



NEAR EAST UNIVERSITY

FACULTY OF THE ECONOMICS AND
ADMINISTRATIVE SCIENCES

DEPARTMENT OF COMPUTER
INFORMATION SYSTEMS

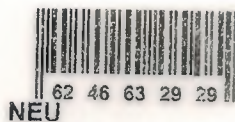
2006/2007 SPRING TERM
CIS 400

STUDENT REGISTRATION SYSTEM

SUBMITTED TO : DEPARTMENT OF CIS

SUBMITTED BY : MEHMET YUSUF AKKILIÇ
20020661

Lefkoşa
(2007)



CONTENTS



ACKNOWLEDGEMENT.....	2
ABSTRACT.....	3
INTRODUCTION.....	4
METHODS.....	6
DESIGN OF THE SYSTEM.....	6
• Explanation.....	6
• System Block Diagram.....	20
• Data Flow Diagram.....	21
• Database Design.....	30
USER MANUAL.....	41
• User Flow Charts.....	41
REFERENCES.....	59
APPENDIX1.....	60
• Screen Outputs.....	60
APPENDIX2.....	91
• Source Code.....	91

ACKNOWLEDGEMENT

anks from:

y instructors for their supports in my project to complete it.

anks to;

. Asist.Prof.Dr Yalçın AKÇALI, Miss.Dr. Nadire ÇAVUŞ, Mr.Assoc.Prof.Dr. İlham
USEYINOV and Mr. Tansel DEVİN, for their supports in my project. Also I thanks from
y another instructors for their contributions .

so thanks to ; my family for providing me with financial support while preparing the
oject and for psychological support in all parts of life by providing me comfort and
ecessary conditions to develop myself and educate in high level.

ABSTRACT

Nowadays we know that, the Information Technology is very important in the human life. Almost we use the IT in our life every time. Why do we use it ? Because it provides us to do our business very easily, quickly and securely. Therefore this project called STUDENT REGISTRATION SYSTEM. System will provide for workers of the universities, for teachers, as well for students to do their business.

In the world is possible to find and to reach to each different information from the internet or any places. The technology is improving very fast. By this technological improvement of course the computer and internet world has more a big improvement in the technology. If we want to follow the technology or if there is a competition between the business or in any subjects we must produce and use like this programs.

INTRODUCTION

The name of the project and document is called Student Registration System. This project will be used in the universities or who is using for student processes. These processes can be student information, the university information, registration information and teacher. These are the availability of these projects.

There will be an easy user interface of this project. The aim of the simple interface is to provide easily to use this project for secretary or who is using this program. Especially this document will be very useful when this program is worked. For some unknown places in the project, the user can take benefit from this document.

When we design this project, we need some programs and some documents as well we need a good research about Registration System. Before starting of the project a useful researching, we must do. In fact this researching is valid for all the like this project.

We had explained a little above about the aim of the project. This project or program will provide to learn and make process for student registration and grades, also for situation of teachers.

User Requirement definition

1. The system must cover all the registration activities.
2. The system must be flexible and easy to use either to novice or to professional computer users.
3. The system users should have a permeation to login inside the system space.
4. To increase the speed of the operation by the users, and must have good server machine for local and for internet.
5. Generally by the users to prevent loss of data.
6. To restrict some privileges from the system for attachment of the hacker.
7. Should have passwords of system, according to use it.

– System Requirements specification

1. The authorized individuals (computer center authorized staff) should be provided with facilities to access information about the structure of the system.
2. Unauthorized individuals are unable to gain access to the structure of the website to make their operations.
3. Making constraints on users; every user has his limited authorizations.
4. The system should be connected to the university server.

✓

I am going to build this system to solve whole problems in the old one. These problems look as: the time as a main problem and end-users needs.

I want here to classify some problems face us in the current system:

1. A lot of paper using in registration operation.
2. The student kills a lot of time in the registration operation; “about one or two work days to make a complete operation”.
3. A lot of money expenses on papers, printers, electricity, and salaries for registration staff.
4. The old operation is not flexible. It is very complex and cumbersome.
5. The back-up operation for the registration and students data is difficult and not safety. Because it depends on the paper and hard copies.
6. Registration employees staff mistakes in transferring data inside the system.

I want to talk that there are more organizations in this time built and developed computerized solutions for their problems and tasks. Of course, they are certainly believed that this new technology is very feasible and efficient to them. So, you can see the internet technology enters the commercial and scientific sectors as a same level. This system we are going to build shows a very good efficiency in more universities and institutes over the world.

METHODS

As stated before, Object Oriented approach is a good approach for designing software products, since it uses objects around and trying to implement those objects in software. Here after analyzing the software after reading the requirements below:

1. The registration system could not work over the web, maybe in future we will set for web.
2. The system must keep track of all the students' information
3. The system would be able to show the students grades.
4. The system would have error checking mechanism for wrong entries.
5. The instructors would be able to put grades to students.
6. Student in a class/lab should not be more than 30 students.
7. Security should be taking into the consideration.

I assumed that students paid their tuition fees before registering themselves.

DESIGN OF THE SYSTEM

EXPLANATION :

The minimum requirements of the system are as follows;

- Intel Pentium 3 processor or Celeron 2.00
- 550 Mhz
- 128 MB of RAM
- 30 GB HDD
- 14'' Monitor
- Microsoft Windows 98/2000

- CD-ROM 52xmax

The recommended requirements of the system are as follows;

- Intel Pentium 4 processor
- 800 Mhz
- 512 MB of RAM
- 80 GB HDD
- 17'' Monitor
- Microsoft Windows XP Professional
- CD-ROM 52xmax

A Brief Description of How the Current System Works

We can classify the whole registration operation to three main points as shown below:

1. Selects courses for each semester and makes the time tables for the lectures (done by the chairman of each department)
2. The students registration operation (done by the registration department employees staff)
3. Paying fees for the university accountants (done by students themselves and accounting department staff)

I want now to detail the steps of each point above in small sentences:

- How the selecting semester courses operation done?

A Description of How the Current System Works

- a. The chairman of the department, plus some helps from the dean and registration department, selects how many courses will be offer in this semester, and what are there.
- b. Assigning the number expectant students in each course in this semester.
- c. Assigning the available classroom for each course and each section depends on time and number of seats.
- d. Assigning the teacher for each course depends on his/her knowledge.
- e. Designing the time table for the weekly lectures.

- How does the registration system work?

- a. For each department in the university there is a specific registrar. The goal of him is entering the registration data for the students' registration operation.
- b. Each student should visit his/her registrar to take his Registration Form to start his registration operation.
- c. Now the student should visit his/her advisor to take advises of what the best courses he/she can enroll in, and then sign the registration form.
- d. After that, the student should return the registration form for his/her registrar to reenter the data into student's official file.

- How the student can pay his/her fees?

- a. After finishing registration operation, all students' financial data transferred to the accounting department in the university.
- b. The accountant calculates the fees on each student.
- c. Each student goes to the accountant to pay his/her fees depends on his/her educational level (under or post graduates)
- d. The accountant adds the health insurance and social activates fees over all students as a same.

In this way, we can say that the registration operation is completed successfully now.

A Description of How the New System Works

We can classify the whole system registration operation to some main points and we will explain later as shown below;

1. The system should have information's for students to make registration processes for the students. (done by secretary, administrator)
2. The system should have a course menu to make add, update, delete and selects courses for each semester and makes the time tables for lecturers (done by chairman of each department)
3. The system should includes lecturers menu as well, to take information about them and makes some operations for them.(done by the administrator)
4. For each faculty we have department(s), the system must include department's information. (done by the administrator)
5. For student's registration, the system allows the user to enroll the students in the courses. (done by the advisor)
6. The system should give report for any registration operation (will see by admin, lecturer etc...)
7. Login panel will be found in the system to register any new user to use this system. (done by the administrator)

Let's detail each point above as steps shown below;

1. How the system should have information's for students to make registration processes for the students?

- Entering the registration data for the students' registration operation.
- The new system making the student processes. After that advisor and the secretary will make some process about the students. So follow the other steps.

- The staff of the university registrars register student or add a new one to the system. System gives the number of students and then accepts it as student. So all personal information of student will be stored in the system database.
- The advisor of the student will give the courses by using advisor access level permission. Advisor and student choose courses, and then registration will be completed.
- System provides to admin, advisor and other staff of the university to update available student information, to change their information.
- As well system allows the authorized staff to delete student registration.
- In other steps will be explained student registration for to take course and for their payments, debts.

2. How the system should have a course menu to make add, update, delete and selects courses for each semester and makes the time tables for lecturers?

- The authorized user from the university will add courses for each departments of the faculty. The system must have some description about the courses. When student takes the course for their choice it will be useful.
- The current stored courses will be updated by the users. May some changes can occur by the departments' courses.
- Also some course may be closed or cancelled. The system allows the users to delete it from system.
- The chairman of the department, plus some helps from the dean and registration department, selects how many courses will be offer in this semester, and what they are.

3. How the system includes Lecturer as well, to take information about lecturer and makes some operations for lecturer:

- New lecturer can be adding to the system by using the lecturer process. It will be added by admin of the system.
- All information of the lecturer like who is in which department and what is his/her title should be stored in the system by using addition process.

- Current lecturer may update by using the system.
- May a lecturer can be deleted from the system when he/she left from the working and may retire.

4. For each faculty we have department(s), the system must include department information:

- For the university faculty of course we have departments name and their description, so we must store these departments' name and their descriptions about them by using addition process in the system.
- Maybe some changes occur from departments' names and their information so will be updated by the authorized users of the university.
- Also we can delete department from the system, when it is closed down from the university.

5. For course registration of student, system allow to the user to enroll the students and give the courses to the students:

- Advisor will give some courses to the student by using the course registration process from the system. Advisor will choose one or more courses and then can give the student account.
- Here also advisor can see the information of the student and their GPA, CGPA, and which course he/she has passed or failed, advisor can give course to student according to this information.
- Maybe some changes can be occur from the student courses, system will allow to the user to drop and add some course for student.
- As well as, system will have the grading for students. After examinations or at the end of the term, staff of the university will store the grade of the students. (Lecturer sends to the staff the list of the students' courses with grading and then realized this process)

6. How the system should give report for any registration operation?

- System show the users some reports, but it will change according to the authorized user

- Generally some of these reports will be teacher, department, and course reports; just it will show to the users' information about these things.
- Also reports for departments to teachers and courses; courses to teacher and departments.
- Most importantly the reports will show the transcript of each student. In transcript is shown the term of the year, which course in which term of year, as well GPA and CGPA of the student will be shown.

7. Login panel will be in system, to register any user to use the system to do some operations:

- Settings of the system will be specified here by specify the user according to their privileges and limitations.
- New user can be specified here as advisor, secretary, admin and normal staff.
- Also current users' information, their privileges and limitations can be updated from here.

The Feasibility Study

Firstly, I want to remember the goal of our project. It is to change the traditional view of the registration operation to a new one to save time and money if we can.

In software engineering process, we classify this study to three major parts:

- **Economic Feasibility**: that's mean assigning the costs to build the system and then compare it with the current and future benefits. The result should be positive. Means the benefits is more the costs.

Benefits

- **Tangible Benefits:**
Increasing the speed of registration operation and making the process more flexible.
- **Intangible benefits:**

Updating the traditional way of registration to be modern.

Costs

- Tangible costs:

We shall calculate the building project expenses and new staff in computer center expenses, plus that we need; servers and web space to active our system.

- Intangible Costs:

User should have knowledge in using computer and internet.

- **Technical Feasibility**: Here we must know the organization needs, to decide if the system technically efficient or not.

In tangible costs we were denote that we should have some new equipments. These equipments will make our system more efficient to make it in a good technical status.

- **Operational Feasibility**: to define the ability of operation the system. Our project is designed to make the registration process for the students computerized so as to develop the registration process to become speedy and easy.

Determining Requirements

We have two general methods for determining requirements: traditional and modern method. In this project, I will choose the modern method to determine the requirements. Joint Application Design (JAD) one of the modern method manners. We will use this manner for these tow reasons:

- Because we want to collect system requirements simultaneously from key users (registration staff and students).
- At the end of using this method we should have detailed features of the proposed system.

- **Functional Requirements**

- The user of the system *shall* be able to gain access to the desired information he/she wants.
- The system *shall* inform the users what on-line registration system is.
- The system *shall* provide some security issues, such as making it difficult to make vandalism in the website.
- The system *shall* provide appropriate tools to deal with it for expected users, e.g. new students.

- **Non-Functional Requirements**

(2.1) Product Requirement:

- a. Performance requirements: the system should have high performance (relatively high speed).
- b. Reliability requirements:
 - i. Hardware reliability: As we know, hardware exhibits relatively high failure rate early in its life, then defects are corrected and failure rate drops to a steady level, then, later on the failure rates rise again. But the key point here is that these defects and failures of the hardware can be easily fixed.
 - ii. Software reliability: the situation here is different from hardware reliability because software doesn't wear out, it does deteriorate.
 - iii. Operator reliability: the operator of the system will rarely produce errors and if so these errors should be handled quickly.

(2.2) Organizational requirements:

In these requirements according to the policies and procedures between customers (receipts of the system; that are students and teachers) and developers, we will deliver the system and its documentation at the specified time, and we will explain the design method that was used.

(2.3) External requirements:

We will ensure that the system will operate within the law and ethical requirements, to make the system acceptable to its users and to the public, in general.

Privileges, Responsibilities And Limitations

Our system has four general of the users, which are;

1. Administrator (computer center staff)
2. Advisor(Teacher, Administrator)
3. Students(User)

When we want to see the privileges, responsibilities and limitations of these users as detail we can define like that;

1. Administrator (computer center staff)

The following is a list of all privileges, responsibilities, administration events and tools for administration team;

- Updating the site as needed.
- Administrating the system database.
- Administrating the registration times.
- Add users, update and remove them from the system.
- Giving limited privileges to the authorized users
- Receiving and studying the feedback information.
- To change the access level of the users.
- Backup the system, when is needed.
- There are no any limitations for the administrator.
- To evaluate the all system reports.
- Control the system in specific time.

- Add department, if there is a new.

2. Advisor(Teacher, Administrator):

For completion the registration of the students, advisor or chairman must give them the courses. So some authority is giving to the advisors. The Lecturer must give the necessary information to the system. The secretary is responsible to enter data of the students.

The list of the authorized privileges, responsibilities and limitations for the registrars in the new on-line system is written below ;

- Controlling the registration operations
- Inserting the courses and sections data for each semester to the system
- Entering the students marks to calculate the GPA
- Making the registration reports for the chairman of each department
- Controlling the penalties on the students if exist
- To transfer the marks of the student to the Registrar
- To prepare some documents for system or for departments
- They can not interfere inside of the system
- Registrar staff can not have all privileges of the administrator
- Can not give courses to the students
- Can not change their access level in the system, they must want from the administrator
- They can not open a new account for login the system themselves, administrator will open their account, specify the access level.
- Advisor must give courses to the students
- Can drop some course for student
- Add or change a course after course registration
- Responsible to interest the student's problems and to reflect on system
- Can not change the access level in system

3.Students(User):

In the new online registration system, the student can now make the registration operation by himself in any time and from any where he/she wants. The privileges for students in new on-line system are written below;

- Sign up to the system by using their registration information like their registrar code.
- Submitting all registration operation data to the system
- To look their mark, debts and payments from the system
- Can not update or change the course or their mark.
- Can not change the access level of the system

Conceptual Database Design Description

Our system contains a database. This database classified all entities in the system. Now, we need to discuss this database and the tables' contains. All the information about registration will be stored in this database. We were designing and implementing the tables and their relationships by using the Microsoft Access programming.

The database has a number of tables, which are:

1. Department
2. Student
3. Teacher
4. Course
5. Transcript
6. Login

We are going now to discuss all these tables in detail. Some tables are not major tables, we defined they here to validate the many-to-many relationships between some tables. We will discuss this idea later on this report.

Department:

Department table has a strong relation with the previous table, faculty table. This table contains a complete description about all departments inside the university. Each

department has an identification code; this code will be as a unique code for each department. It will contains three characters denotes the name of the department. It will help us to name codes of courses offered from the department.

This table has a strong relation with the faculty table to reference each department with its faculty belongs to.

Student :

These tables are a very important table inside our database. They will contain the most important information for all students registered to the university. The student identification number; it is a unique number for all students registered to the university. They give to each student when he registered for the first time. We will use this number to identify all students inside our system.

All the data attributes above stored in the database to help us when the system makes the student registration report after he/she finishing his/her registration operation.

Teacher:

These tables contain some usable information about the Lecturers, Advisors, and Chairmans inside each department. So, we should connect all lecturers with their departments by the department code.

We used the lecturer identification number to define the instructor number here. We can get the lecturer identification number form the employee's affairs department inside the university.

Course:

Any course must offer from a department and not more that one department. Each course has a unique code over all university's courses. This code contains six characters. The first three characters are the department code which it belongs to, and the second three are a unique number denotes the courses offered from a department and its level.

Each course record inside the database will contain the course name, credit hours for the course, number of groups for the course depends on how many students registered to the course, a checking variable if the course has a lab lectures or not.

This table helps students in the registration operation to see all information about the courses offered which they can enroll on it. The system will show just the courses that the student can take them from his/her department.

Transcript:

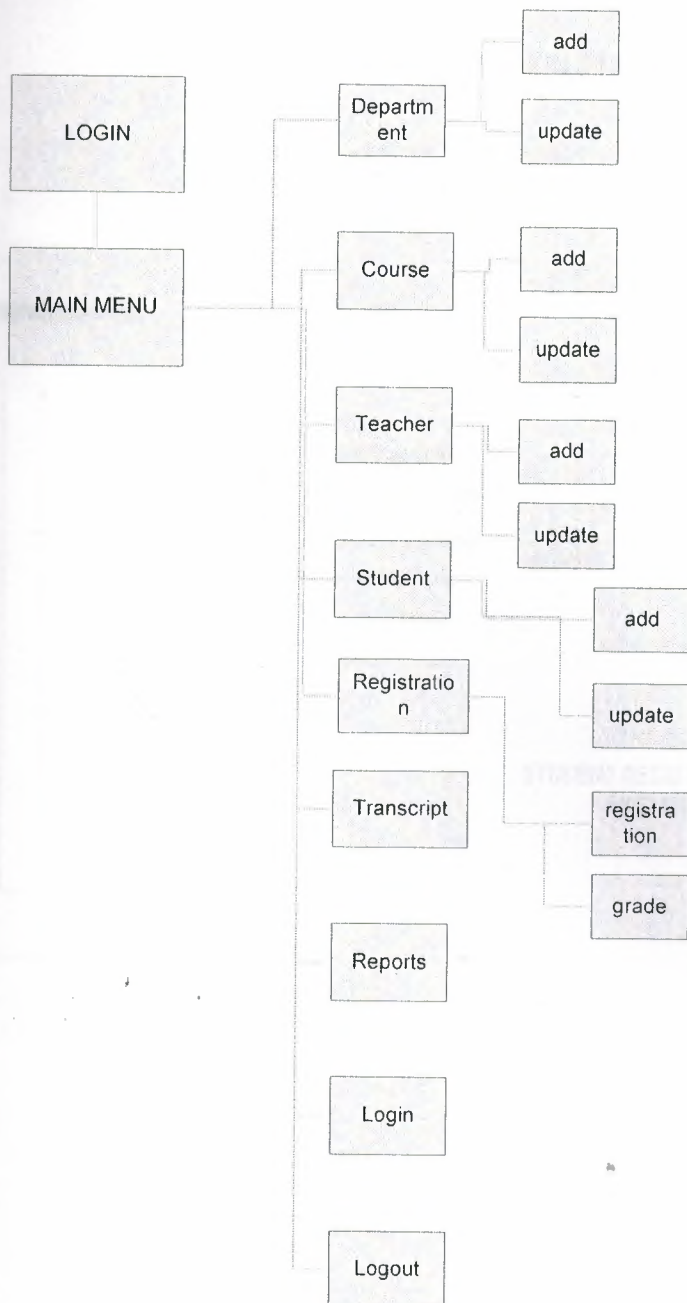
At end of each semester, each instructor or secretary should upload all grades for his/her students that he/she instruct them. The table grade will be containing all this data. The data will be stored here are: the student number and the course code that he/she enrolled in also the grade for his/her in a specific course.

