1.teacher.Edit
Form1.teacher.Fields("tusername") = Text3.Text
Form1.teacher.Fields("tpassword") = Trim(Text4.Text)
upcontrol = MsgBox("are ou sure to update?", vbOKCancel)
If upcontrol = 1 Then

```

```

Private Sub Command6_Click()
If Text9.Text <> "" And Text10.Text <> "" Then
Call Form1.tabloac
Form1.teacher.MoveFirst
While Not Form1.teacher.EOF
If Form1.teacher.Fields("tusername") = Text9.Text Then
If Form1.teacher.Fields("tpassword") = Text10.Text Then
Form5.Show
Form3.Hide
control5 = 1
If control5 = 1 Then
Form1.teacher.MoveNext
End If
Else
MsgBox ("your password is wrong")
End If
Else
MsgBox ("user name is wrong")
End If
Form1.teacher.MoveNext
Wend
End If
Call Form1.tablokapat
End Sub
Private Sub Command7_Click()
Text9.Text = ""
Text10.Text = ""
Frame2.Visible = False
End Sub
Private Sub Form_Load()
'MnStudentRegister.Enabled = False
End Sub
Private Sub Label10_Click()
End Sub
Private Sub MnAllLecture_Click()
Form13.Show
Form3.Hide
End Sub
Private Sub MnLookYourDetails_Click()
Call Form1.tabloac
Form1.advisor.MoveFirst
While Not Form1.advisor.EOF
If Form1.advisor.Fields("tid") = Text8.Text Then
Form1.tidcontrol = 1
Frame2.Visible = True
Label13.Caption = "you are advisor for that term...!"
Text9.Text = Form1.teacher.Fields("tusername")
Text10.Text = Form1.teacher.Fields("tpassword")
If Form1.tidcontrol = 1 Then
Exit Sub

```



```

End If
End If
Form1.advisor.MoveNext
Wend
Form1.tablokapat
End Sub
Private Sub MnNewRegister_Click()
Form4-Source Code
Dim control4 As Integer
Dim control41 As Integer
Private Sub Command1_Click()
Command3.Visible = True
Command4.Visible = True
'search by number
control4 = 0
If Text6.Text <> "" Then
    Call Form1.tabloac
    Form1.student.Index = "primarykey"
    Form1.student.Seek "=", Text6.Text
    If Form1.student.NoMatch = 0 Then
        Text1.Text = Form1.student.Fields("stno")
        Text2.Text = Form1.student.Fields("stname")
        Text3.Text = Form1.student.Fields("stsurname")
        Text5.Text = Form1.student.Fields("deptid")
        Form1.department.MoveFirst
        While Not Form1.department.EOF
            If Form1.department.Fields("deptid") = Text5.Text Then
                Text5.Text = Form1.department.Fields("deptname")
                Text4.Text = Form1.department.Fields("faculty")
                control4 = 1
            End If
            If control4 = 1 Then
                Exit Sub
            End If
            Form1.department.MoveNext
        Wend
    Else
        MsgBox ("student was not found...!")
        Text6.Text = ""
        Text6.SetFocus
    End If
Else
End If
Call Form1.tablokapat
End Sub
Private Sub Command2_Click()
Command3.Visible = True
Command4.Visible = True
control4 = 0
'search by name

```

```

If Text7.Text <> "" Then
    Call Form1.tabloac
    Form1.student.MoveFirst
    While Not Form1.student.EOF
        If LCase(Form1.student.Fields("stname")) = LCase(Trim(Text7.Text)) Then
            Text1.Text = Form1.student.Fields("stno")
            Text6.Text = Form1.student.Fields("stno")
            Text2.Text = Form1.student.Fields("stname")
            Text3.Text = Form1.student.Fields("stsurname")
            Text5.Text = Form1.student.Fields("deptid")
            Form1.department.MoveFirst
            While Not Form1.department.EOF
                If Form1.department.Fields("deptid") = Text5.Text Then
                    Text5.Text = Form1.department.Fields("deptname")
                    Text4.Text = Form1.department.Fields("faculty")
                    control4 = 1
                End If
                If control4 = 1 Then
                    Call Form1.tablokat
                    Exit Sub
                End If
            Form1.department.MoveNext
        Wend
    Else
        MsgBox ("student name was not found....!!")
        Call Form1.tablokat
        Text7.Text = ""
        Text7.SetFocus
        Exit Sub
    End If
    Form1.student.MoveNext
Wend
Else
    Call Form1.tablokat
End If
End Sub
Private Sub Command3_Click()
    control4 = 0
    Frame4.Visible = True
    'student texts is filled with command3 button
    If Text1.Text <> "" Then
        Call Form1.tabloac
        Form1.student.Index = "primarykey"
        Form1.student.Seek "=", Text6.Text
        If Form1.student.NoMatch = 0 Then
            Text8.Text = Form1.student.Fields("stgender")
            If Text8.Text = "m" Then
                Text8.Text = "Male"
            Else
                Text8.Text = "Female"
            End If
        End If
    End If

```



```

End If
control4 = 1
End If
If control4 = 1 Then
Call Form1.tablokapat
Exit Sub
End If
Else
MsgBox ("record not found")
End If
Call Form1.tablokapat
End Sub

Private Sub Command4_Click()
Frame4.Visible = False
End Sub
Private Sub Form_Load()
Command3.Visible = False
Command4.Visible = False
End Sub
Private Sub Form_Unload(Cancel As Integer)
Form3.Show
Unload Me
End Sub
Private Sub Option1_Click()
If Option1.Value = True Then
Frame3.Visible = True
Text7.Text = ""
Text7.Enabled = False
Text6.Enabled = True
Text6.SetFocus
Command2.Enabled = False
Command1.Enabled = True
End If
End Sub
Private Sub Option2_Click()
If Option2.Value = True Then
Frame3.Visible = True
Text6.Text = ""
Text6.Enabled = False
Text7.Enabled = True
Text7.SetFocus
Command1.Enabled = False
Command2.Enabled = True
End If
End Sub
Form5-Source Code
Dim departmentcontrol As String
Dim courseidcontrol As String
Dim deptidcontrol As String

```

```

Dim courseidcontrol2 As String
Public jump2
Dim stnocontrol As String
Dim controlterm As String
Dim termcon As Integer
Dim listcontrol As Integer
Dim registercontrol As Integer
Private Sub Command1_Click()
If Text1.Text <> "" Then
stnocontrol = Text1.Text
Call Form1.tabloac
Form1.student.Index = "primarykey"
Form1.student.Seek "=", Text1.Text
If Form1.student.NoMatch = 0 Then
departmentcontrol = Form1.student.Fields("deptid")
Form1.department.Index = "primarykey"
Form1.department.Seek "=", departmentcontrol
If Form1.department.NoMatch = 0 Then
Text2.Text = Form1.student.Fields("stname")
Text3.Text = Form1.student.Fields("stsurname")
Text4.Text = Form1.department.Fields("deptname")
Form1.grade.MoveFirst
While Not Form1.grade.EOF
If Form1.grade.Fields("stno") = Text1.Text Then
courseidcontrol = Form1.grade.Fields("courseid")
controltid = Form1.grade.Fields("tid")
If IsNull(Form1.grade.Fields("grade")) Then
List9.AddItem ""
Else
List9.AddItem Form1.grade.Fields("grade")
End If
Form1.courseinfo.Index = "primarykey"
Form1.courseinfo.Seek "=", courseidcontrol
If Form1.courseinfo.NoMatch = 0 Then
List6.AddItem Form1.courseinfo.Fields("coursecode")
List7.AddItem Form1.courseinfo.Fields("coursename")
If IsNull(Form1.courseinfo.Fields("credit")) Then
List8.AddItem ""
Else
List8.AddItem Form1.courseinfo.Fields("credit")
End If
List10.AddItem Form1.courseinfo.Fields("coursetype")
Form1.teacher.Index = "primarykey"
Form1.teacher.Seek "=", controltid
Command2.Enabled = True
If Form1.teacher.NoMatch = 0 Then
List11.AddItem StrConv(Form1.teacher.Fields("tname"), vbProperCase) + "
" + StrConv(Form1.teacher.Fields("tsurname"), vbUpperCase)
Else
List11.AddItem ""

```

```

        End If
        Else
        'MsgBox ("course not found")
        End If
Else
    'MsgBox ("now there is no eny lecture")
    End If
    Form1.grade.MoveNext
    Wend
Else
    MsgBox ("department was not found")
    End If
Else
    MsgBox ("Record not Found")
    End If
Else
    MsgBox ("Enter Student Number.....!!")
    End If
'Call Form11.findtermid
'Call Form2.average(Text1.Text, Form11.termid11)
'Text5.Text = Form2.totproduct
End Sub
Private Sub Command2_Click()
List1.Clear
List2.Clear
List3.Clear
List4.Clear
List5.Clear
counting = 0
If Text1.Text <> "" Then
If Text2.Text <> "" And Text3.Text <> "" Then
Call Form1.tabloac
    Frame2.Enabled = True
    'finding department ID
    Form1.student.Index = "primarykey"
    Form1.student.Seek "=", Text1.Text
    If Form1.student.NoMatch = 0 Then
    deptidcontrol = Form1.student.Fields("deptid")
    Form1.courseinfo.MoveFirst
    While Not Form1.courseinfo.EOF
        If Form1.courseinfo.Fields("deptid") = deptidcontrol Then
            courseidcontrol2 = Form1.courseinfo.Fields("courseid")
            counting = 0
            Form1.grade.MoveFirst
            'For counting = 1 To Form1.grade.RecordCount
            While Not Form1.grade.EOF
                If Form1.grade.Fields("courseid") = courseidcontrol2 Then
                    If IsNull(Form1.grade.Fields("grade")) Then GoTo jump
                    If UCase(Form1.grade.Fields("grade")) <> "FF" Then GoTo jump
                    List1.AddItem Form1.courseinfo.Fields("coursecode")

```



```

List2.AddItem Form1.courseinfo.Fields("coursename")
If IsNull(Form1.courseinfo.Fields("credit")) Then
List3.AddItem ""
Else
List3.AddItem Form1.courseinfo.Fields("credit")
End If
List4.AddItem Form1.courseinfo.Fields("coursetype")
List5.AddItem Form1.courseinfo.Fields("prereq")
Else
    counting = counting + 1
    If counting = Form1.grade.RecordCount Then
        List1.AddItem Form1.courseinfo.Fields("coursecode")
        List2.AddItem Form1.courseinfo.Fields("coursename")
        If IsNull(Form1.courseinfo.Fields("credit")) Then
            List3.AddItem ""
        Else
            List3.AddItem Form1.courseinfo.Fields("credit")
        End If
        List4.AddItem Form1.courseinfo.Fields("coursetype")
        If IsNull(Form1.courseinfo.Fields("prereq")) Then
            List5.AddItem ""
        Else
            List5.AddItem Form1.courseinfo.Fields("prereq")
        End If
    Else: GoTo jump2
    End If
End If

```

Jump2:

```

    Form1.grade.MoveNext
Wend
Next
End If
jump:
    Form1.courseinfo.MoveNext
Wend
Else
    MsgBox ("student was not found")
End If
Call Form1.tablokat
Else
    MsgBox ("enter student information....!!")
End If
Else
    MsgBox ("you have to enter student number")
End If
'Call Form1.tablokat
MsgBox ("to do course registration select courses from list...!!!")
List1.SetFocus
End Sub

```