To define the transcript uniquely here in this table, we should insert the student number and the course code together. That's mean, each course has a group of students and each student has a number of grades. Number of grades for each student is equal to number of courses which he/she enrolled in.

Login:

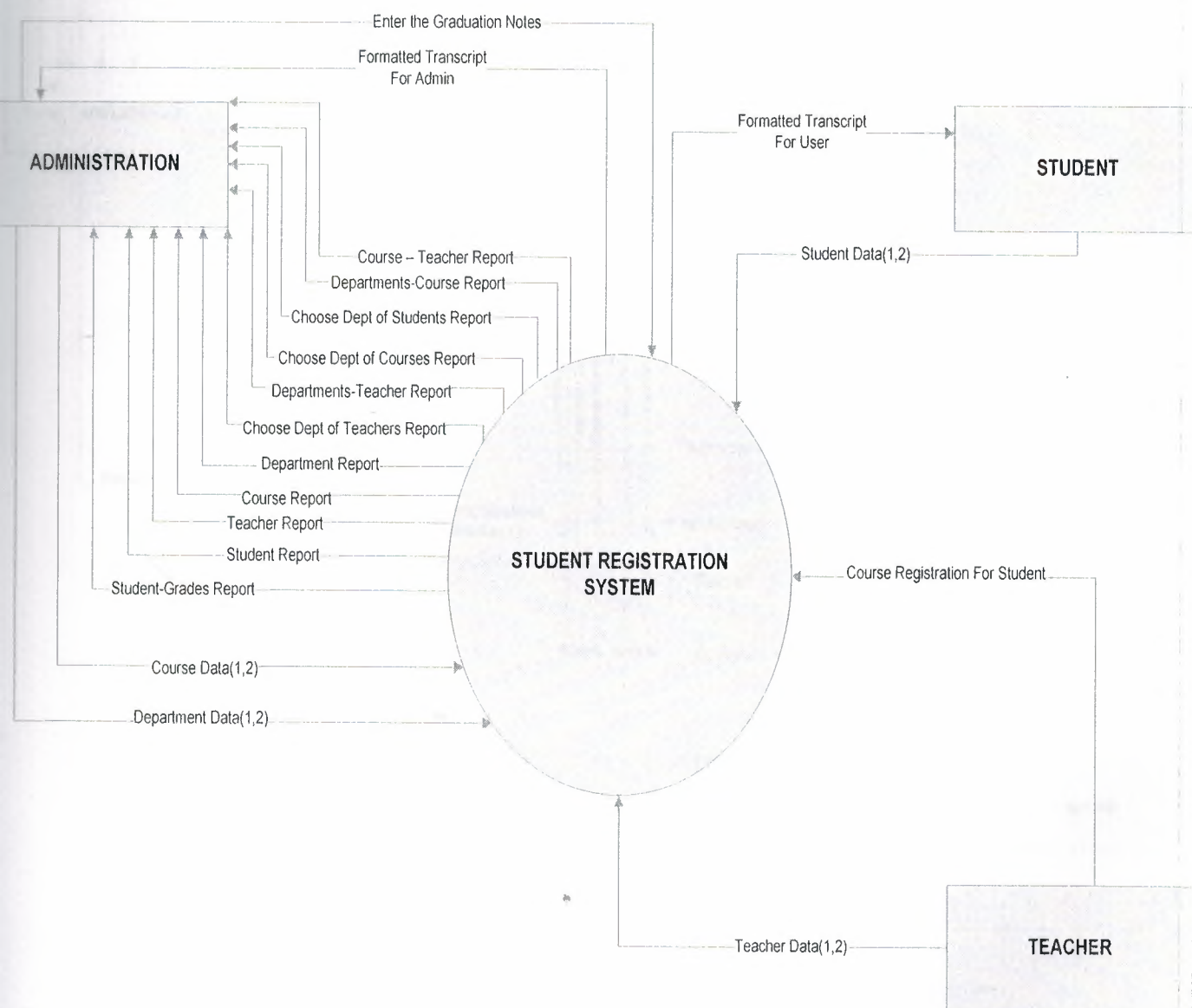
The goal of this table is to register new users and control all the log-in and log-out operations for all users. The system checks all log-in operations in the first page of the system over internet. This table contains a list of usernames, passwords, and the categories for the system's users.

SYSTEM BLOCK DIAGRAM

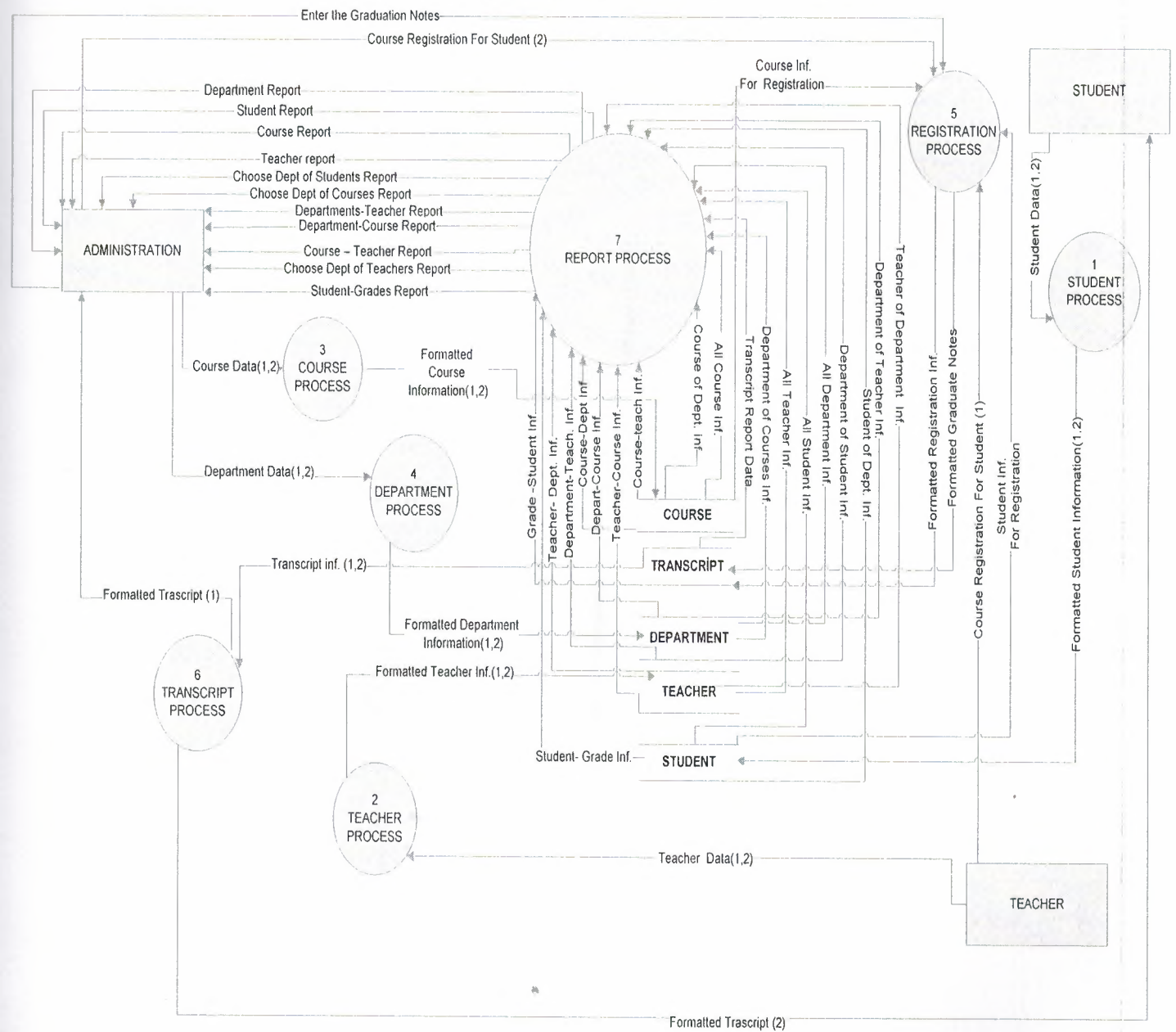


DATA FLOW DIAGRAM:

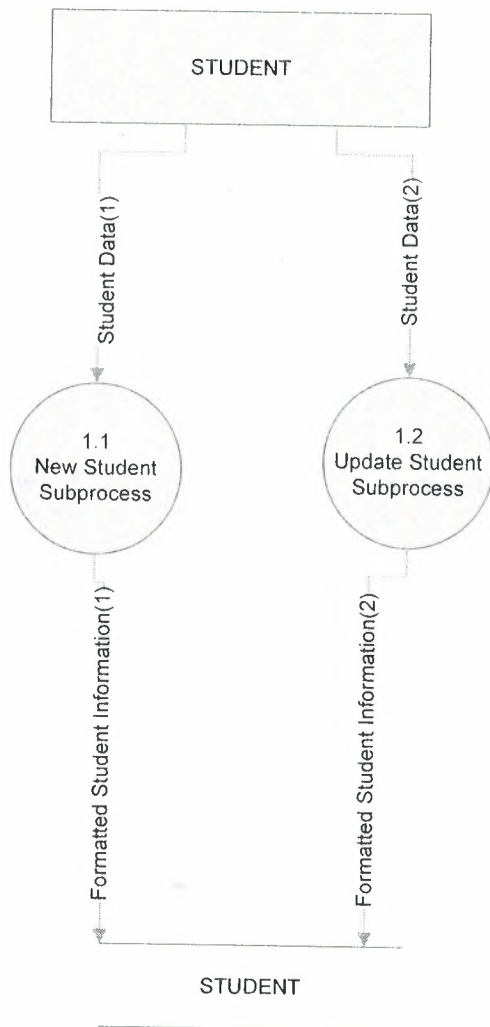
STUDENT REGISTRATION SYSTEM FOR CONTEX DIAGRAM



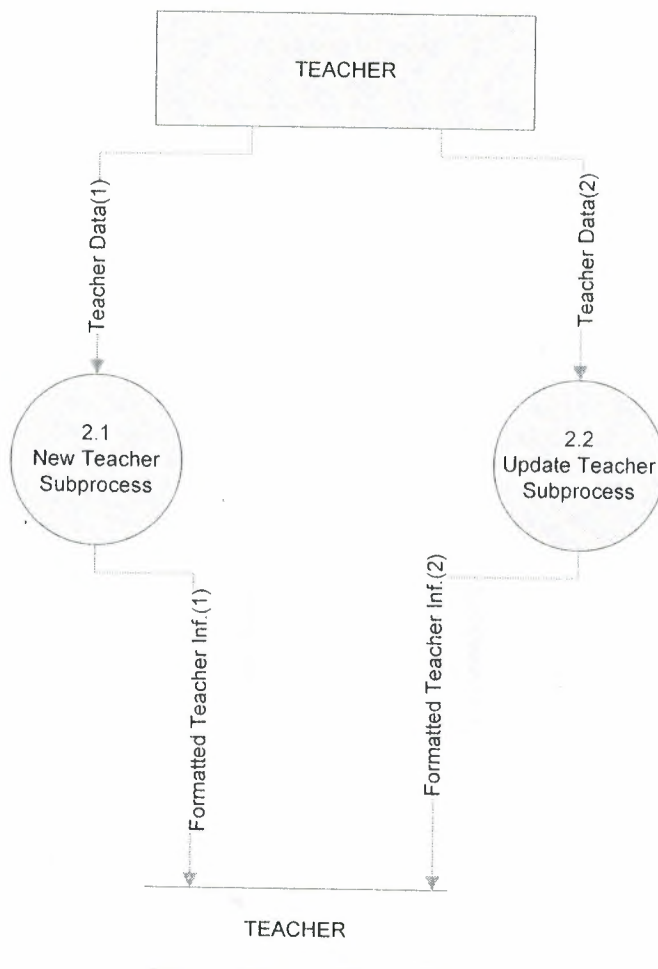
STUDENT REGISTRATION SYSTEM FOR TOP LEVEL DFD