```

Private Sub Command3_Click()
If List12.ListIndex <> -1 Then
List12.RemoveItem List12.ListIndex
List13.RemoveItem List13.ListIndex
Else
MsgBox ("select from list")
End If
End Sub
Private Sub Command4_Click()
termcon = 0
    Call Form1.tabloac
    Form1.term.MoveFirst
    While Not Form1.term.EOF
    If IsNull(Form1.term.Fields("termstart")) Then GoTo state
    If Date > CDate(Form1.term.Fields("termstart")) Then
    If IsNull(Form1.term.Fields("termend")) Then GoTo state
    If Date < CDate(Form1.term.Fields("termend")) Then
        controlterm = Form1.term.Fields("termid")
    End If
    End If
state:
        Form1.term.MoveNext
    Wend
    Call Form1.tablokapat
If List12.ListCount <> 0 Then
    Call Form1.tabloac
    Form1.student.Index = "primarykey"
    Form1.student.Seek "=", stnocontrol
    If Form1.student.NoMatch = 0 Then
        deptidcontrol = Form1.student.Fields("deptid")
        For listcontrol = 0 To List1.ListCount - 1
            Form1.courseinfo.MoveFirst
            While Not Form1.courseinfo.EOF
                'okunuyor
                If Form1.courseinfo.Fields("coursecode") = Trim(List12.List(listcontrol))
Then
                    If Form1.courseinfo.Fields("deptid") = deptidcontrol Then
                        'controllin course in grade table
                        Form1.grade.MoveFirst
                        While Not Form1.grade.EOF
                            If Form1.courseinfo.Fields("courseid") = Form1.grade.Fields("courseid")
Then
                                If Form1.grade.Fields("termid") = controlterm Then
                                    MsgBox ("this course already was include" + " " +
Form1.courseinfo.Fields("coursecode"))
                                    Call Form1.tablokapat
                                    Call clearlist
                                    Exit Sub
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If
End If

```

```

Form1.grade.MoveNext
Wend
'end of controlling course in grade
Form1.grade.AddNew
Form1.grade.Fields("courseid") = Form1.courseinfo.Fields("courseid")
Form1.grade.Fields("credit") = Form1.courseinfo.Fields("credit")
Form1.grade.Fields("tid") = Form1.courseinfo.Fields("tid")
Form1.grade.Fields("stno") = stnocontrol
Form1.grade.Fields("termid") = controlterm
registercontrol = MsgBox("will be done", vbYesNo)
If registercontrol = 6 Then
On Error GoTo mesaj
Form1.grade.Update
MsgBox ("register is successfull....!!!!")
Else
Call Form1.tablokatap
Exit Sub
End
End If
End If
Form1.courseinfo.MoveNext
Wend
Next
Else
MsgBox (stnocontrol + " didnt find")
End If
mesaj:
If Err = 3022 Then
MsgBox ("you can not enter course two times in same term")
Call clearlist
End If
Call Form1.tablokatap
Else
MsgBox ("select lecture from list" + " " + Frame2.Caption + "")
End If
'List1.TopIndex = List2.TopIndex
'List1.NewIndex = List2.NewIndex
Label17.Caption = Date
Frame2.Enabled = False
Frame4.Enabled = False
Call Form1.l.findtermid
Label20.Caption = Form1.l.termname11
End Sub
Private Sub Form_Unload(Cancel As Integer)
Form5.Hide
Form3.Show
Text1.Text = ""
Text2.Text = ""
End Sub
Dim control6 As Integer

```



```

Dim courseidcontrol As Integer
Dim courseinfocontrol As String
Dim listcontrol As Integer
Dim coursecodecontrol As String
Dim pcontrol As Integer
Private Sub Command1_Click()
If Text3.Text <> "" And Text4.Text <> "" Then
Call Form1.tabloac
Form1.personal.Index = "primarykey"
Form1.personal.Seek "=", Text11.Text
If Form1.personal.NoMatch = 0 Then
Form1.personal.Edit
Form1.personal.Fields("pusername") = Text3.Text
Form1.personal.Fields("ppassword") = Text4.Text
pcontrol = MsgBox("are you sure to update?", vbOKCancel)
If pcontrol = 1 Then
Form1.personal.Update
MsgBox ("updating is successfull new: " + Text4.Text)
Frame1.Visible = False
Else
Text3.Text = ""
Text4.Text = ""
Frame1.Visible = False
End If
Else
MsgBox ("you can not chang password")
End If
Call Form1.tablokapat
Else
MsgBox ("enter yuur password and user name.....!!!!")
End If
End Sub
Private Sub Command2_Click()
Text3.Text = ""
Text4.Text = ""
Frame1.Visible = False
End Sub
Private Sub Command3_Click()
control6 = 0
'finding student and writing information
If Text5.Text <> "" Then
Call Form1.tabloac
Form1.student.Index = "primarykey"
Form1.student.Seek "=", Text5.Text
If Form1.student.NoMatch = 0 Then
Text6.Text = StrConv(Form1.student.Fields("stname"), vbProperCase)
Text7.Text = StrConv(Form1.student.Fields("stsurname"), vbProperCase)
Text8.Text = Form1.student.Fields("deptid")
Form1.department.Index = "primarykey"
Form1.department.Seek "=", Text8.Text

```

```

If Form1.department.NoMatch = 0 Then
    Text8.Text = StrConv(Form1.department.Fields("deptname"), vbProperCase)
    control6 = 1
End If
If control6 = 1 Then
    Call Form1.tablokapat
Exit Sub
End If
End If
Call Form1.tablokapat
Else
    MsgBox ("Enter student number")
End If
End Sub
Private Sub Command4_Click()
    courseidcontrol = 0
    listcontrol = 0
    Call Form11.findtermid
    If Text5.Text <> "" And Text6.Text <> "" Then
        Call Form1.tabloac
        ' filling list1
        Form1.grade.MoveFirst
        While Not Form1.grade.EOF
            If Form1.grade.Fields("stno") = Text5.Text Then
                If IsNull(Form1.grade.Fields("grade")) Then
                    If Form1.grade.Fields("termid") = Form11.termid11 Then
                        Label14.Caption = Form11.termname11
                        courseinfocontrol = Form1.grade.Fields("courseid")
                        Form1.courseinfo.Index = "primarykey"
                        Form1.courseinfo.Seek "=", courseinfocontrol
                        If Form1.courseinfo.NoMatch = 0 Then
                            listcontrol = listcontrol + 1
                            List1.AddItem Form1.courseinfo.Fields("coursecode")
                        End If
                    End If
                End If
            End If
        End If
        Form1.grade.MoveNext
    Wend
    Command4.Enabled = False
    Call Form1.tablokapat
    'end of filling list1
Else
    MsgBox ("firstly write student number and find...!")
End If
End Sub
Private Sub Command5_Click()
    If Text5.Text <> "" Then
        If Text9.Text <> "" Then
            If Text10.Text <> "" Then

```

```

Call Form11.findstudent(Text5.Text)
Call Form1.tabloac
Call Form1.courseinfo.MoveFirst
While Not Form1.courseinfo.EOF
'Text12.Text = Form11.stdeptid
If Form1.courseinfo.Fields("coursecode") = Text9.Text And
Form1.courseinfo.Fields("deptid") = Form11.stdeptid Then
Form1.grade.MoveFirst
'MsgBox ("")
While Not Form1.grade.EOF
If Form1.grade.Fields("courseid") = Form1.courseinfo.Fields("courseid") Then
Form1.grade.Edit
Form1.grade.Fields("grade") = Text10.Text
MsgBox ("grade was entered successfully....!")
Form1.grade.Update
End If
Private Sub Command6_Click()
Private Sub Form_Unload(Cancel As Integer)
Form6.Hide
Form1.Show
Form1.Text1.Text = ""
Form1.Text2.Text = ""
Form1.Combo1.SetFocus
End Sub
Private Sub List1_Click()
Text9.Text = Trim(List1.Text)
Text10.SetFocus
End Sub
Private Sub MnAddAdvisor_Click()
Form7.Show
Form6.Hide
End Sub
Private Sub MnEentergrade_Click()
Frame2.Visible = True
Text5.SetFocus
End Sub
Private Sub mnenterdisact_Click()
Form9.Show
Form6.Hide
End Sub
Private Sub Mnhide_Click()
Frame1.Visible = False
End Sub
Private Sub mnopencourses_Click()
Form14.Show
Form6.Hide
End Sub
Private Sub mnopenterm_Click()
Form8.Show
Form6.Hide

```



```

End Sub
Private Sub mnupdatemydata_Click()
Frame1.Visible = True
Call Form1.tabloac
Form1.personal.Index = "primarykey"
Form1.personal.Seek "=", Text11.Text
If Form1.personal.NoMatch = 0 Then
Text3.Text = Form1.personal.Fields("pusername")
Text4.Text = Form1.personal.Fields("ppassword")
End If
Call Form1.tablokat
End Sub
Private Sub Text5_Change()
List1.Clear
Text9.Text = ""
Text10.Text = ""
Command4.Enabled = True
End Sub
Form7-Source Code
Public controltid As String
Public controldeptid As String
Public controltermid As String
Public stcon As Integer
Dim countdeptid As Integer
Dim namecontrol As String
Dim i As Integer
Dim snamecontrol As Integer
Dim snamecontrol2 As String
Dim findname As String
Dim findsname As String
Dim findnamecontrol As Integer
Public controlst As Double
Dim deptidcontrol As String
Dim termidcontrol As String
Dim studentcontrol As Double
Dim cntrterm As String
Public deptcontrol As String
Private Sub Combo1_Click()
List1.Clear
Call Form1.tabloac
Form1.department.MoveFirst
While Not Form1.department.EOF
If LCase(Form1.department.Fields("deptname")) = LCase(Combo1.Text) Then
Text13.Text = Form1.department.Fields("deptid")
Form1.teacher.MoveFirst
While Not Form1.teacher.EOF
If Form1.teacher.Fields("deptid") = Text13.Text Then
List1.AddItem StrConv(Form1.teacher.Fields("tname"), vbProperCase) + " " +
UCase(Form1.teacher.Fields("tsurname"))
Form1.teacher.MoveFirst

```

```

While Not Form1.teacher.EOF
If Form1.teacher.Fields("deptid") = LCase(Text13.Text) Then
List1.AddItem StrConv(Form1.teacher.Fields("tname"), vbProperCase) + " " +
UCase(Form1.teacher.Fields("tsurname"))
countdeptid = 1
controldeptid = Text13.Text
'Else
'MsgBox ("you wrote wong term id try again")
'Text13.Text = ""
'Text13.SetFocus
'Call Form1.tablokat
'Exit Sub
End If
Form1.teacher.MoveNext
Wend
If countdeptid <> 1 Then
MsgBox ("there is no deptid in table chang deptid and try again...!")
Text13.Text = ""
Text13.SetFocus
Call Form1.tablokat
Exit Sub
End If
Command1.Enabled = False
'filling dept name
Form1.department.Index = "primarykey"
Form1.department.Seek "=", controldeptid
If Form1.department.NoMatch = 0 Then
Text9.Text = Form1.department.Fields("deptname")
End If
Call Form1.tablokat
List1.SetFocus
'Else
'MsgBox ("")
'Else
'MsgBox ("you mast enter department id....!")
'Text1.SetFocus
End If
End Sub
Private Sub Command3_Click()
Text12.Text = ""
Text13.Text = ""
Form7.SetFocus
End Sub
Private Sub Command4_Click()
Dim control As Integer
Dim cnt As Long
If Text1.Text <> "" Then
If Text2.Text <> "" Then
If Text3.Text <> "" Then
If Text4.Text <> "" And Text5.Text <> "" Then

```