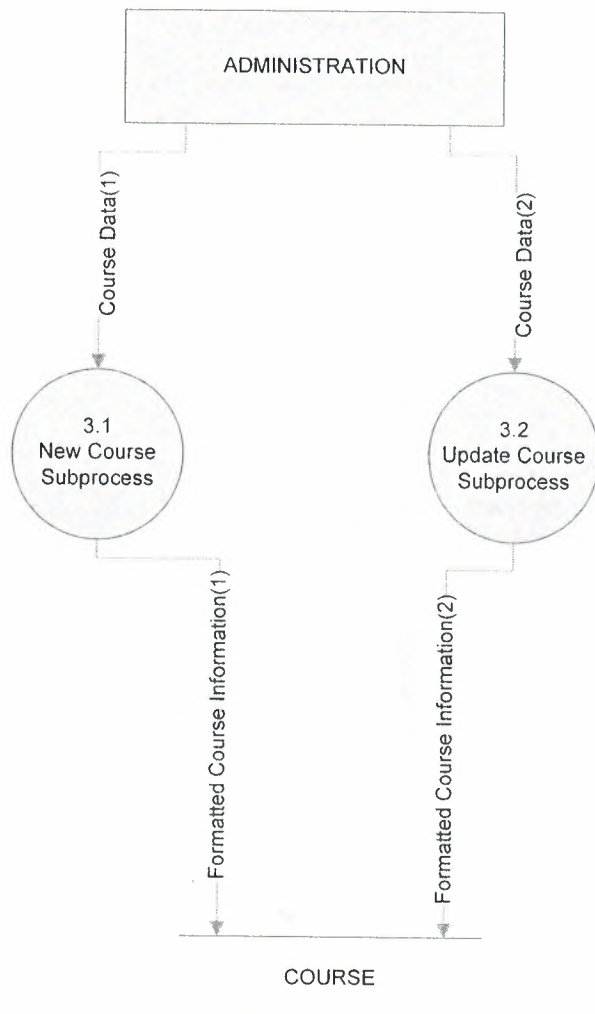
STUDENT PROCESS DETAIL DFD FOR STUDENT REGISTRATION SYSTEM



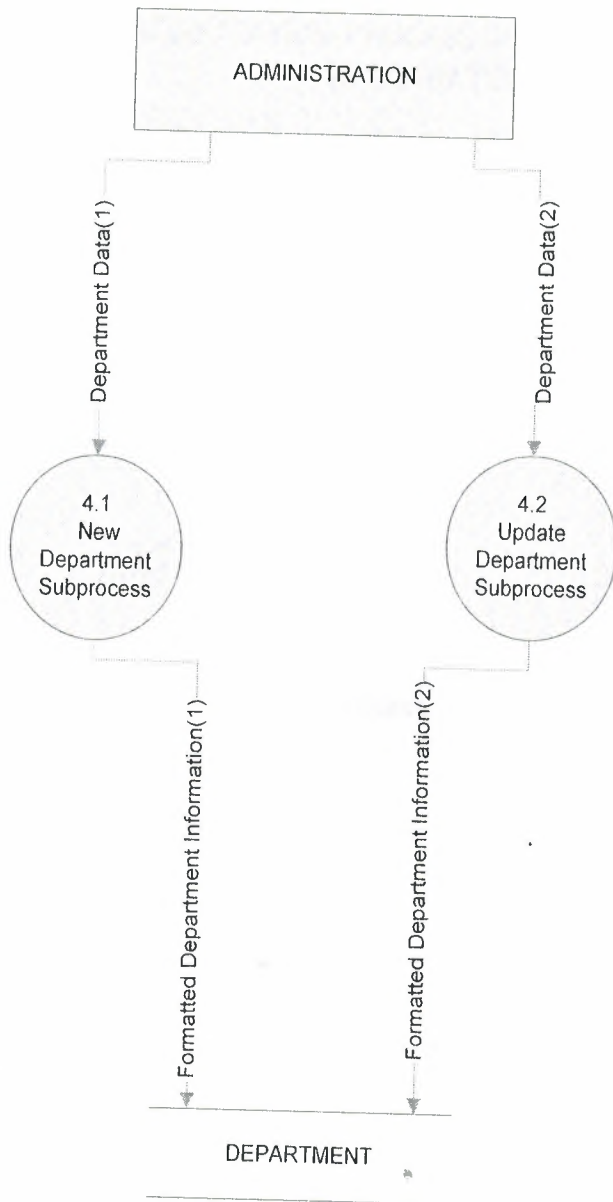
TEACHER PROCESS DETAIL DFD FOR STUDENT REGISTRATION SYSTEM



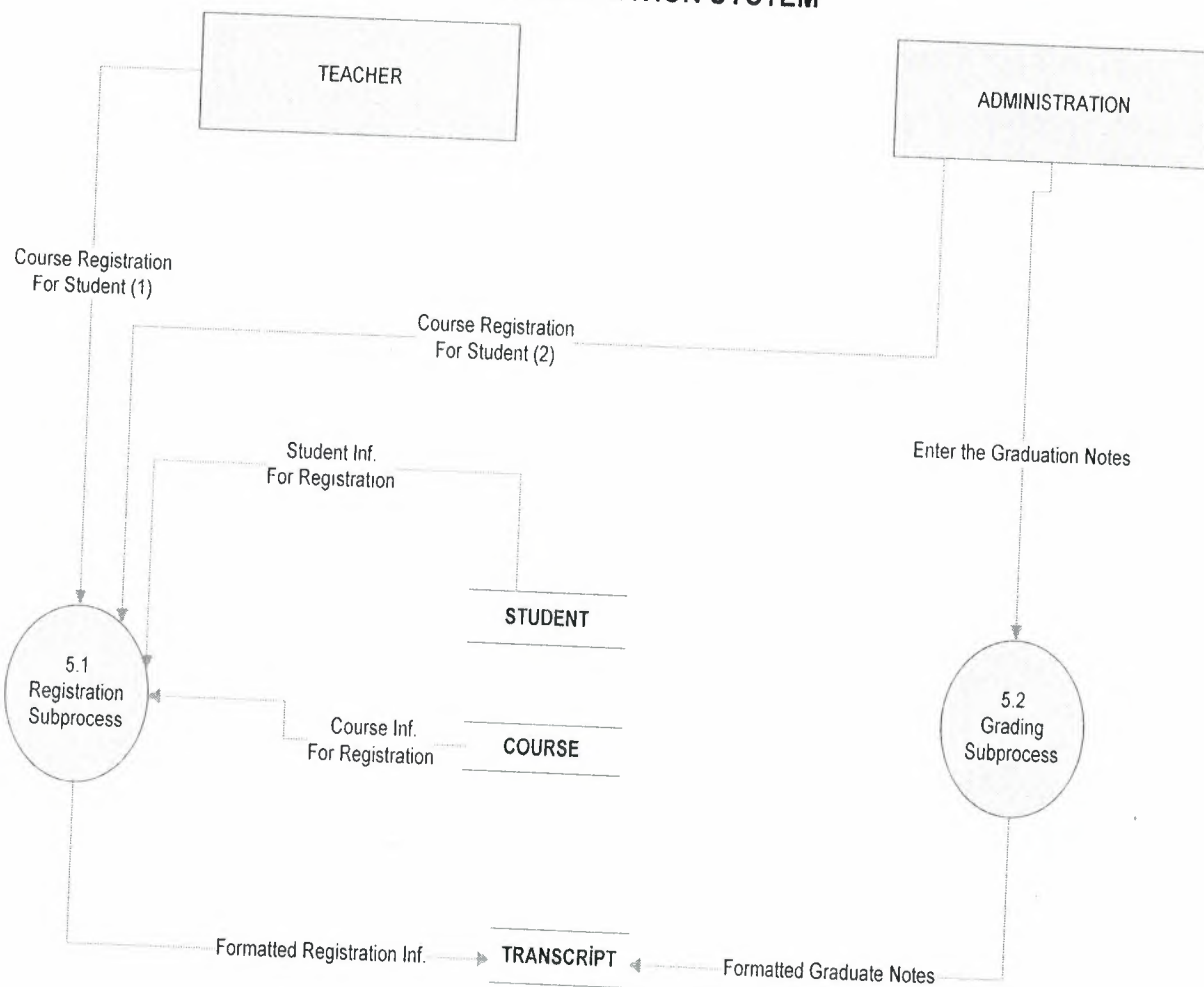
COURSE PROCESS DETAIL DFD FOR STUDENT REGISTRATION SYSTEM



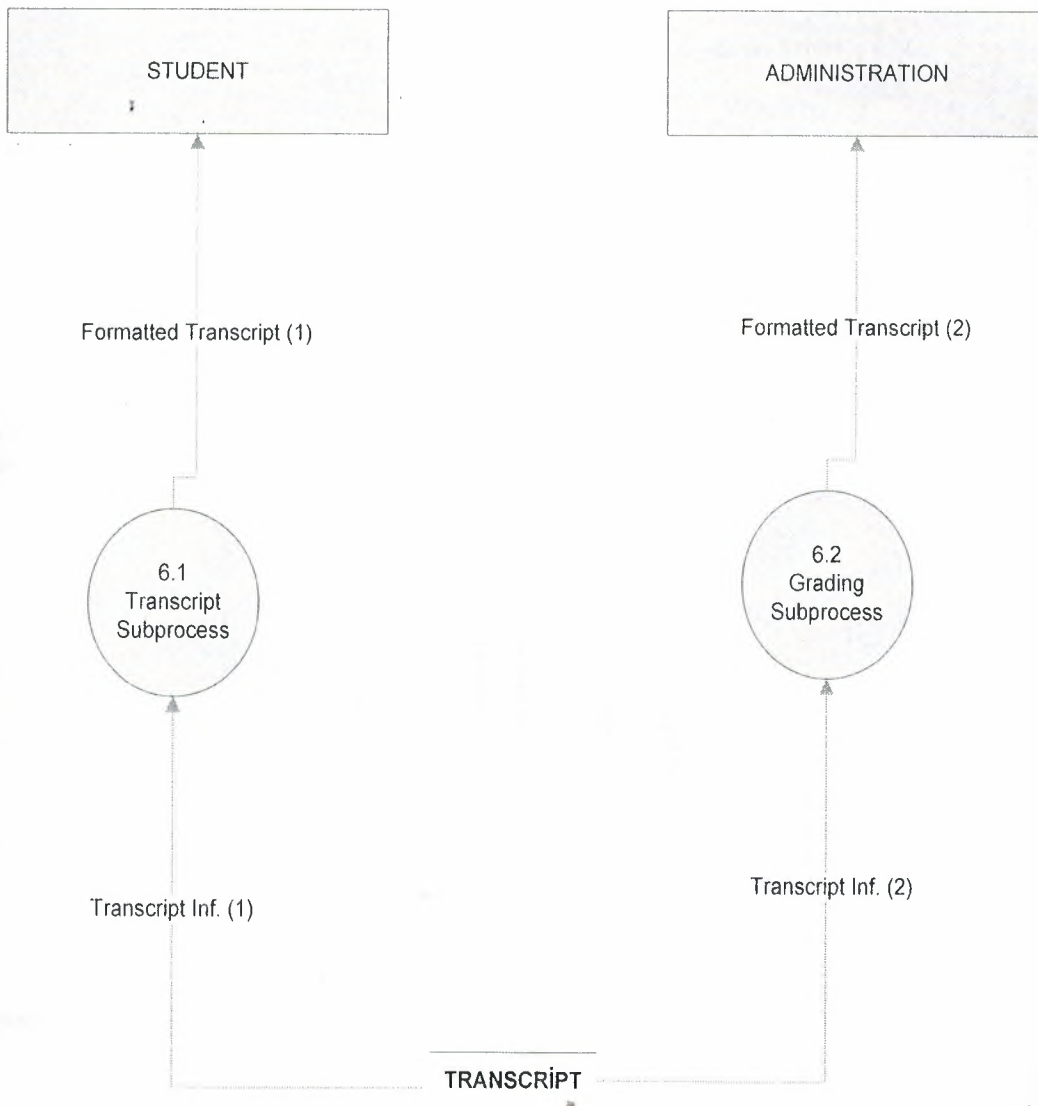
DEPARTMENT PROCESS DETAIL DFD FOR STUDENT REGISTRATION SYSTEM



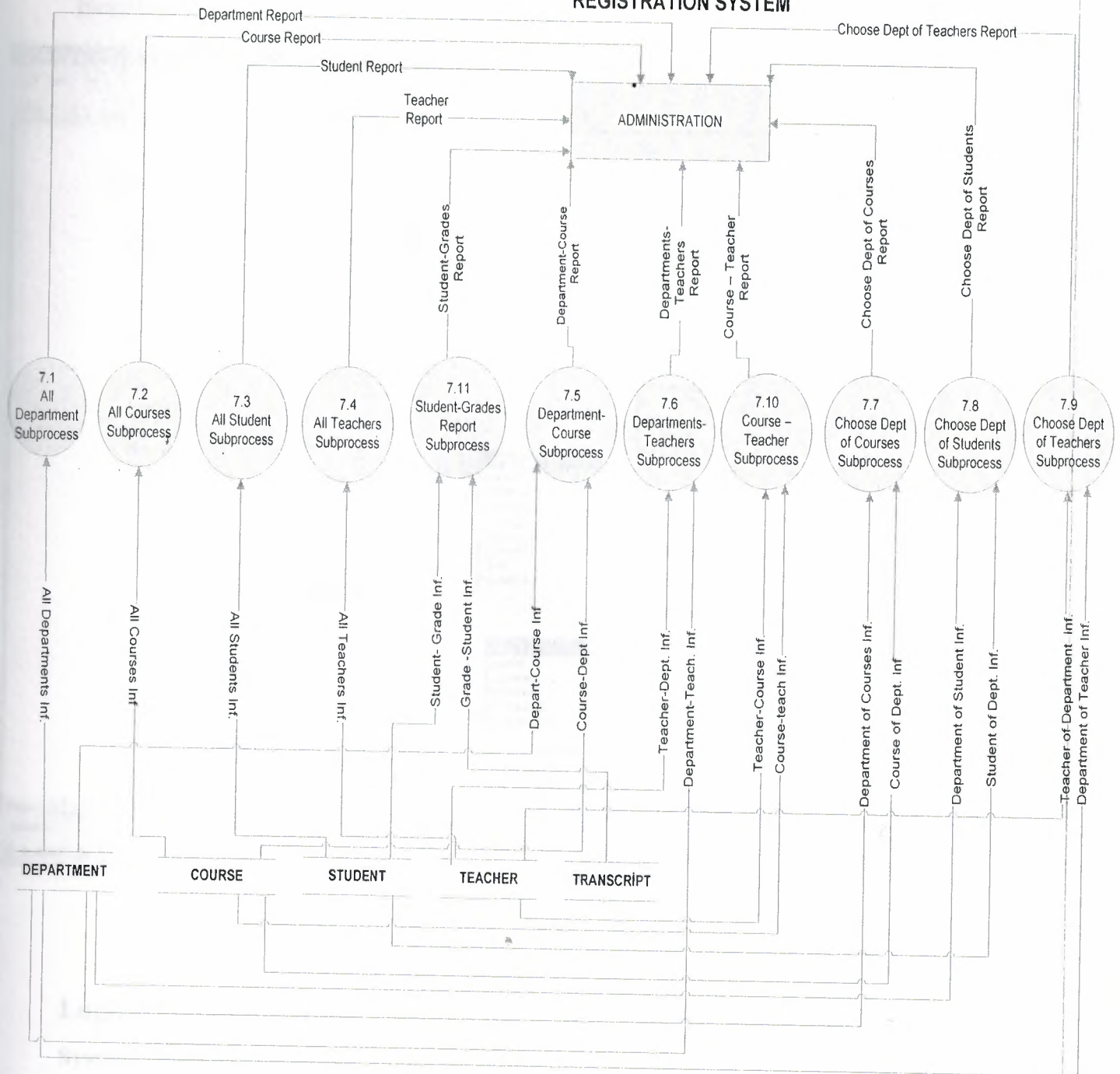
REGISTRATION PROCESS DETAIL DFD FOR STUDENT REGISTRATION SYSTEM



TRANSCRIPT PROCESS DETAIL DFD FOR STUDENT REGISTRATION SYSTEM

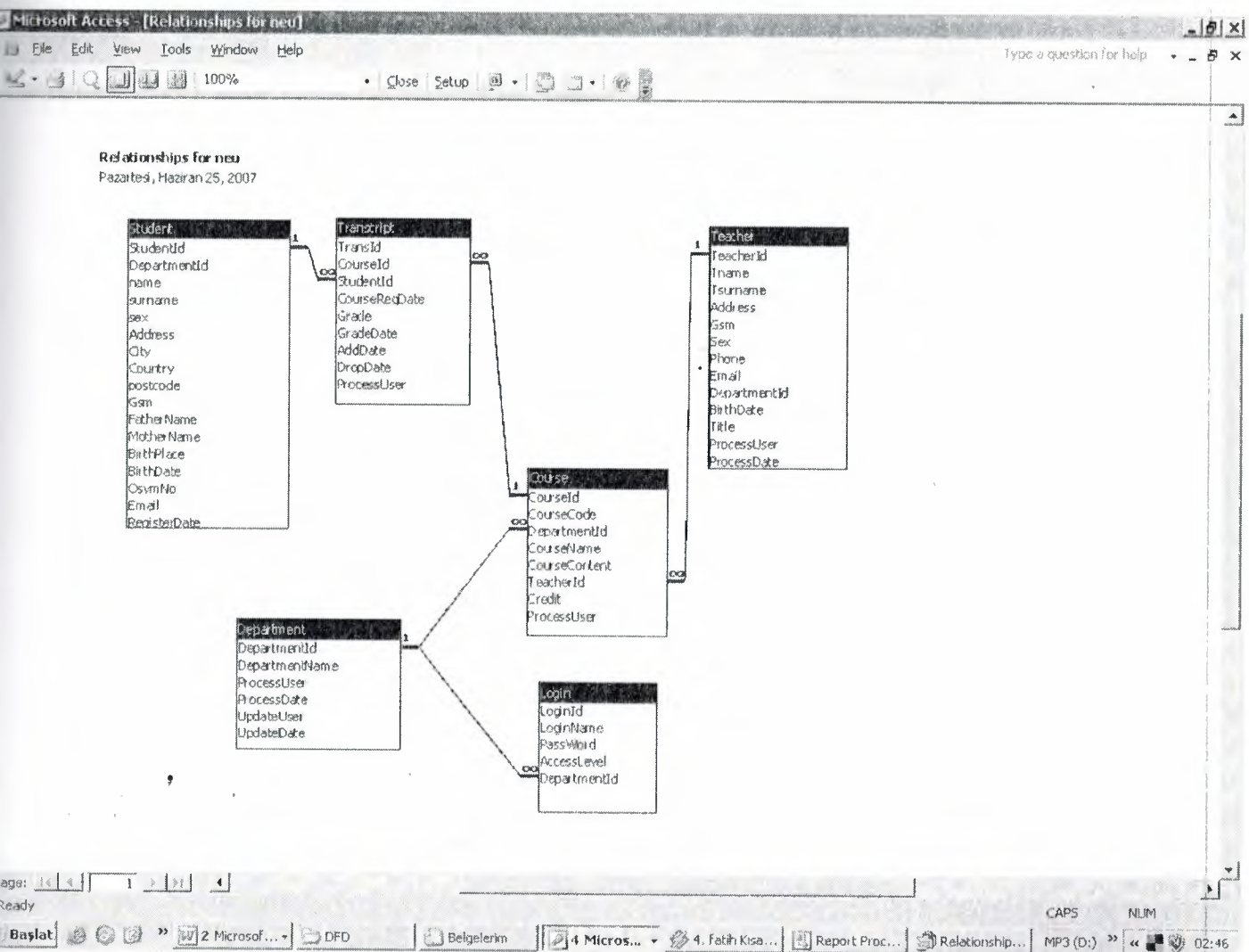


REPORTS PROCESS DETAIL DFD FOR STUDENT REGISTRATION SYSTEM



DATABASE DESIGN:

Relationships:

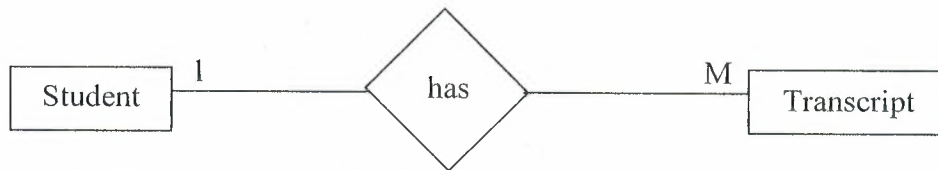


Logical Database Design Description

System Relationships Types

As we mentioned before our system include set of entities and we have between these entities many relationships. These relationships must be clear and simple. In this part we are going to mention and explain all these relationships and all business rules are related to these relationships.

1. Relationship between Student and Transcript:



Cardinality: 1: M (One-to-many)

Figure 2

Business Rule: - One student has one more transcripts.

- Many transcripts are for one student.

2. Relationship between Department and Course :

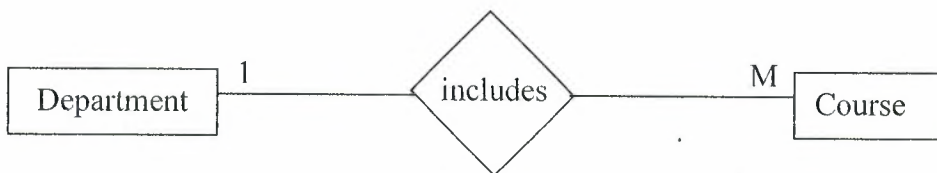


Figure 3

Cardinality: 1: M (One-to-Many)

Business Rule:- One department includes many Courses.

- One course included in one department.

3. Relationship between Teacher and Course :



Figure 4

Cardinality: 1: M (One-to-Many)

Business Rule:- One teacher teaches many courses.

- Many courses can be taught by one teacher.

4. Relationship between Course and Transcript :

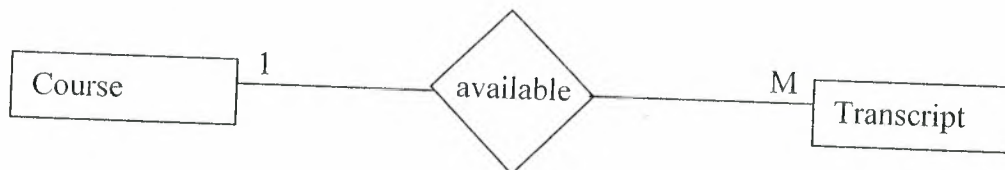


Figure 5

Cardinality: 1: M (One-to-many)

Business Rule:- One course can be available in more transcripts.

- One transcript has single course(not same more).

5. Relationship between Department and Login:



Figure 6

Cardinality: 1: M (One-to-Many)

Business Rule:- In a department can be logged many users

- Many users can be logged in one department.

Document Attributes for the System

Entity	Attributes names	Data type and length	Constraint	Real Name
TRANSCRIPT	TransId	4 auto number	PK FK2 FK1	Transcript Id
	Courseld	6 number		Course id
	StudentId	10 Text		Student id
	CourseRegDate	Date/time		Course Reg. Date
	Grade	2 text		Grade
	GradeDate	Date/time		Grade Date
	AddDate	Date/time		Adding Date
	ProcessUser	20 text		Process user
	ProcessDate	Date/time		Process date
	UpdateUser	20 text		Update by user
STUDENT	StudentId	10 text	PK	Student id
	DepartmentId	10 text		Department id
	Name	30 text		Student first name
	surname	20 text		Student lastname
	sex	4 text		Sex
	address	50 text		Address
	city	20 text		City
	country	20 text		Country
	postcode	20 text		Postcode
	Gsm	20 text		Mobile phone
	Fathername	20 text		Father name
	Mothername	20 text		Mother name
	Birthdate	Date/time		Birth date
	Birthplace	20 text		Birth place
	Osymno	20 text		Osym no
	Email	20 text		Email address
	Registerdate	Date/Time		Registered date
	Phone	20 text		Phone
	Isgraduated	Yes/No		Is graduated?
	Graduationdate	Date/time		Graduation date
	Graduationcom	20 text		Graduation comment
	Isdismissed	Yes/no		Is dismissed?

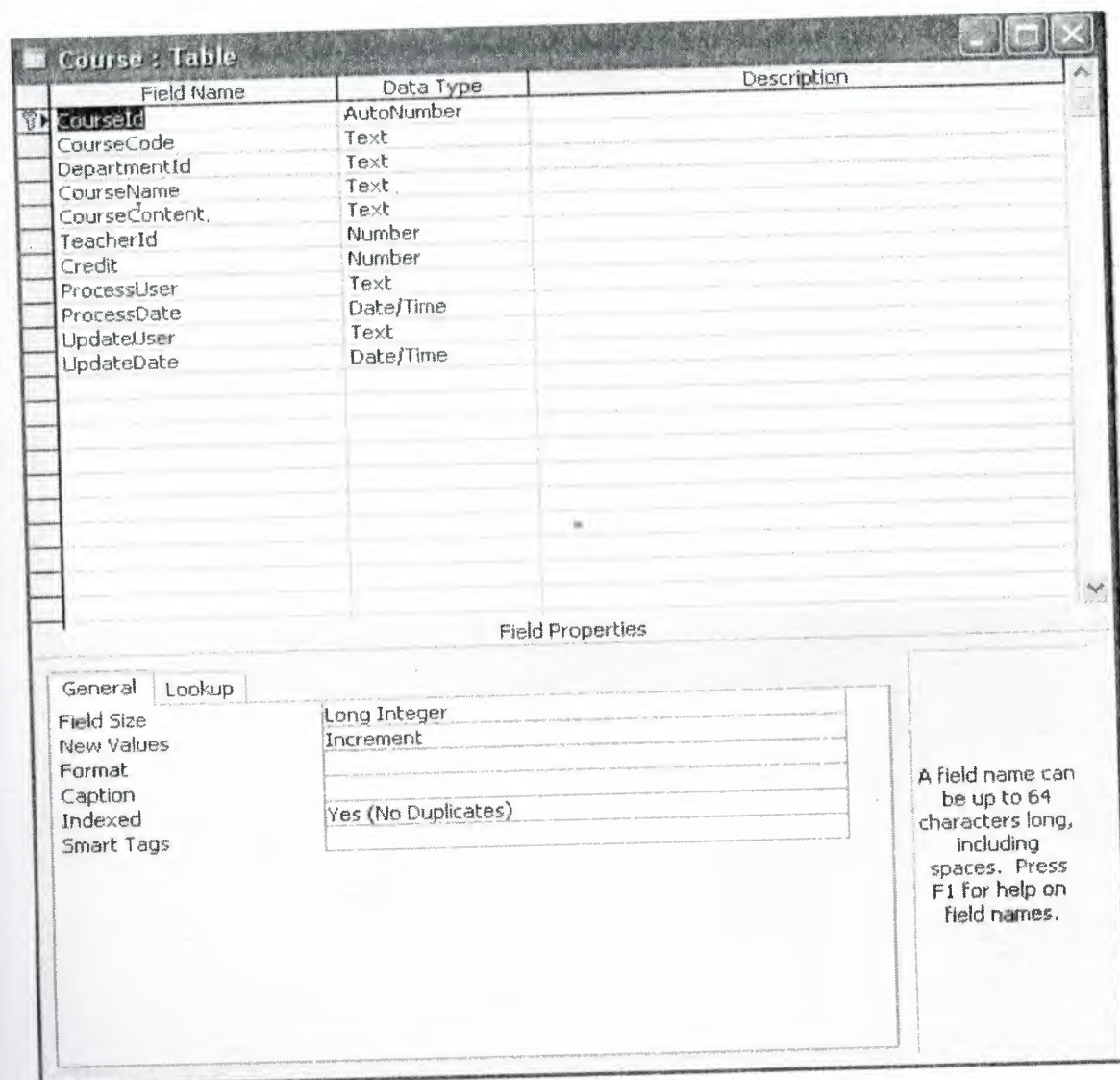
	Dismissdate Dismisscomment Processuser Updateprocess Updatedate	Date/time 20 text 20 text 20 text Date/time		Dismiss date Dismiss comment Process user Update process Update date
SE	courseId coursecode departmentId courseName coursecontent teacherId credit processuser processdate updateuser updatedate	auto number 10 text 10 text 30 text 40 text 2 number 2 number 20 text Date/time 20 text Date/time	PK FK1 FK2	Course id Course Code Department id Course Name Course Content Teacher id Course Credit Process user Process date Update process Update date
RTMENT	DepartmentId DeptName processuser processdate updateuser updatedate	10 text 50 30 text 20 text Date/time 20 text Date/time	PK	Department id Department Name Process user Process date Update process Update date
HER	TeacherId Tname Tsurname Address Gsm Sex Phone Email DepartmentId Birthdate Title processuser processdate updateuser	auto number 20 text 20 text 50 text 15 text 10 text 15 text 20 text 10 text Date/time 15 text 20 text Date/time 20 text	PK	Teacher id Teacher First name Teacher Last name Address Mobile phone Sex Home Phone Email address Department id Birth date Title Process user Process date Update process

	updatedate	Date/time		Update date
IN	LoginId	auto number	PK	Login id
	Loginame	25 text		User name
	Password	20 text		Password
	Accesslevel	15 text		Access level
	DepartmentId	10 text	FK1	Department id

Physical Database Description

In this part we have physical database design. Details table design of database is draw here. So, I will show tables of database with their details properties. Also the relationships of the tables will be shown.

Course :



Field Name	Data Type	Description
CourseId	AutoNumber	
CourseCode	Text	
DepartmentId	Text	
CourseName	Text	
CourseContent	Text	
TeacherId	Number	
Credit	Number	
ProcessUser	Text	
ProcessDate	Date/Time	
UpdateUser	Text	
UpdateDate	Date/Time	

Field Properties

General Lookup

Field Size: Long Integer

New Values: Increment

Format:

Caption:

Indexed: Yes (No Duplicates)

Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Department:

The screenshot displays the Microsoft Access interface. At the top, a table named 'Department' is shown with the following fields:

Field Name	Data Type	Description
DepartmentId	Text	
DepartmentName	Text	
ProcessUser	Text	
ProcessDate	Date/Time	
UpdateUser	Text	
UpdateDate	Date/Time	

Below the table, the 'Field Properties' task pane is open, showing the 'General' tab. The 'Field Size' property is set to 50. Other properties listed include Format, Input Mask, Caption, Default Value, Validation Rule, Validation Text, Required, Allow Zero Length, Indexed, Unicode Compression, IME Mode, IME Sentence Mode, and Smart Tags.

A text box on the right side of the task pane provides a note: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names."

Login : Table

Field Name	Data Type	Description
LoginId	AutoNumber	
LoginName	Text	
PassWord	Text	
AccessLevel	Text	
DepartmentId	Text	

Field Properties

General | Lookup

Field Size: Long Integer
 New Values: Increment
 Format:
 Caption:
 Indexed: Yes (No Duplicates)
 Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Student :

Field Name	Data Type	Description
StudentId	Text	
DepartmentId	Text	
name	Text	
surname	Text	
sex	Text	
Address	Text	
City	Text	
Country	Text	
postcode	Text	
Gsm	Text	
FatherName	Text	
MotherName	Text	
BirthPlace	Text	
BirthDate	Date/Time	
QsymNo	Text	
Email	Text	
RegisterDate	Date/Time	
phone	Text	
isGraduated	Yes/No	
GraduationDate	Date/Time	
GraduationComment	Text	

Field Properties

General	Lookup
Field Size	8
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Teacher:

Teacher : Table

Field Name	Data Type	Description
TeacherId	AutoNumber	
Tname	Text	
Tsurname	Text	
Address	Text	
Gsm	Text	
Sex	Text	
Phone	Text	
Email	Text	
DepartmentId	Text	
BirthDate	Date/Time	
Title	Text	
ProcessUser	Text	
ProcessDate	Date/Time	
UpdateUser	Text	
UpdateDate	Date/Time	

Field Properties

General **Lookup**

Field Size: Long Integer

New Values: Increment

Format:

Caption:

Indexed: Yes (No Duplicates)

Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Transcript :

Field Name	Data Type	Description
TransId	AutoNumber	
CourseId	Number	
StudentId	Text	
CourseRegDate	Date/Time	
Grade	Text	
GradeDate	Date/Time	
AddDate	Date/Time	
DropDate	Date/Time	
ProcessUser	Text	
ProcessDate	Date/Time	
UpdateUser	Text	

Field Properties

General Lookup

Field Size Long Integer

New Values Increment

Format

Caption

Indexed Yes (No Duplicates)

Smart Tags

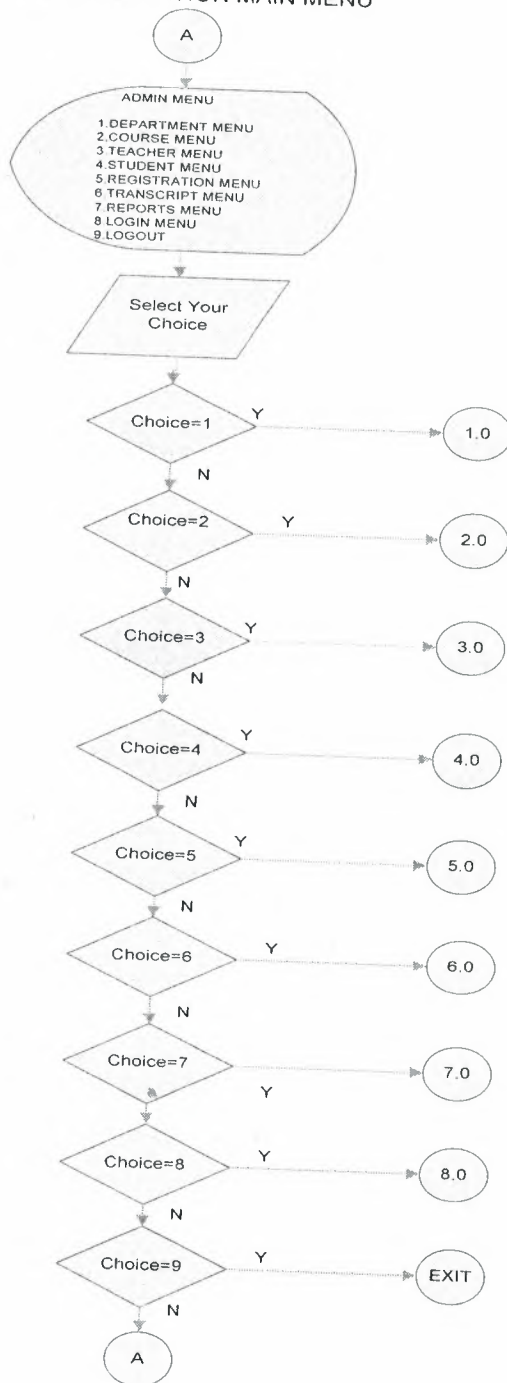
A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

USER MANUEL:

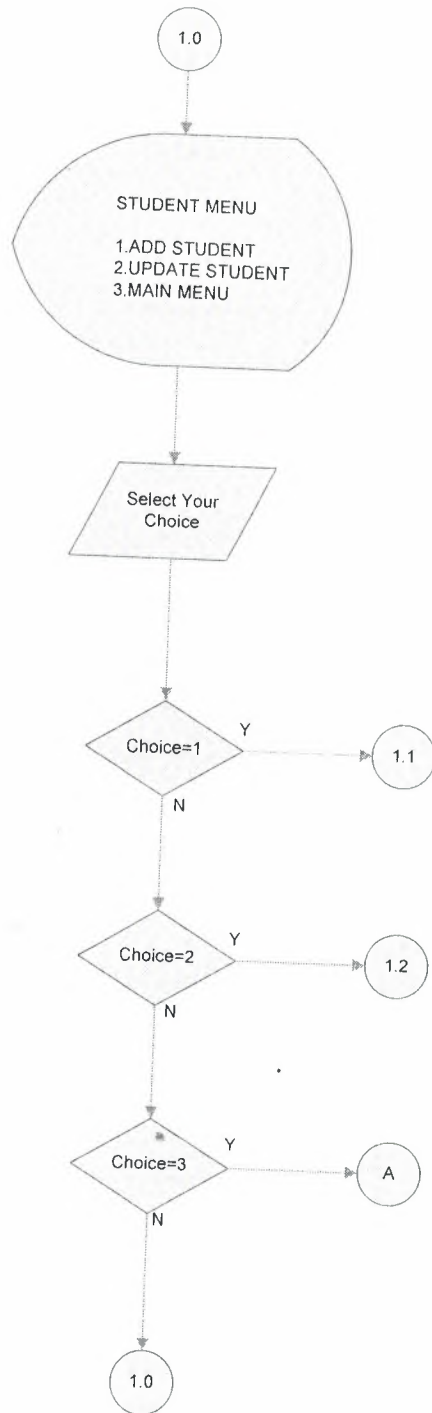
adham yab.

USER FLOW CHARTS

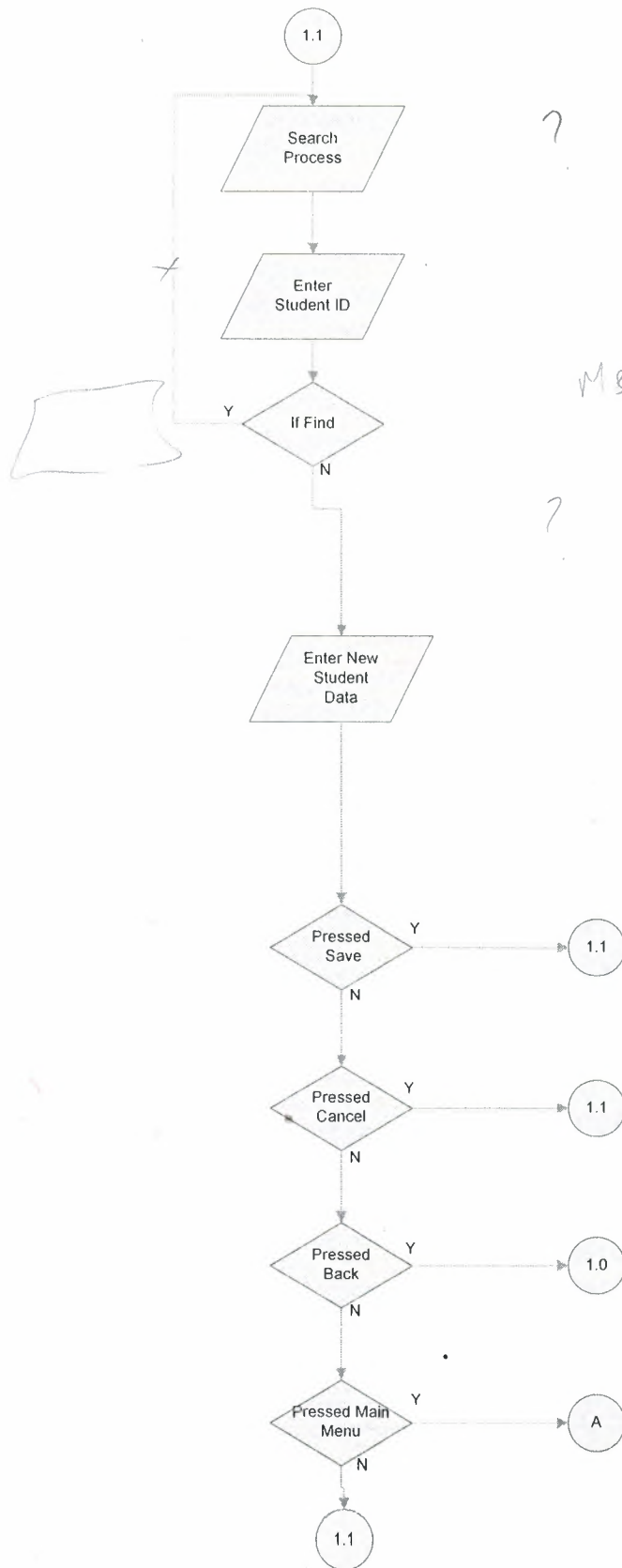
ADMINISTRATION MAIN MENU



STUDENT MENU



NEW STUDENT
SUBPROCESS

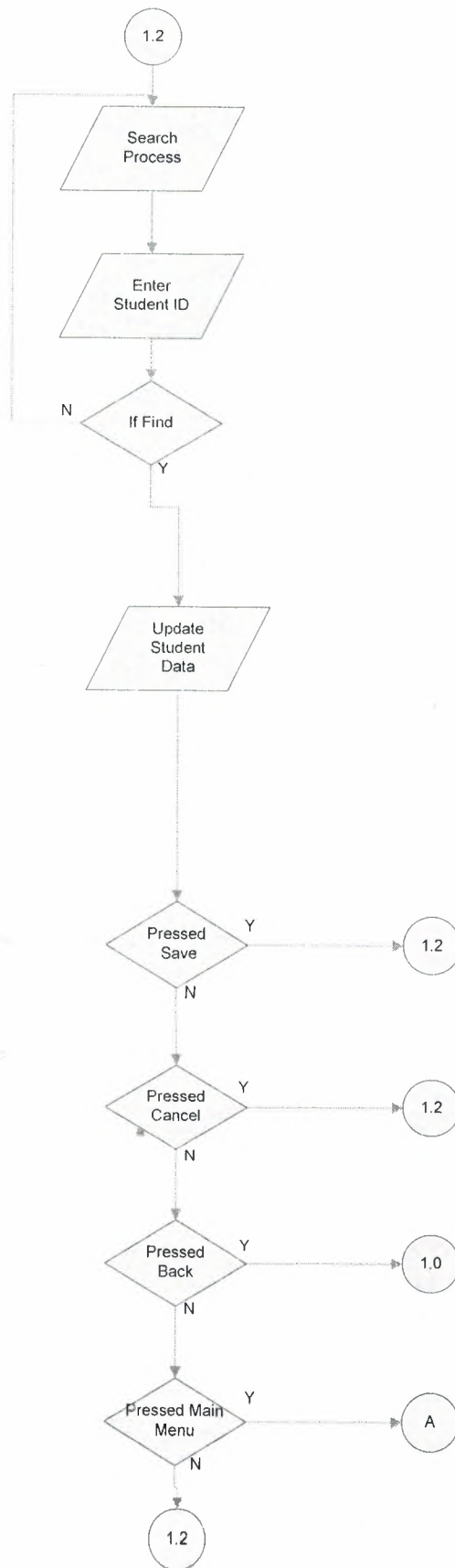


?