```

Call Form11.findtermid
If Val(Text4.Text) < Val(Text5.Text) Then
    Call Form1.tabloac
    For cnt = Val(Text4.Text) To Val(Text5.Text)
        Form1.student.Index = "primarykey"
        Form1.student.Seek "=", Trim(Str(cnt))
        If Form1.student.NoMatch = 0 Then
            If Form1.student.Fields("deptid") = Text13.Text Then
                Form1.advisor.MoveFirst
                While Not Form1.advisor.EOF
                    If Form1.advisor.Fields("stno") = Trim(Str(cnt)) And
Form1.advisor.Fields("termid") = Form11.termid11 The
                        MsgBox ("you can not enter two times")
                    Else
                        control = control + 1
                        If control = Form1.advisor.RecordCount Then
                            Form1.advisor.AddNew
                            Form1.advisor.Fields("tid") = Text3.Text
                            Form1.advisor.Fields("stno") = Trim(Str(cnt))
                            Form1.advisor.Fields("termid") = Form11.termid11
                            On Error Resume Next
                            Form1.advisor.Update
                        End If
                    End If
                    Form1.advisor.MoveNext
                Wend
            End If
        End If
    Next
    Call Form1.tablokapat
    Else
        Call Form1.tabloac
        For cnt = Val(Text5.Text) To Val(Text4.Text)
            Form1.student.Index = "primarykey"
            Form1.student.Seek "=", Trim(Str(cnt))
            If Form1.student.NoMatch = 0 Then
                If Form1.student.Fields("deptid") = Text13.Text Then
                    Form1.advisor.MoveFirst
                    While Not Form1.advisor.EOF
                        If Form1.advisor.Fields("stno") = Trim(Str(cnt)) And
Form1.advisor.Fields("termid") = Form11.termid11
                            MsgBox ("you can not enter two times")
                        Else
                            control = control + 1
                            If control = Form1.advisor.RecordCount Then
                                Form1.advisor.AddNew
                                Form1.advisor.Fields("tid") = Text3.Text
                                Form1.advisor.Fields("stno") = Trim(Str(cnt))
                                Form1.advisor.Fields("termid") = Form11.termid11
                                On Error Resume Next

```



```

        Form1.advisor.Update
    End If
End If
Form1.advisor.MoveNext
Wend
End If
End If
Next
Call Form1.tablokapat
End If
Private Sub Command5_Click()
If Text6.Text <> "" Then
Call Form1.tabloac
Form1.teacher.Index = "primaryKey"
Form1.teacher.Seek "=", Text6.Text
If Form1.teacher.NoMatch = 0 Then
Text7.Text = Form1.teacher.Fields("tname")
Text8.Text = Form1.teacher.Fields("tsurname")
deptcontrol = Form1.teacher.Fields("deptid")
Form1.department.Index = "primaryKey"
Form1.department.Seek "=", deptcontrol
If Form1.department.NoMatch = 0 Then
Text10.Text = Form1.department.Fields("deptname")
Private Sub Command6_Click()
Dim cnt As Long
Dim control As Integer
If Text6.Text <> "" Then
If Text11.Text <> "" And Text12.Text <> "" Then
Call Form11.findtermid
If Val(Text11.Text) < Val(Text12.Text) Then
Call Form1.tabloac
For cnt = Val(Text11.Text) To Val(Text12.Text)
Form1.student.Index = "primaryKey"
Form1.student.Seek "=", Trim(Str(cnt))
If Form1.student.NoMatch = 0 Then
If Form1.student.Fields("deptid") = deptcontrol Then
Form1.advisor.MoveFirst
While Not Form1.advisor.EOF
If Form1.advisor.Fields("stno") = Trim(Str(cnt)) And
Form1.advisor.Fields("termid") = Form11.termid11 Then
MsgBox ("you can not enter two times")
Else
control = control + 1
If control = Form1.advisor.RecordCount Then
Form1.advisor.AddNew
Form1.advisor.Fields("tid") = Text6.Text
Form1.advisor.Fields("stno") = Trim(Str(cnt))
Form1.advisor.Fields("termid") = Form11.termid11
On Error Resume Next
Form1.advisor.Update

```

```

        End If
    End If
    Form1.advisor.MoveNext
Wend
End If
End If
Next
Call Form1.tablokapat
Else
    Call Form1.tabloac
    For cnt = Val(Text12.Text) To Val(Text11.Text)
Form1.student.Index = "primarykey"
        Form1.student.Seek "=", Trim(Str(cnt))
        If Form1.student.NoMatch = 0 Then
            If Form1.student.Fields("deptid") = deptcontrol Then
                Form1.advisor.MoveFirst
                While Not Form1.advisor.EOF
                    If Form1.advisor.Fields("stno") = Trim(Str(cnt)) And
Form1.advisor.Fields("termid") = Form11.termid11 Then
                        MsgBox ("you can not enter two times")
                    Else
                        control = control + 1
                        If control = Form1.advisor.RecordCount Then
                            Form1.advisor.AddNew
                            Form1.advisor.Fields("tid") = Text6.Text
                            Form1.advisor.Fields("stno") = Trim(Str(cnt))
                            Form1.advisor.Fields("termid") = Form11.termid11
                            On Error Resume Next
                            Form1.advisor.Update
                        End If
                    End If
                    Form1.advisor.MoveNext
                Wend
            End If
        End If
    Next
    Call Form1.tablokapat
End If
Else
    MsgBox ("enter student number between")
End If
Else
    MsgBox ("enter teacher ID...!")
End If
End Sub
Private Sub clrtext()
End Sub
Private Sub Command9_Click()
countdeptid = 0

```