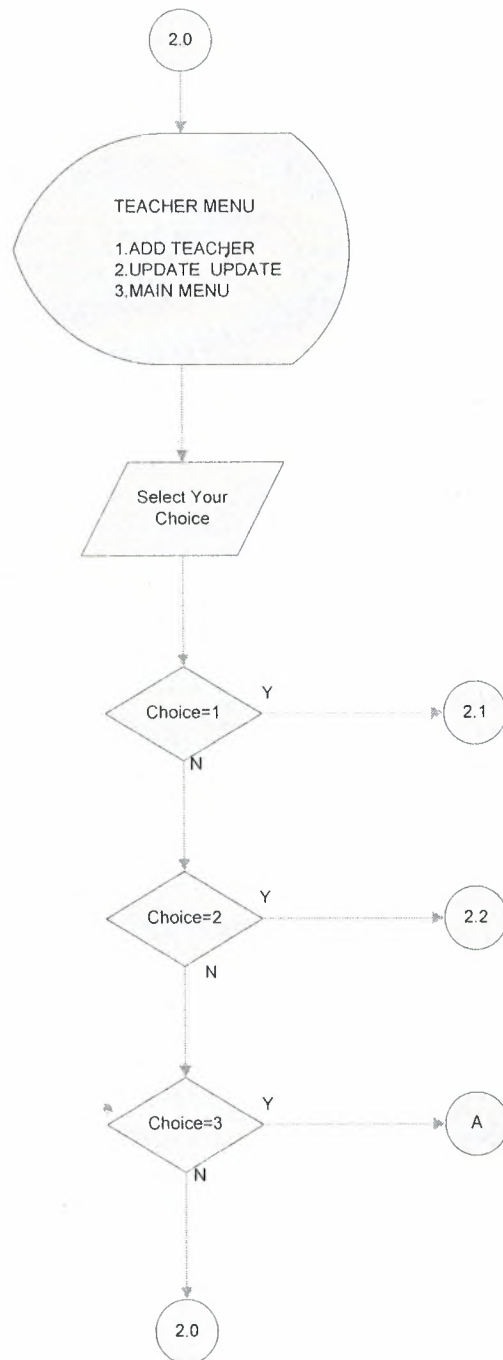
Message:
Student is not
found.

?

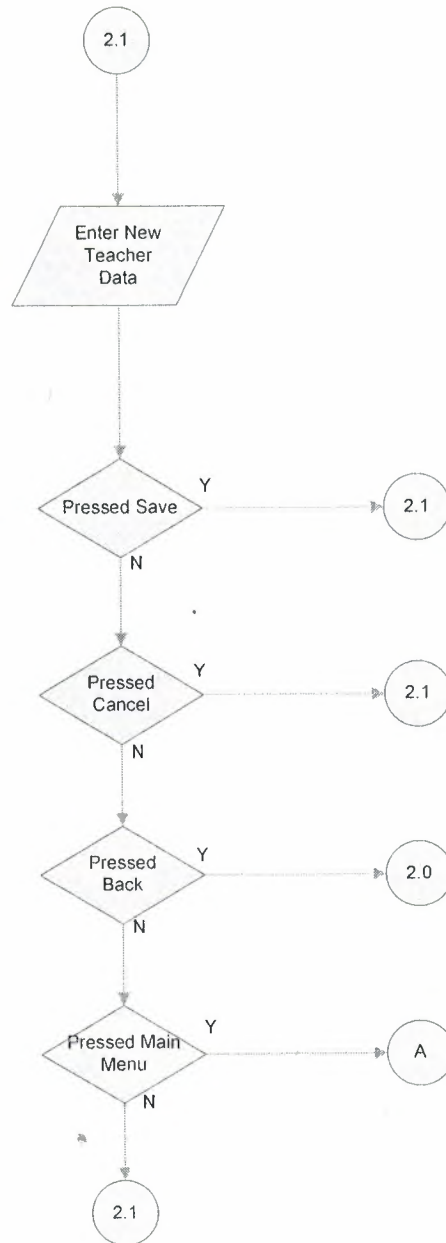
UPDATE STUDENT
SUBPROCESS



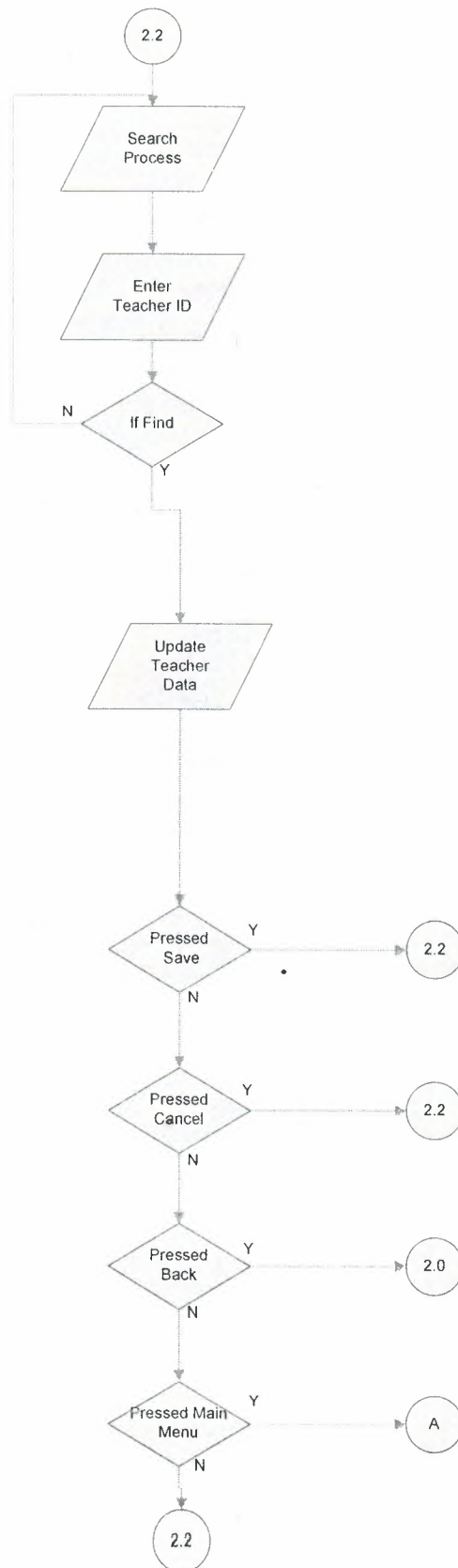
TEACHER MENU



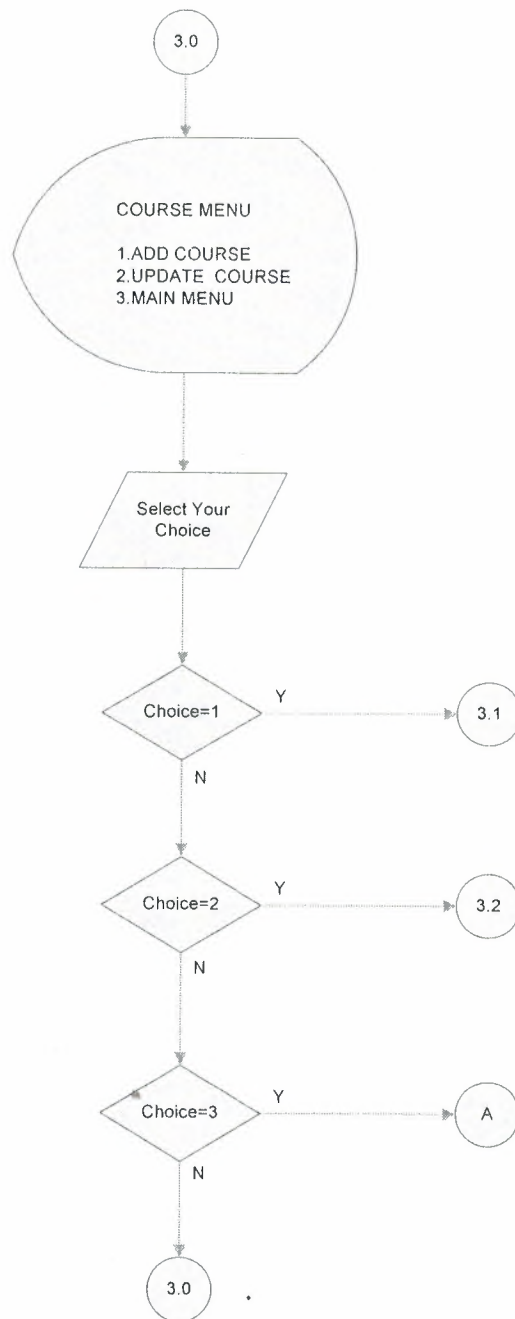
NEW TEACHER
SUBPROCESS



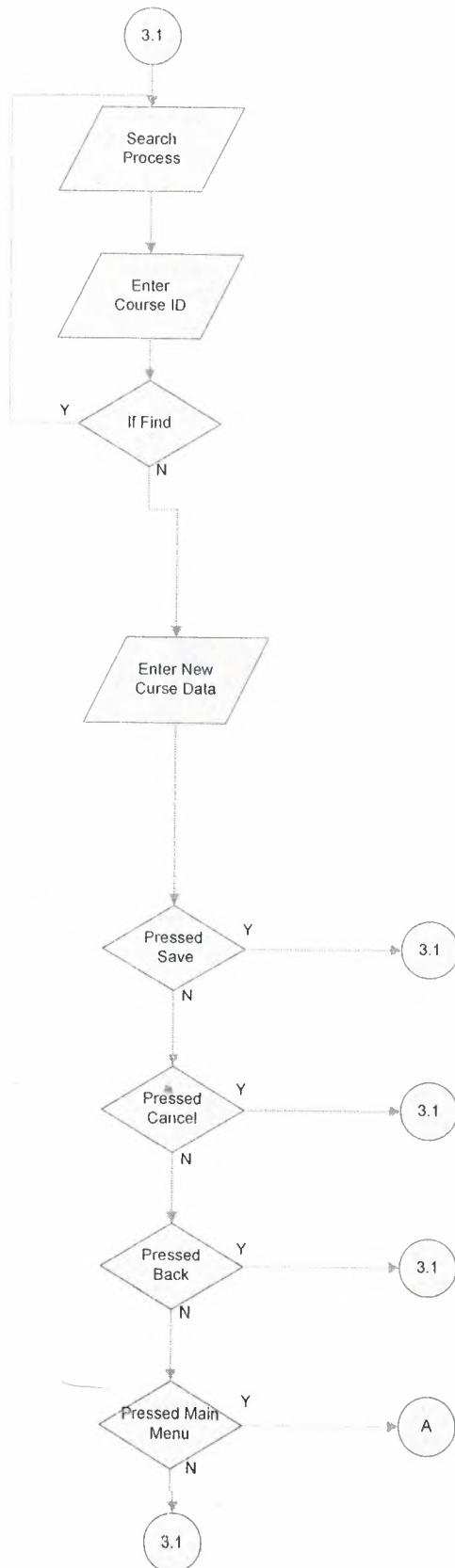
UPDATE TEACHER
SUBPROCESS



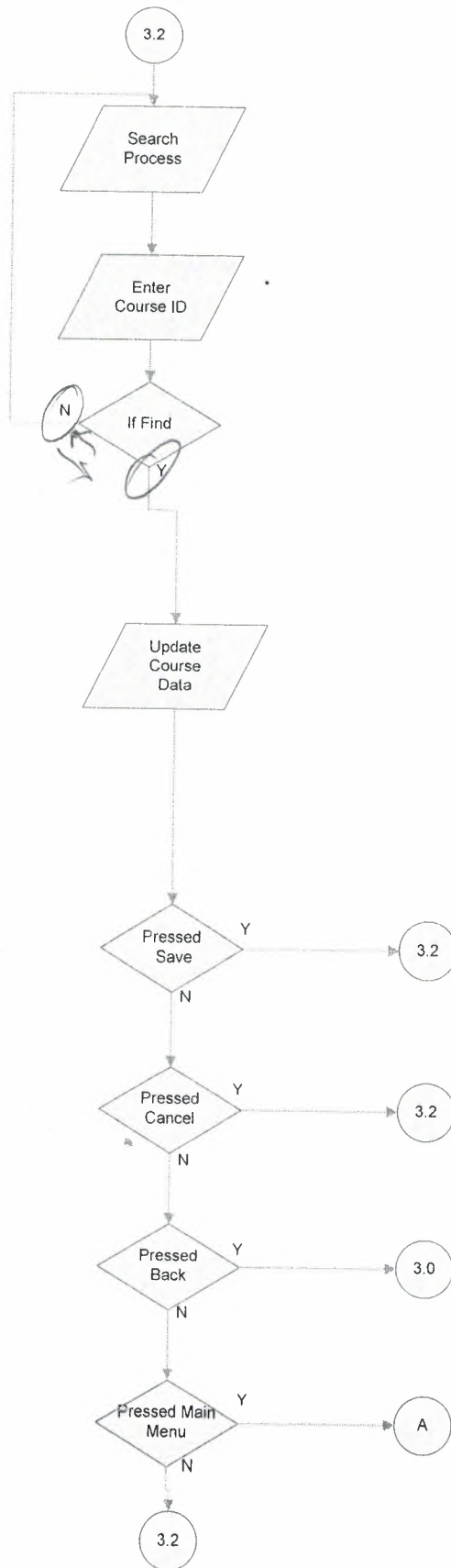
COURSE MENU



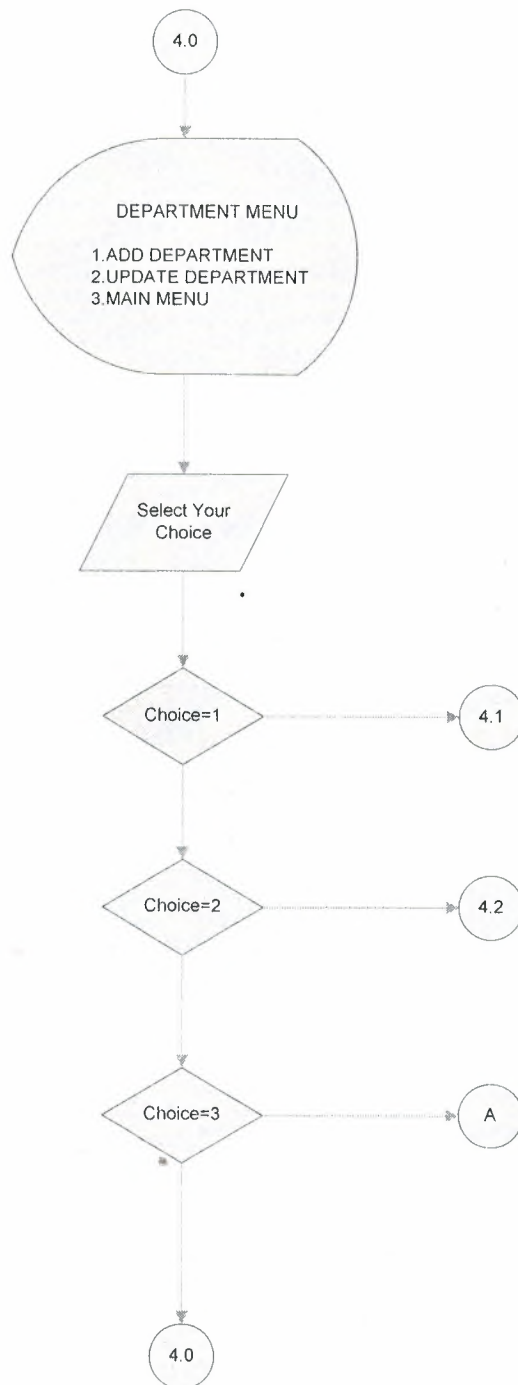
NEW COURSE
SUBPROCESS



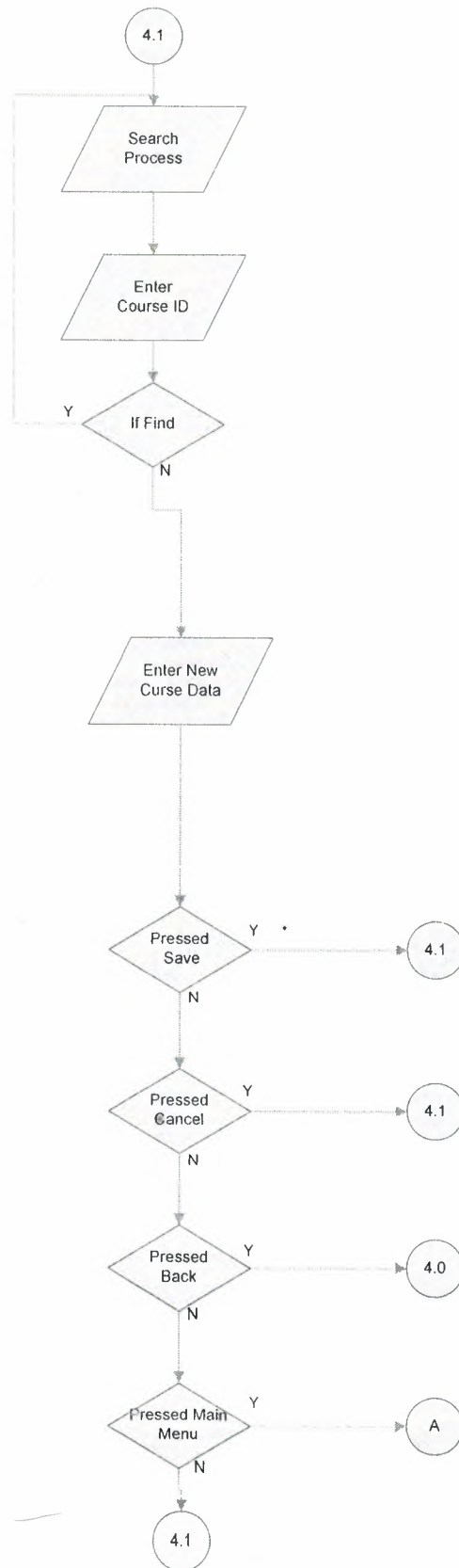
UPDATE COURSE
SUBPROCESS



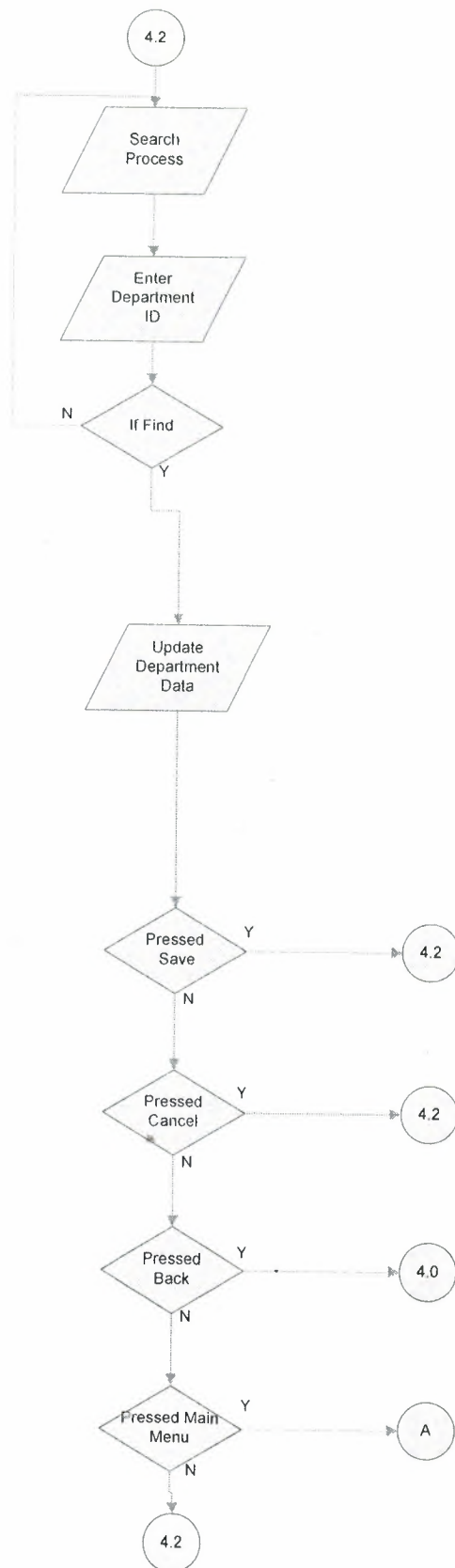
DEPARTMENT MENU



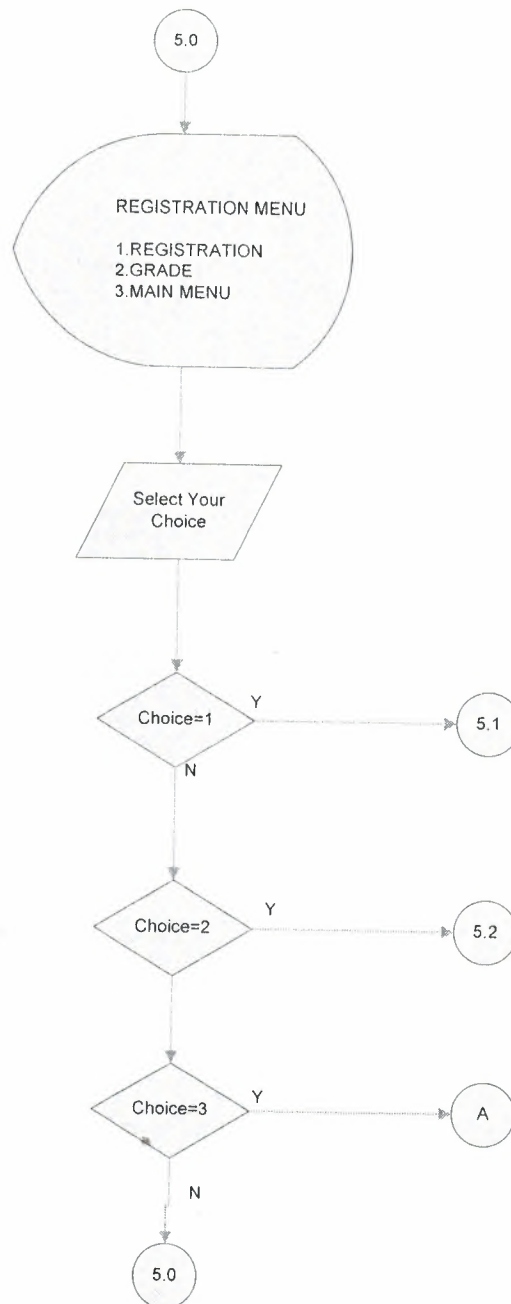
NEW DEPARTMENT
SUBPROCESS



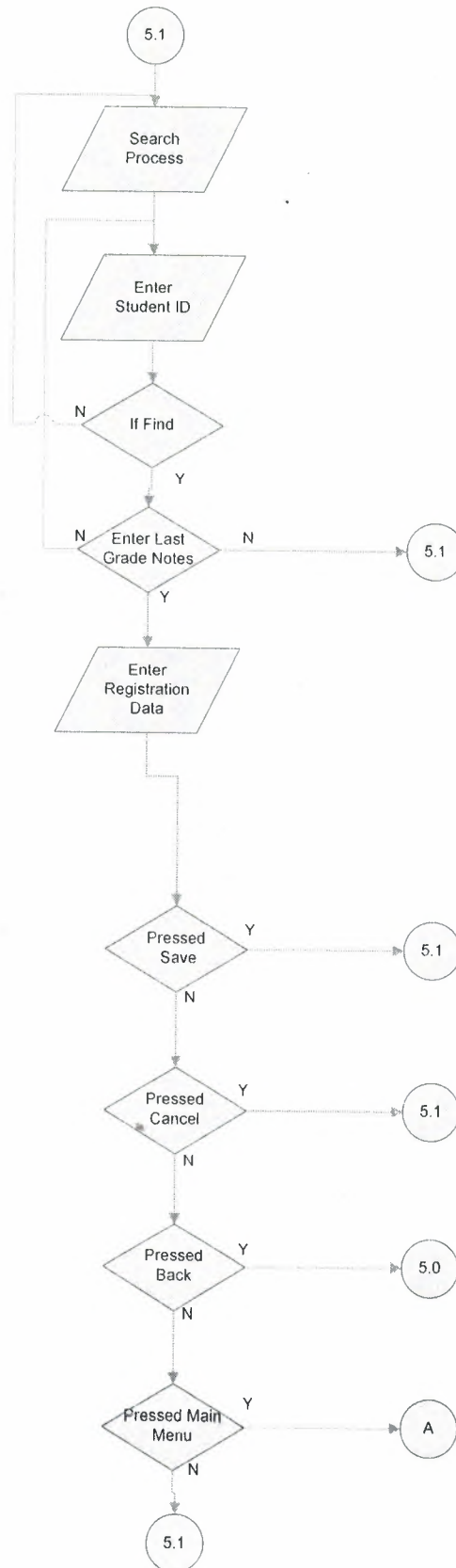
UPDATE DEPARTMENT
SUBPROCESS



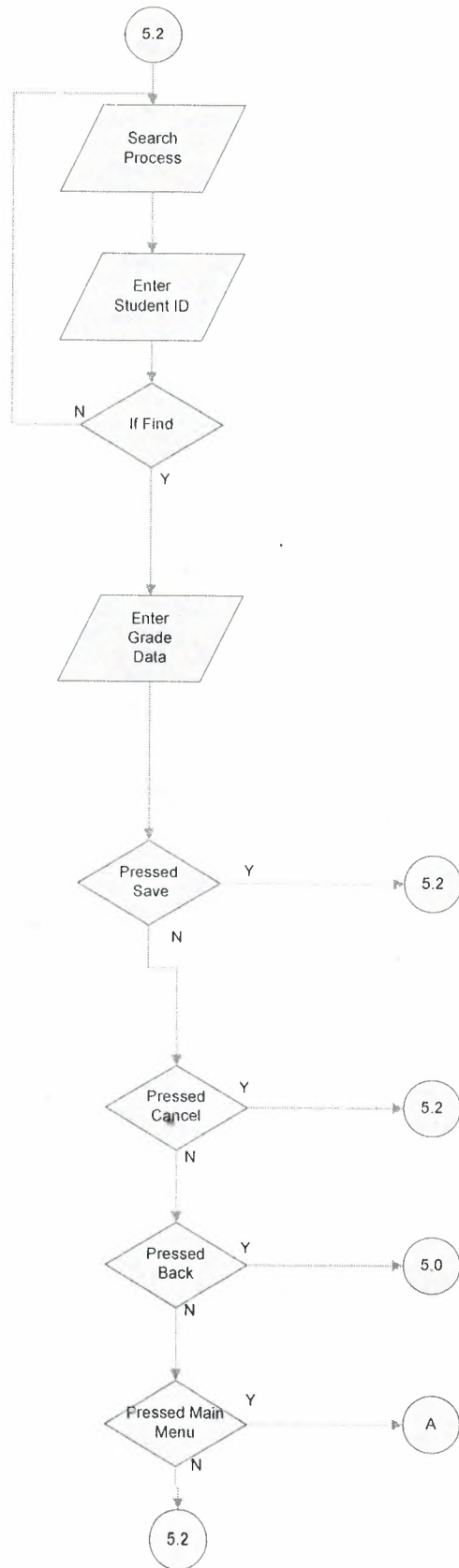
REGISTRATION MENU



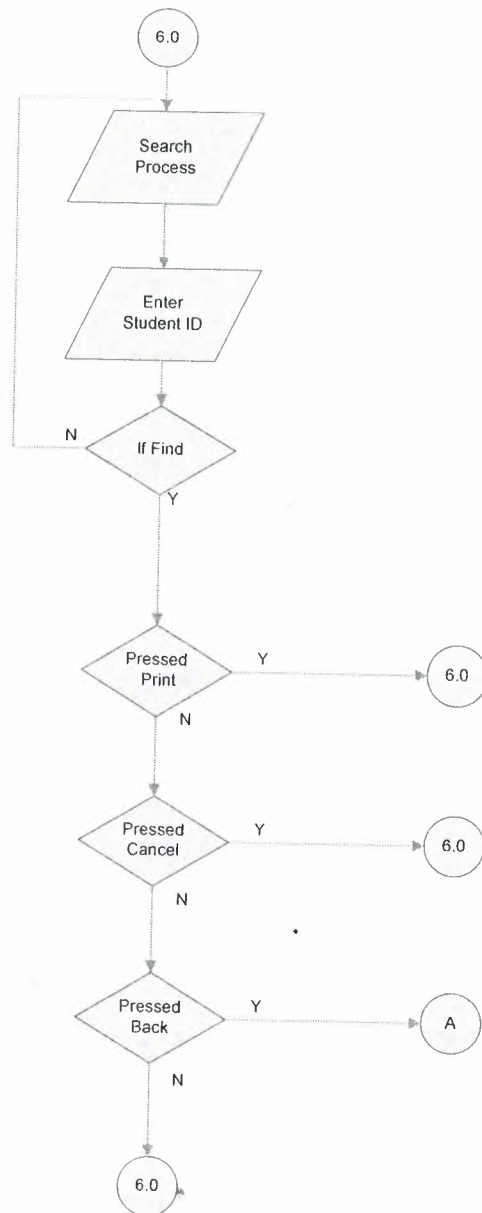
REGISTRATION
SUBPROCESS



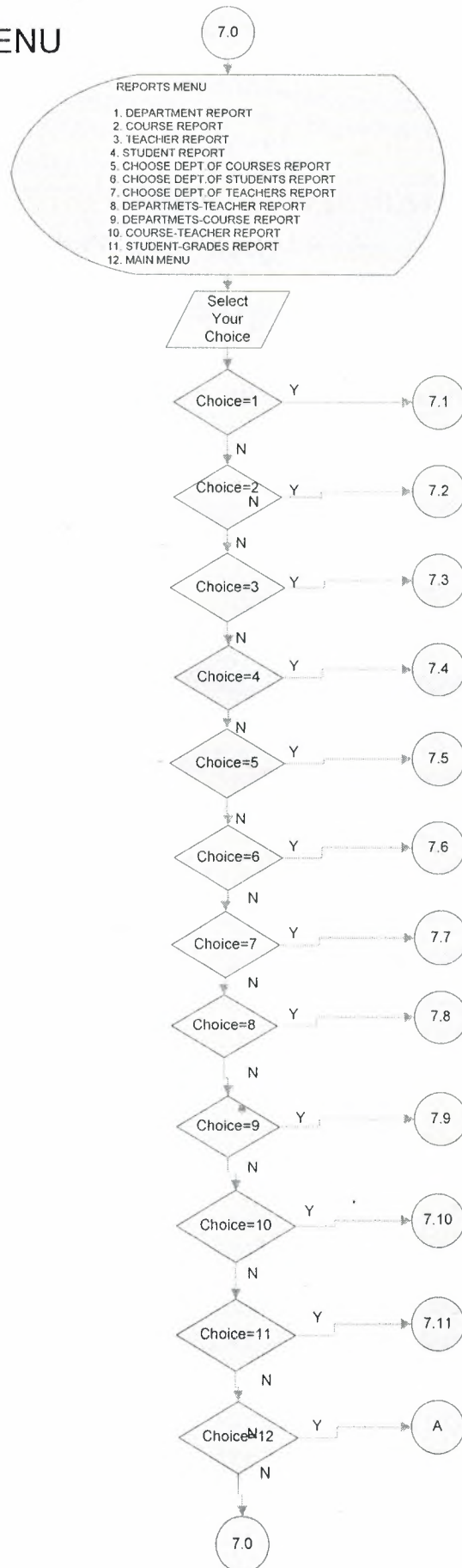
GRADING SUBPROCESS



TRANSCRIPT MENU



REPORTS MENU



References

1. Database Systems a practical approach to design, implementation, ad managment 2nd edition, Thomas M. Connolly, Carolyn E.Begg; Addison Wesley Longman Limited, 1999.
2. Modern System Analysis and Design, 3rd edition, Jeffrey Hoffer, Joey George, and Joseph Valacich, Prentice Hall, 2002.
3. Software Engineering, 6th edition, Ian Sommerville, Addison Wesley, 2001.
4. Database System Concepts. 4th edition, Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, McGraw-Hill, 2002.

APPENDIX I

SCREEN OUTPUT

Access Control:

The screenshot displays a web-based access control interface for Near East University. The window title is "ACCESS CONTROL". The header features the university's name "NEAR EAST UNIVERSITY" flanked by two circular logos. On the left, a "LOGIN" section contains input fields for "User Name" (with the value "1") and "Password" (with a masked value "2"), and "ENTER" and "EXIT" buttons. The main content area shows a large circular logo, the text "Near East University", and "Welcome to Registration System.". The taskbar at the bottom shows several open applications: "Başlat", "Document1 - Microsoft ...", "mezuniyet - Microsoft Word", "Flow Chart", and "Project1". The system clock indicates the time is 04:26 on the MP3 (D:) drive.

ACCESS CONTROL

NEAR EAST UNIVERSITY

LOGIN

User Name : 1

Password : 2

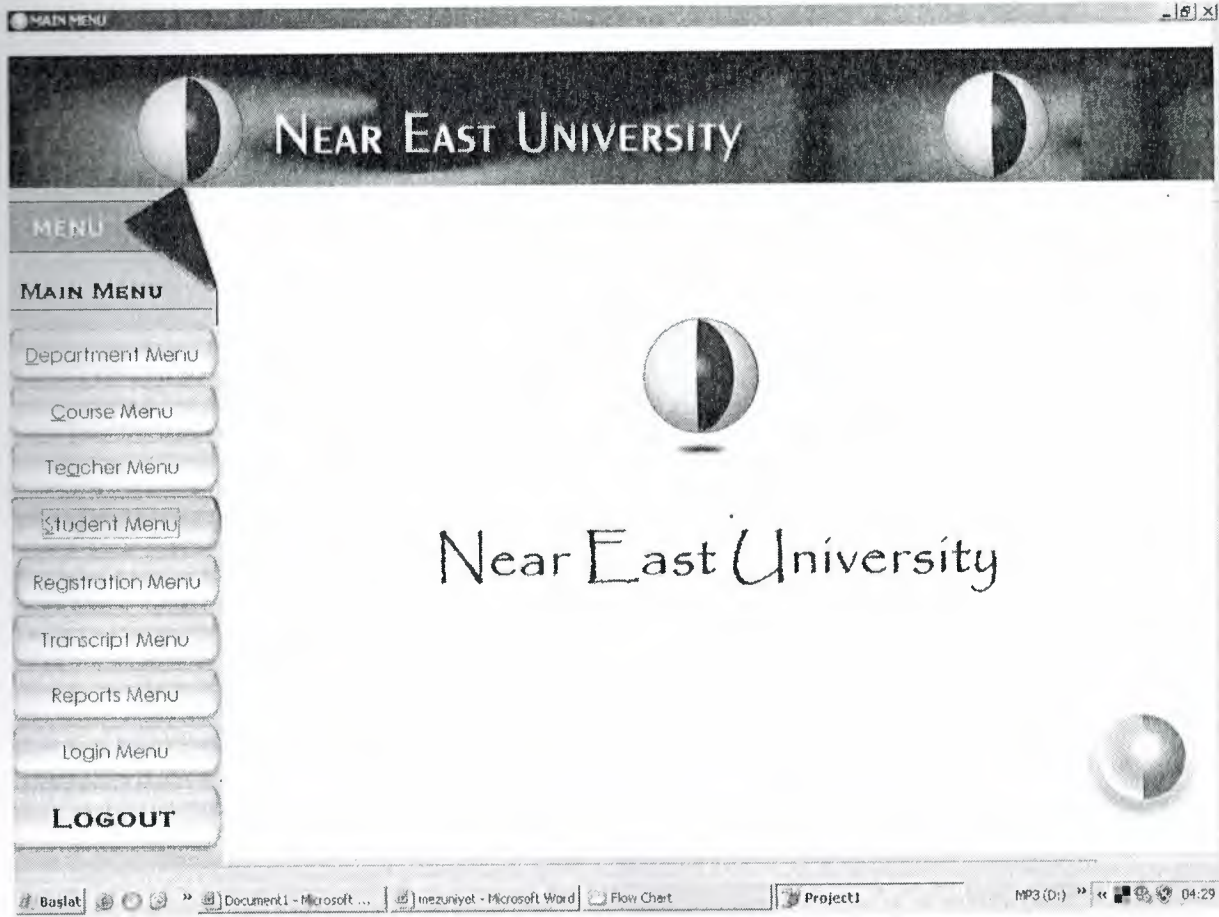
ENTER EXIT

Near East University

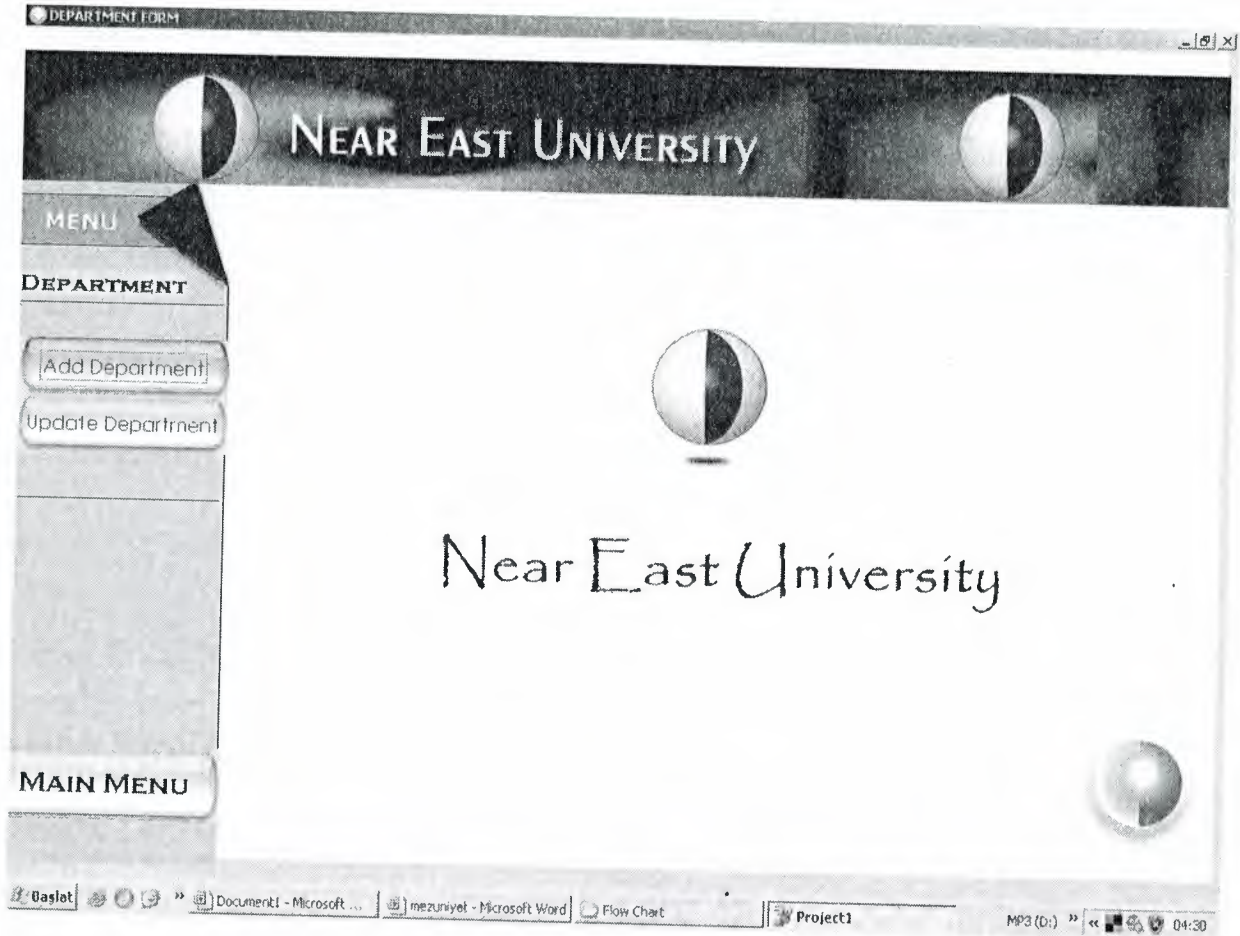
Welcome to Registration System.