```

If Text1.Text <> "" And Text2.Text <> "" Then
Call Form1.tabloac
'finding teacher with name
Form1.teacher.MoveFirst
While Not Form1.teacher.EOF
If Form1.teacher.Fields("tname") = LCase(Text1.Text) Then
If Form1.teacher.Fields("tsurname") = LCase(Text2.Text) Then
Text3.Text = Form1.teacher.Fields("tid")
controltid = Text3.Text
Text13.Text = Form1.teacher.Fields("deptid")
controldeptid = Form1.teacher.Fields("deptid")
    Form1.department.Index = "primarykey"
    Form1.department.Seek "=", controldeptid
    If Form1.department.NoMatch = 0 Then
        Text9.Text = Form1.department.Fields("deptname")
        countdeptid = 1
Frame3.Enabled = False
Call Form1.tabloac
Form1.department.MoveFirst
While Not Form1.department.EOF
Combo1.AddItem Form1.department.Fields("deptname")
Form1.department.MoveNext
Wend
Call Form1.tablokapat
Combo1.ListIndex = 0
End Sub
Private Sub Form_Unload(Cancel As Integer)
Form7.Hide
Form6.Show
End Sub
Private Sub List1_Click()
findnamecontrol = 0
'For i = 1 To Len(List1.Text)
namecontrol = InStr(List1.Text, " ")
'If namecontrol = "" Then
'Text1.Text = Left(List1.Text, i)
'snamecontrol = Len(List1.Text) - i
'snamecontrol2 = Right(List1.Text, snamecontrol)
'Text1.Text = Trim(snamecontrol2)
'End If
Text1.Text = Left(List1.Text, namecontrol)
Text1.Text = Trim(Text1.Text)
snamecontrol = Len(List1.Text) - namecontrol
snamecontrol2 = Right(List1.Text, snamecontrol)
Text2.Text = Trim(snamecontrol2)
'finding teacher id
findname = LCase(Text1.Text)
findsname = LCase(Text2.Text)
Call Form1.tabloac
Form1.teacher.MoveFirst

```



```

While Not Form1.teacher.EOF
If Form1.teacher.Fields("tname") = findname Then
If Form1.teacher.Fields("tsurname") = findsname Then
Text3.Text = Form1.teacher.Fields("tid")
controltid = Text3.Text
controldeptid = Text13.Text
findnamecontrol = 1
MsgBox ("enter student number between and hit add advisor button")
Private Sub Option1_Click()
If Option1.Value = True Then
Text13.SetFocus
Command9.Enabled = False
Command1.Enabled = True
MsgBox ("enter department ID")
End If
End Sub
End Sub
Form8-Source Code
Dim control8 As Integer
Private Sub Command1_Click()
If Text1.Text <> "" Then
Call Form1.tabloac
Form1.term.Index = "primarykey"
Form1.term.Seek "=", Text1.Text
If Form1.term.NoMatch = 0 Then
MsgBox ("you can not write same termid")
Text1.Text = ""
Text1.SetFocus
Call Form1.tablokapat
Exit Sub
End If
If Text2.Text <> "" Then
If Text3.Text <> "" Then
If Text4.Text <> "" Then
'opening term is starting
Form1.term.MoveFirst
While Not Form1.term.EOF
If Date > CDate(Form1.term.Fields("termstart")) Then
If Date < CDate(Form1.term.Fields("termend")) Then
MsgBox ("term already was opened you can not open two times")
Call Form1.tablokapat
Form1.term.MoveNext
Wend
control8 = MsgBox("term will be added check data correct is not", vbYesNo)
If control8 = 6 Then
Form1.term.AddNew
Form1.term.Fields("termid") = Text1.Text
Form1.term.Fields("termname") = Text2.Text
Form1.term.Fields("termstart") = Text3.Text
Form1.term.Fields("termend") = Text4.Text

```

```

Form1.term.Update
MsgBox ("new term opened successfully")
Label1.Caption = "NEW TERM OPENED"
Label1.FontBold = True
Label1.FontSize = 28
Label1.FontItalic = True
Text4.Text = ""
Text1.SetFocus
End Sub
Private Sub Command3_Click()
Dim msgcontrol As Integer
If Text1.Text <> "" Then
Call Form1.tabloac
Form1.term.Index = "primarykey"
Form1.term.Seek "=", Text1.Text
If Form1.term.NoMatch = 0 Then
Form1.term.Edit
Form1.term.Fields("termid") = Text1.Text
Form1.term.Fields("termname") = Text2.Text
Form1.term.Fields("termstart") = Text3.Text
Form1.term.Fields("termend") = Text4.Text
msgcontrol = MsgBox("everything will be change", 4 + 32, "Attention")
If msgcontrol = 6 Then
Form1.term.Update
MsgBox ("updating has been successfull")
Form1.ForeColor = vbGreen
'MsgBox ("Please Enter All Data To The Texts Fully")
Label6.Caption = "Example For Termid: s01"
Call Form1.tabloac
Form1.term.MoveFirst
While Not Form1.term.EOF
List1.AddItem Form1.term.Fields("termname")
Form1.term.MoveNext
Wend
Call Form1.tablokapat
End Sub
Private Sub Form_Unload(Cancel As Integer)
Form6.Show
Form8.Hide
End Sub
Private Sub List1_Click()
Call Form1.tabloac
Form1.term.MoveFirst
While Not Form1.term.EOF
If LCase(Form1.term.Fields("termname")) = LCase(List1.Text) Then
Text1.Text = Form1.term.Fields("termid")
Text2.Text = Form1.term.Fields("termname")
Text3.Text = Form1.term.Fields("termstart")
Text4.Text = Form1.term.Fields("termend")
If Text4.Text <> "" Then

```