Başlat Document1 - Microsoft ... mezuniyet - Microsoft Word Flow Chart Project1 MP3 (D:) 04:26

Main menu:



Department Menu:



Add Department:

ADD DEPARTMENT FORM

NEAR EAST UNIVERSITY

MENU

ADD DEPARTMENT

Department ID :

Department Name :

Search

Save

Cancel

<< BACK

DepartmentId	DepartmentName
ACC	ACCOUNTING
ARC	ARCHITECTURE
BESYO	BEDEN EGITIM SPOR MESLEK
BF	BANKING AND FINANCE

DEPARTMENT MENU

MAIN MENU

Baslat

Document1 - Microsoft ...

mezuniyet - Microsoft Word

Flow Chart

Project1

MP3 (D:) »

04:32

Update Department:

UPDATE DEPARTMENT FORM

NEAR EAST UNIVERSITY

MENU

UPDATE DEPARTMENT

Department ID :

Department ID :

Department Name :

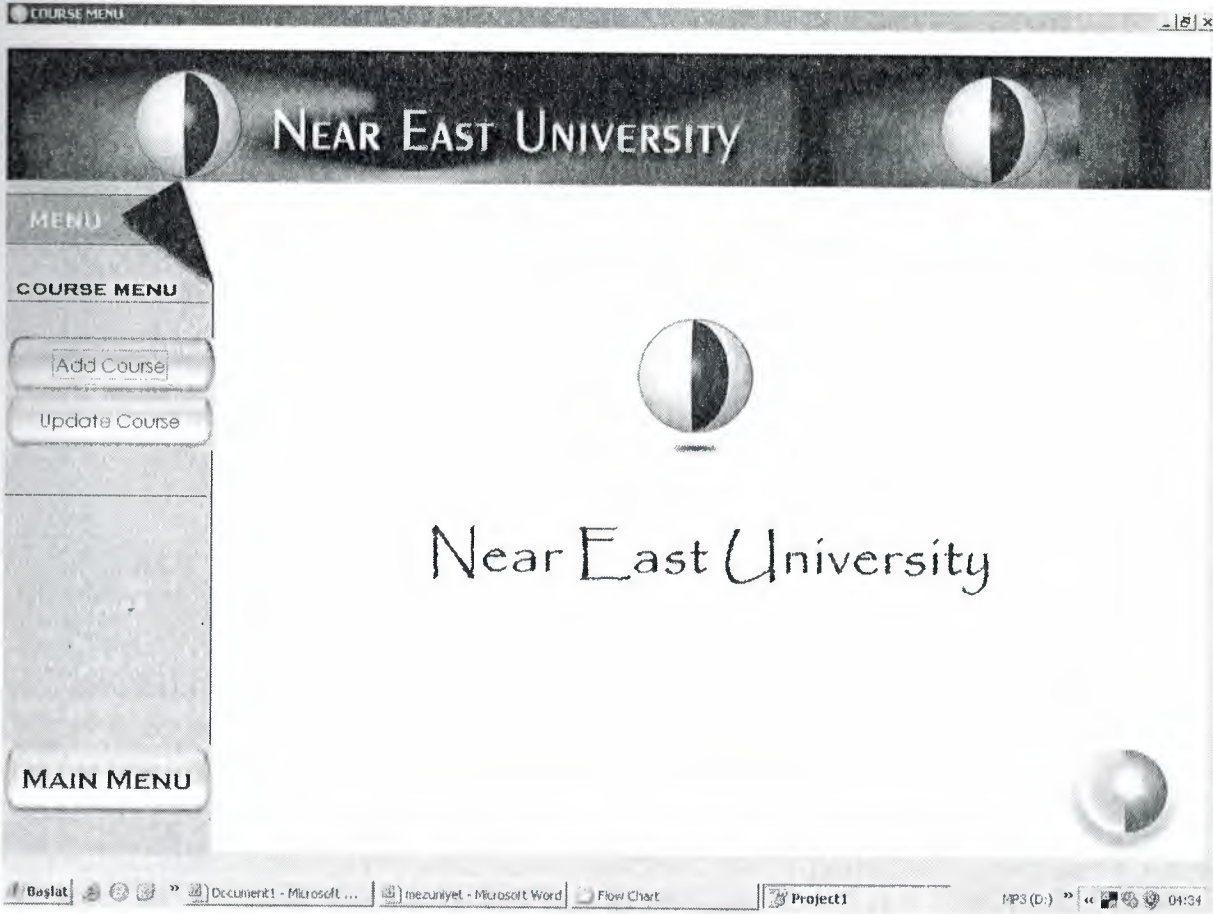
DepartmentId	DepartmentName
ACC	ACCOUNTING
ARC	ARCHITECTURE
BESYO	BEDEN EGITIM SPOR MESLEK
BF	BANKING AND FINANCE

DEPARTMENT MENU

MAIN MENU

Baglat Document1 - Microsoft ... mezunlyst - Microsoft Word Flow Chart Project1 NP3 (D:) 09:33

Course Menu:



Add Course:

ADD COURSE FORM

NEAR EAST UNIVERSITY

MENU

ADD COURSE

COURSE MENU

MAIN MENU

ADD COURSE

Course Code

Course Name

Course Content

Department ID

Teacher ID

Course Credit

Save

Cancel

<< BACK

Course List

CourseId	CourseCode	DepartmentId	CourseName
1	MAN404	CIS	INTRODUCTION
2	MAN401	MAN	SMALL BUSINESS
3	MAN303	MAN	MARKETING
4	MAN306	MAN	MARKETING 2
5	CIS412	CIS	MIS
6	CIS240	CIS	DELPHI

Teacher List

Project1

MP3 (D:) 04:35

Update Course:

UPDATE COURSE INFORMATION

NEAR EAST UNIVERSITY

UPDATE COURSE

MENU

UPDATE COURSE

COURSE MENU

MAIN MENU

CourseCode

Course Name **Teacher ID**

Course Content **Course Credit**

Department ID

Course List

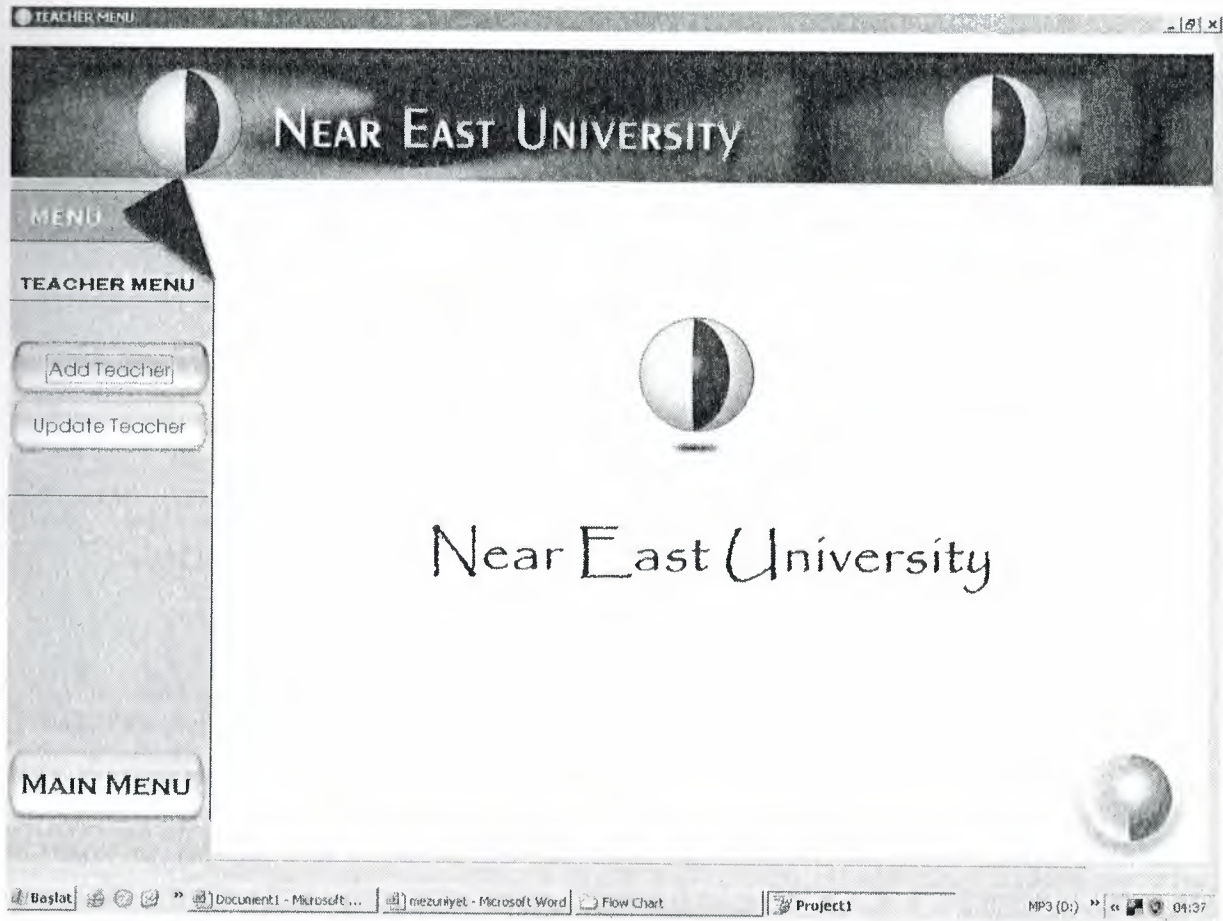
CourseId	CourseCode	DepartmentId
1	MAN404	CIS
2	MAN401	MAN
3	MAN303	MAN
4	MAN306	MAN

Teacher List

TeacherId	TeacherName
-----------	-------------

Baslat Document1 - Microsoft Word mezuniyet - Microsoft Word Flow Chart Project1 MP3 (D:) 04:37

Teacher Menu:



Add Teacher:

ADD TEACHER

NEAR EAST UNIVERSITY

MENU

ADD TEACHER

Teacher ID 27

Name *

Surname *

Birth Date * 17/1/1970

Gender

Phone

GSM No

Address

E-Mail

Department ID *

Title *

Save

Cancel

<< BACK

TEACHER MENU

MAIN MENU

Bosfat Document1 - Microsoft Word mezuniyet - Microsoft Word Flow Chart Project1 MP3 (D:) 04:40

Update Teacher:

UPDATE TEACHER INFORMATION

NEAR EAST UNIVERSITY

MENU

UPDATE TEACHER

TEACHER MENU

MAIN MENU

UPDATE TEACHER

TeacherID
Name
Surname

Search

Name *
Surname *
Birth Date *
Gender
Phone,
GSM No

Address
E-Mail
Department ID *
Title *

Save
Cancel
<< BACK

Başlat

Document1 - Microsoft ...

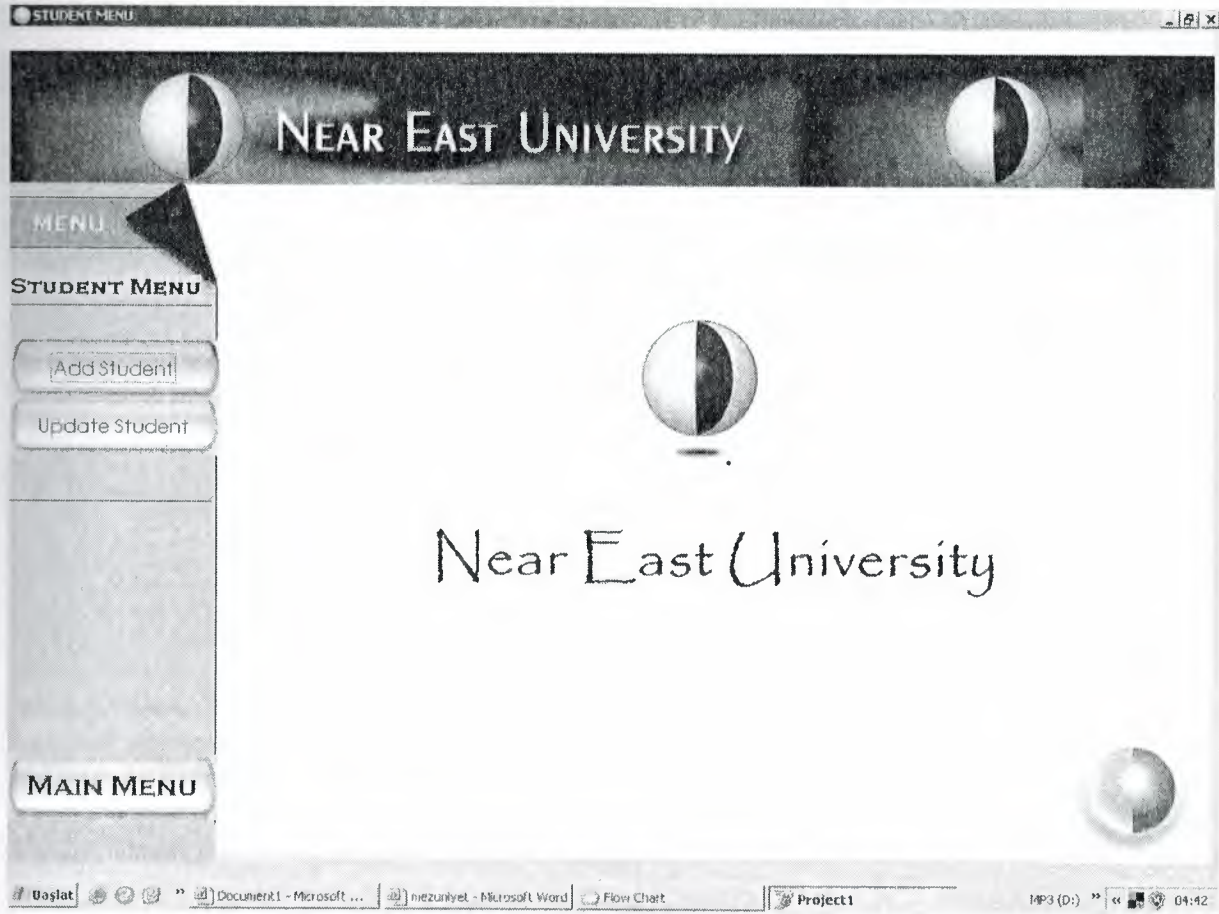
mezuniyet - Microsoft Word

Flow Chart

Project1

MP3 (D:) 04:41

Student Menu:



Add Student:

ADD STUDENT FORM

NEAR EAST UNIVERSITY

MENU

ADD STUDENT

STUDENT MENU

MAIN MENU

ADD STUDENT

Student ID :

Name *

Surname *

Place of Birth

Birth Date *

Gender

Mother Name

Father Name

Address

City

Country *

Postal Code

Phone

Gsm

E-mail

Osym No

Department *

Save

Cancel

<< BACK

Update Student:

UPDATE STUDENT INFORMATION

NEAR EAST UNIVERSITY

MENU

UPDATE STUDENT

Student ID : Search

Name *	City
Surname *	Country *
Place of Birth	Postal Code
Birth Date *	Phone
Gender	Gsm
Mother Name	E-mail
Father Name	Ceyin No
Address	Department *

Save

Cancel

<< BACK

STUDENT MENU

MAIN MENU

Baslat

Document1 - Microsoft ...