```

Call Form11.findtermid
Call Form1.tabloac
Form1.disact.MoveFirst
While Not Form1.disact.EOF
If Form1.disact.Fields("disactid") = Text1.Text Then
MsgBox ("you can not give same disactid two times")
Else
count = count + 1
If count = Form1.disact.RecordCount Then
Form1.disact.AddNew
Form1.disact.Fields("disactid") = Text1.Text
Form1.disact.Fields("stno") = Text2.Text
Form1.disact.Fields("disactinfo") = Text3.Text
Form1.disact.Fields("disactdate") = Date
Form1.disact.Fields("termid") = Form11.termid11
Form1.disact.Fields("disactresult") = Text4.Text
MsgBox ("disact will be added")
Form1.disact.Update
a = MsgBox("Disact has been added successfully...!", 64)
Call Form1.tablokat
Private Sub clrtext()
Private Sub Command2_Click()
Call clrtext
Call clrlist
Form9.SetFocus
End Sub
Private Sub Command3_Click()
Dim msgcontrol As Integer
msgcontrol = MsgBox("are you sure to delete?", 36, "will be deleted")
If msgcontrol = 6 Then
If List1.ListIndex <> -1 Then
Call Form1.tabloac
Form1.disact.MoveFirst
While Not Form1.disact.EOF
If Form1.disact.Fields("stno") = List1.Text And Form1.disact.Fields("disactdate") =
List3.Text Then
Form1.disact.Delete
If Form1.disact.Fields("stno") = List1.Text And Form1.disact.Fields("disactdate") =
List3.Text Then
Text5.Text = Form1.disact.Fields("disactid")
End If
Call Form11.findtermname(controldeptid)
List4.AddItem Form11.findtermname11
Form1.disact.MoveNext
Wend
End If
End Sub
Dim deptid14 As String
Dim cid14 As String
Private Sub Combo1_Change()

```



```

Dim coursecode14 As String
Dim tid14 As String
Dim deptid14 As String
Call clearlist
Call Form1.tabloac
Form1.department.MoveFirst
While Not Form1.department.EOF
MsgBox ("")
If LCase(Form1.department.Fields("deptname")) = LCase(Combo1.Text) Then
deptid14 = Form1.department.Fields("deptid")
'starting of finding course
Form1.courseinfo.MoveFirst
While Not Form1.courseinfo.EOF
If Form1.courseinfo.Fields("deptid") = deptid14 Then
coursecode14 = Form1.courseinfo.Fields("coursecode")
tid14 = Form1.courseinfo.Fields("tid")
Call Form1.tablokat
Call Form11.findteacher(tid14)
Call Form1.tabloac
List1.AddItem StrConv(coursecode14, vbProperCase)
List2.AddItem UCase(Form11.tname11 + " " + Form11.tsurname11)
End If Form1.courseinfo.MoveNext
Wend
End If
'end of finding courses
Form1.department.MoveNext
Wend
Call Form1.tablokat
End Sub
Private Sub Command1_Click()
Dim coursecode14 As String
Dim tid14 As String
Call clearlist
'filling all lecture
Call Form1.tabloac
Form1.department.MoveFirst
While Not Form1.department.EOF
If LCase(Form1.department.Fields("deptname")) = LCase(Combo1.Text) Then
deptid14 = Form1.department.Fields("deptid")
Text1.Text = deptid14
Form1.courseinfo.MoveFirst
While Not Form1.courseinfo.EOF
If Form1.courseinfo.Fields("deptid") = deptid14 Then
sql14 = ""
If IsNull(Form1.courseinfo.Fields("tid")) Then
List1.AddItem Form1.courseinfo.Fields("coursecode")
List2.AddItem ""
Else
Call Form11.findteacher(Form1.courseinfo.Fields("tid"))
List1.AddItem Form1.courseinfo.Fields("coursecode")

```

```

List2.AddItem Form11.tname11 + " " + Form11.tsurname11
End If
End If
Form1.courseinfo.MoveNext
Wend
End If
Form1.department.MoveNext
Wend
Call Form1.tablokatpat
'end of filling all lecture
Call Form11.findtermid
If Form11.termid11 = "" Then
MsgBox ("firstly open term")
Else
Call Form1.tabloac
Form1.course.MoveFirst
While Not Form1.course.EOF
If Form1.course.Fields("termid") = Form11.termid11 Then
Form1.courseinfo.Index = "primarykey"
Form1.courseinfo.Seek "=", Form1.course.Fields("courseid")
If Form1.courseinfo.NoMatch = 0 Then
If Form1.courseinfo.Fields("deptid") = Text1.Text Then
List4.AddItem Form1.courseinfo.Fields("coursecode")
End If
End If
End If
Form1.course.MoveNext
Wend
Call Form1.tablokatpat
End If
'donemlik derslerin secimi sonu
End Sub
Private Sub Command2_Click()
Dim c2 As Integer
Dim c14 As Integer
Call Form11.findtermid
If Form11.termid11 = "" Then
MsgBox ("firstly open the term")
End If
If List3.ListCount > 0 Then
Call Form1.tabloac
For c2 = 0 To List3.ListCount
Form1.courseinfo.MoveFirst
While Not Form1.courseinfo.EOF
If Form1.courseinfo.Fields("deptid") = Text1.Text Then
If Form1.courseinfo.Fields("coursecode") = List3.List(c2) Then
On Error GoTo jump
Form1.course.AddNew
Form1.course.Fields("courseid") = Form1.courseinfo.Fields("courseid")
Form1.course.Fields("termid") = Form11.termid11

```

```

If IsNull(Form1.courseinfo.Fields("tid")) Then
Form1.course.Fields("tid") = ""
End If
jump:
If Err = 3022 Then
MsgBox ("you can not enter course two times in same term")
Call Form1.tablokapat
List3.Clear
End If
If c14 = 2 Then
MsgBox ("updating is successful...!!!")
End If
End Sub
Private Sub Command3_Click()
If List3.ListIndex = -1 Then
MsgBox ("select course")
Else
List3.RemoveItem (List3.ListIndex)
End If
End Sub
Private Sub Form_Load()
Dim cid As String
Call Form1.tabloac
Form1.department.MoveFirst
While Not Form1.department.EOF
Combo1.AddItem StrConv(Form1.department.Fields("deptname"), vbProperCase)
Form1.department.MoveNext
Wend
Call Form1.tablokapat
Combo1.ListIndex = 0
Call Form11.findtermid
Call Form1.tabloac
Form1.course.MoveFirst
While Not Form1.course.EOF
If Form1.course.Fields("termid") = Form11.termid11 Then
cid = Form1.course.Fields("courseid")
Form1.courseinfo.Index = "primarykey"
Form1.courseinfo.Seek "=", cid
If Form1.courseinfo.NoMatch = 0 Then
List4.AddItem Form1.courseinfo.Fields("coursecode")
End If
End If
Form1.course.MoveNext
Wend
Call Form1.tablokapat
Label1.Caption = Form11.termname11
Label4.Caption = Date
End Sub
Private Sub Form_Unload(Cancel As Integer)
Form6.Show

```



```

Form14.Hide
Call clearlist
End Sub
Private Sub List1_Click()
Dim i As Integer
List1.TopIndex = tind
End Sub
Private Sub Command1_Click()
Dim photo As String
sname = ""
stsurname = ""
stdepartment = ""
If Text1.Text <> "" Then
    Call cleanlist
    Call findstudent(Text1.Text)
    Text2.Text = sname
    Text3.Text = stsurname
    Text4.Text = stdepartment
    photo = "C:\University Information\pictures\" + Text1.Text + ".jpg"
On Error GoTo jump
Picture2.Picture = LoadPicture(photo)
    'finding courses
jump:
If Err = 53 Then
photo = "C:\University Information\pictures\Nophoto.jpg"
Picture2.Picture = LoadPicture(photo)
End If
    Call Form1.tabloac
Form1.grade.MoveFirst
While Not Form1.grade.EOF
    If Form1.grade.Fields("stno") = Text1.Text Then
        Combo1.Enabled = True
        Text1.Locked = True
        If IsNull(Form1.grade.Fields("grade")) Then
            Call findcourse(Form1.grade.Fields("courseid"))
            Call findteacher(tid11)
            List6.AddItem StrConv(tname11, vbProperCase) + " " +
StrConv(tsurname11, vbUpperCase)
            List6.Refrance
        End If
    End If
Form1.grade.MoveNext
Wend
    Call Form1.tablokapat
' end of finding courses
Else
a = MsgBox("enter student number", 64)
Call cleantext
Text1.SetFocus
Call cleanlist
End If

```

```

If Text1.Text <> "" Then
    Call finddeptid(Text1.Text)
'departmentid
    Call Form1.tabloac
    Form1.courseinfo.MoveFirst
    While Not Form1.courseinfo.EOF
controlequ = 0
controlnotequ = 0
    Text5.Text = departmentid
If Form1.courseinfo.Fields("deptid") = departmentid Then
    Form1.grade.MoveFirst
    While Not Form1.grade.EOF
If Form1.grade.Fields("courseid") = Form1.courseinfo.Fields("courseid") Then
    controlequ = controlequ + 1
    Elsecontrolnotequ = controlnotequ + 1
    End If
    Form1.grade.MoveNext
    When
If controlnotequ = Form1.grade.RecordCount Then
    Combo1.AddItem UCase(Form1.courseinfo.Fields("coursecode"))
    End If
Form1.courseinfo.MoveNext
Wend
Call Form1.tablokapat
Else
a = MsgBox("enter student number", 64)
End If
End Sub
Public Function studentcourse(stnol1 As String)
Call finddeptid(stnol1)
'departmentid
Call Form1.tabloac
Form1.courseinfo.MoveFirst
While Not Form1.courseinfo.EOF
If Form1.courseinfo.Fields("deptid") = departmentid Then
    Form1.grade.MoveFirst
    While Not Form1.grade.EOF
        If Form1.grade.Fields("courseid") = Form1.courseinfo.Fields("courseid") Then
GoTo jump11
        For crcontrol11 = 0 To Form1.grade.RecordCount
            If Form1.grade.Fields("courseid") = Form1.courseinfo.Fields("courseid") Then
                controlequ = controlequ + 1
            Else
                controlnotequ = controlnotequ + 1
            End If
            Form1.grade.MoveNext
        Next
        If controlnotequ = Form1.grade.RecordCount Then
            Combo1.AddItem Form1.courseinfo.Fields("coursecode")
        End If
    End If

```

```

    Form1.grade.MoveNext
Wend
jump11:
Form1.courseinfo.MoveNext
Wend
Call Form1.tablokapat
End Function
Public Function findcourse(courseid11 As String)
Form1.courseinfo.Index = "primarykey"
Form1.courseinfo.Seek "=", courseid11
If Form1.courseinfo.NoMatch = 0 Then
If IsNull(Form1.courseinfo.Fields("tid")) Then
tid11 = ""
Else
tid11 = Form1.courseinfo.Fields("tid")
End If
coursecode11 = Form1.courseinfo.Fields("coursecode")
coursename11 = Form1.courseinfo.Fields("coursename")
If IsNull(Form1.courseinfo.Fields("credit")) Then
credit11 = ""
Else
credit11 = Form1.courseinfo.Fields("credit")
End If
coursetype11 = Form1.courseinfo.Fields("coursetype")
If IsNull(Form1.courseinfo.Fields("prereq")) Then
prereq11 = ""
Else
prereq11 = Form1.courseinfo.Fields("prereq")
End If
Else
a = MsgBox("course not found", 64)
End If
End Function
Public Function findcourses(stno As String)
count11 = 1
'Call Form1.grade
Form1.grade.MoveFirst
While Not Form1.grade.EOF
If count11 = 2 Then GoTo finish
If Form1.grade.Fields("stno") = stno Then
courseidcontrol11 = Form1.grade.Fields("courseid")
End If
finish:
Form1.grade.MoveNext
Wend
'Call Form1.grade
End Function
End sub

```