mezuniyet - Microsoft Word

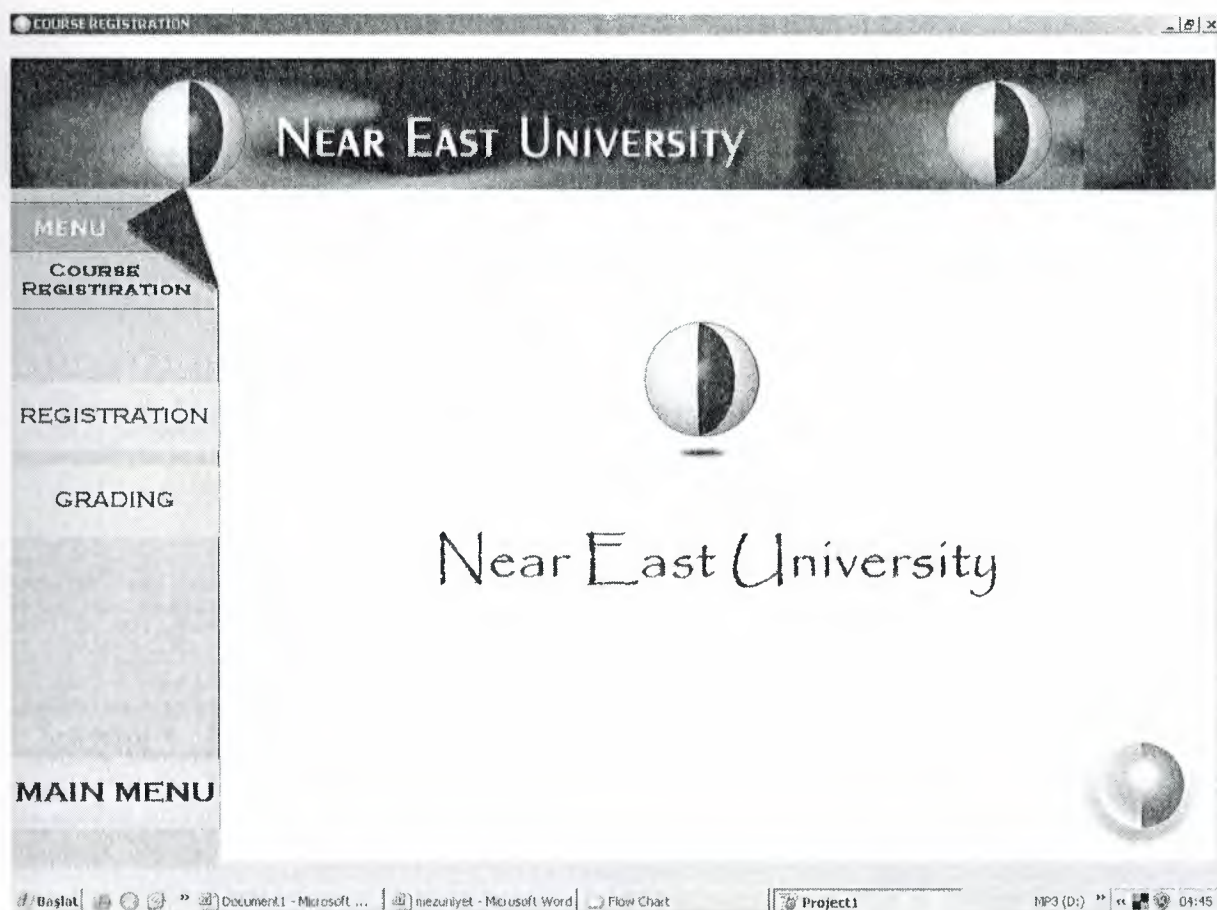
Flow Chart

Proje1

MP3 (D:)

03:44

Registration Menu:



[illegible]

Grading:

GRADING

NEAR EAST UNIVERSITY

MENU

GRADING

GRADING

Student No Search

Student Name

Student Surname

Department

REGISTRATION MENU

MAIN MENU

<< BACK Save Cancel

Baslat Document1 - Microsoft ... mezuniyet - Microsoft Word Flow Chart Project1 MP3 (D:) 01:47

Transcript Menu:

Transcript Form

NEAR EAST UNIVERSITY

MENU

TRANSCRIPT MENU

MAIN MENU

TRANSCRIPT MENU

Student Id

Student ID: 20071980
Student Name: MEHMET YUSUF
Student Surname: AKKILIC
Date Of Birth: 12/09/1980
Student Department: CIS

CGPA : 3.22

2007 - Fall Term	2007 - Spring Term	2008 - Fall Term	2008 - Spring Term																																				
<table border="1"> <tr><td>CIS101</td><td>AA</td></tr> <tr><td>ECON101</td><td>CR</td></tr> <tr><td>ENG101</td><td>BA</td></tr> <tr><td>CR131</td><td>RR</td></tr> <tr><td>CR161</td><td>CR</td></tr> </table>	CIS101	AA	ECON101	CR	ENG101	BA	CR131	RR	CR161	CR	<table border="1"> <tr><td>CIS201</td><td>CB</td></tr> <tr><td>MAN308</td><td>BA</td></tr> <tr><td>CIS331</td><td>DC</td></tr> <tr><td>CIS347</td><td>DC</td></tr> <tr><td>CIS305</td><td>BB</td></tr> </table>	CIS201	CB	MAN308	BA	CIS331	DC	CIS347	DC	CIS305	BB	<table border="1"> <tr><td>MAN401</td><td>AA</td></tr> <tr><td>MAN303</td><td>AA</td></tr> <tr><td>CIS412</td><td>BA</td></tr> <tr><td>MAN403</td><td>BA</td></tr> <tr><td>ENG102</td><td>FF</td></tr> </table>	MAN401	AA	MAN303	AA	CIS412	BA	MAN403	BA	ENG102	FF	<table border="1"> <tr><td>MAN401</td><td>AA</td></tr> <tr><td>MAN306</td><td>CC</td></tr> <tr><td>COM318</td><td>CB</td></tr> </table>	MAN401	AA	MAN306	CC	COM318	CB
CIS101	AA																																						
ECON101	CR																																						
ENG101	BA																																						
CR131	RR																																						
CR161	CR																																						
CIS201	CB																																						
MAN308	BA																																						
CIS331	DC																																						
CIS347	DC																																						
CIS305	BB																																						
MAN401	AA																																						
MAN303	AA																																						
CIS412	BA																																						
MAN403	BA																																						
ENG102	FF																																						
MAN401	AA																																						
MAN306	CC																																						
COM318	CB																																						
GPA 3.00	GPA 2.40	GPA 3.00	GPA 2.83																																				

2007 - Spring Term																	
<table border="1"> <tr><td>MAN401</td><td>AA</td></tr> <tr><td>CIS240</td><td>AA</td></tr> <tr><td>CIS244</td><td>AA</td></tr> <tr><td>CIS347</td><td>AA</td></tr> <tr><td>CIS244</td><td>AA</td></tr> <tr><td>CIS456</td><td>BA</td></tr> <tr><td>CIS132</td><td>AA</td></tr> </table>	MAN401	AA	CIS240	AA	CIS244	AA	CIS347	AA	CIS244	AA	CIS456	BA	CIS132	AA			
MAN401	AA																
CIS240	AA																
CIS244	AA																
CIS347	AA																
CIS244	AA																
CIS456	BA																
CIS132	AA																
GPA 3.56	GPA	GPA	GPA														

Baglat

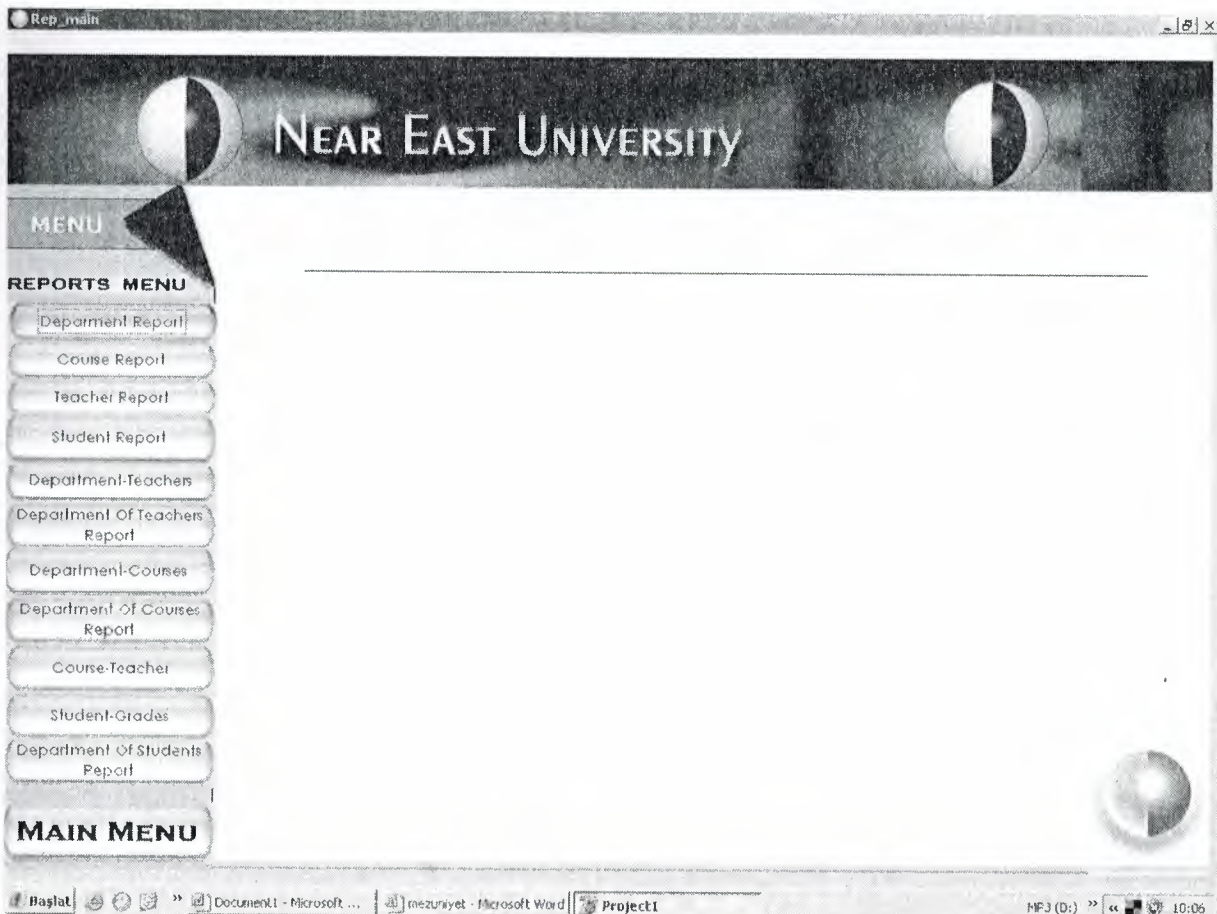
Document1 - Microsoft ...

mezuniyet - Microsoft Word

Project1

MP3 (D:) » 10:04

Reports Menu:



Department Report:

The screenshot displays a web application interface for Near East University. At the top, a banner features the university's name and two circular logos. Below the banner, a 'MENU' section is visible on the left, containing a 'REPORTS MENU' with various report options and a 'MAIN MENU' button. The main content area shows a window titled 'Rep all depts' with the heading 'LISTING ALL DEPARTMENTS'. This window contains a table with two columns: 'DepartmentId' and 'DepartmentName'. The table lists 15 departments, with 'AC' (Accounting) selected. Below the table, there are two buttons: 'Print This Report' and 'Close'. The bottom of the screenshot shows a Windows taskbar with several open applications and a system clock indicating 10:08.

NEAR EAST UNIVERSITY

MENU

REPORTS MENU

- Department Report
- Course Report
- Teacher Report
- Student Report
- Department-Teachers
- Department Of Teachers Report
- Department-Courses
- Department Of Courses Report
- Course-Teacher
- Student-Grades
- Department Of Students Report

MAIN MENU

Rep all depts

LISTING ALL DEPARTMENTS

DepartmentId	DepartmentName
AC	ACCOUNTING
ARC	ARCHITECTURE
BESİD	BEDEN EGİTİM SPOR MESLEK
BF	BANKING AND FINANCE
BIO	BIOLOGY 2
CENG	COMPUTER ENGINEERING
CHEM	CHEMISTRY
CIS	COMPUTER INFORMATION SYSTEM
COM	COMPUTER ENG
ECON	ECONOMY
ELI	ENGLISH LEARNING TEACHINGS
IE	INDUSTRIAL ENGINEERING
MAN	BUSINESS ADMINISTRATION

Print This Report **Close**

Windows taskbar: Başlat, Document1 - Microsoft..., mezuniyet - Microsoft Word, Project1, MP3 (D:), 10:08

Course Report:

The screenshot displays a web application for Near East University. The main header features the university's name, "NEAR EAST UNIVERSITY", flanked by two circular logos. Below the header, a "MENU" section on the left lists various reports under "REPORTS MENU" and a "MAIN MENU". The central area, titled "Listing All Courses", contains a table with columns: CourseId, CourseCode, DepartmentId, and CourseName. The table lists 26 courses. At the bottom of the table area are two buttons: "Print This Page" and "Close". The bottom of the screenshot shows a Windows taskbar with several open applications and a system clock indicating 10:22.

NEAR EAST UNIVERSITY

MENU

REPORTS MENU

- Department Report
- Course Report
- Teacher Report
- Student Report
- Department-Teachers
- Department Of Teachers Report
- Department-Courses
- Department Of Courses Report
- Course-Teacher
- Student-Grades
- Department Of Students Report

MAIN MENU

All Courses

Listing All Courses

CourseId	CourseCode	DepartmentId	CourseName
29	ARC		ASD
23	COM318	CIS	DATA COMMUNICATION
27	CIS243	CIS	DATA STRUCTURE
12	CIS244	CIS	DATA STRUCTURES
6	CIS240	CIS	DELPHI
10	ECON101	ECON	ECONOMY
7	ENG101	MAN	ENGLISH
8	ENG102	MAN	ENGLISH 2
22	CIS347	CIS	HINDAS
15	CIS151	CIS	INT
25	CIS132	CIS	INTRO. COMPUTER
9	CIS131	MAN	INTRODUCTION
1	MAN404	CIS	INTRODUCTION
16	CIS456	CIS	JAVA
3	MAN303	MAN	MARKETING
4	MAN306	MAN	MARKETING 2
19	MAN403	MAN	MARKETING 3
26	MAT101	OS	MATEMATICS

Print This Page Close

Taskbar: Baglari, Document1 - Microsoft..., mezuniyet - Microsoft Word, Project1, MP3 (D:) 10:22

Teacher Report:

NEAR EAST UNIVERSITY

MENU

REPORTS MENU

- Department Report
- Course Report
- Teacher Report
- Student Report
- Department-Teachers
- Department Of Teachers Report
- Department-Courses
- Department Of Courses Report
- Course-Teacher
- Student-Grades
- Department Of Students Report

MAIN MENU

All Teachers

Listing All Teachers

Teacherid	Tname	Tsurname
1	YALCIN	AKCALI
2	ILHAM	HUSEYINOV
3	NADIRE	CAVUS
4	MUSTAFA	MENEKAY
5	TANSEL	DEVIN
6	AHMET	ERTUGAM
7	TULEN	SANER
8	ERDAL	BURYAY
9	ALI	MALEK
10	SERIFE	EYUPOGLU
11	MEHMET	AGA
12	AHMET	TERALI
13	CUMALI	SAHIN
14	CUMALI	SEN

Print This Page **Close**

MP3 (D:) 10:23

Student Report:

The screenshot shows a web application for NEAR EAST UNIVERSITY. The header features the university's name in a dark banner with two circular logos. On the left, a 'MENU' dropdown is open, showing a 'REPORTS MENU' with various report options and a 'MAIN MENU' button. The main content area displays a window titled 'All Students' with a table 'Listing All Students'. The table has three columns: 'StudentId', 'DepartmentId', and 'name'. It lists seven students, with the first one highlighted. Below the table are 'Print This Page' and 'Close' buttons. The Windows taskbar at the bottom shows the 'Başlat' button, open applications (Document1, mezuniyet, Project1), and the system clock (10:24).

NEAR EAST UNIVERSITY

MENU

REPORTS MENU

- Department Report
- Course Report
- Teacher Report
- Student Report
- Department-Teachers
- Department Of Teachers Report
- Department-Courses
- Department Of Courses Report
- Course-Teacher
- Student-Grades
- Department Of Students Report

MAIN MENU

All Students

Listing All Students

StudentId	DepartmentId	name
20071980	CIS	MEHMET YUSUF
20071981	CIS	CEMIL
20071985	MAN	NAZAN
20071986	MAN	OSMAN
20071987	IR	SERVET
20071988	CIS	SALIHA
20071989	CIS	MARIA

Print This Page **Close**

Başlat Document1 - Microsoft Word mezuniyet - Microsoft Word Project1 MF3 (Dr.) 10:24

Department – Teachers:

The screenshot displays a web application interface for NEAR EAST UNIVERSITY. The main header features the university's name in a stylized font. A left sidebar contains a 'MENU' section with various report options, including 'Department-Teachers', which is currently selected. The main content area shows a window titled 'Listing Departments' Teachers'. This window includes a dropdown menu for 'Choose A Department' set to 'MAN'. Below this is a table listing teachers with columns for 'Tname' and 'Tsurname'. The table contains six rows of data. At the bottom of the window are two buttons: 'Print This Page' and 'Close'. The taskbar at the bottom shows several open applications, including 'Document1 - Microsoft ...', 'mezunuyet - Microsoft Word', and 'Project1', along with the system clock showing 10:26.

NEAR EAST UNIVERSITY

MENU

REPORTS MENU

- Department Report
- Course Report
- Teacher Report
- Student Report
- Department-Teachers**
- Department Of Teachers Report
- Department-Courses
- Department Of Courses Report
- Course-Teacher
- Student-Grades
- Department Of Students Report

MAIN MENU

Listing Departments' Teachers

Choose A Department: MAN

Tname	Tsurname
AHMET	ERTUGAN
AHMET	USTUN
ALI	MALEK
CUMALI	SEN
MEHMET	AGA
SERIFE	FUPOGLU
TULEN	SANER

Print This Page Close

Document1 - Microsoft ... mezunuyet - Microsoft Word Project1 MP3 (D:) 10:26

Department of Teachers Report:

The screenshot displays a web application interface for Near East University. At the top, there is a header with the university's name and logo. Below the header, a 'MENU' section is visible on the left, containing a 'REPORTS MENU' with various report options. The main content area shows a report titled 'Listing Teachers According to Departments'. This report is presented as a table with two columns: 'Name' and 'departmentId'. The table lists 15 teachers and their respective departments. At the bottom of the report, there are two buttons: 'Print This Page' and 'Close'. The application is running in a web browser, and the Windows taskbar at the bottom shows several open applications, including 'Başlat', 'Document1 - Microsoft ...', 'mezuniyet - Microsoft Word', and 'Project1'. The system clock indicates the time is 10:27 on MP3 (D:).

NEAR EAST UNIVERSITY

MENU

REPORTS MENU

- Department Report
- Course Report
- Teacher Report
- Student Report
- Department-Teachers
- Department Of Teachers Report
- Department-Courses
- Department Of Courses Report
- Course-Teacher
- Student-Grades
- Department Of Students Report

MAIN MENU

7/ Rep_teach_dept


Listing Teachers According to Departments

Name	departmentId
MEHMET	ACC
AHMET	ARC
SUREYYA	CHEM
ILHAM	CIS
ADIL	CIS
UMIT	CIS
NADIRE	CIS
YALCIN	CIS
MUSTAFA	CIS
TANSEL	CIS
HUSEYIN	COM
ERDAL	ECON
FATMA	ECON
MEHMET	MAN


Print This Page **Close**

Başlat **Document1 - Microsoft ...** **mezuniyet - Microsoft Word** **Project1** **MP3 (D:)** **10:27**

Department – Courses:



NEAR EAST UNIVERSITY



MENU

REPORTS MENU

Department Report

Course Report

Teacher Report

Student Report

Department-Teachers

Department Of Teachers Report

Department-Courses

Department Of Courses Report

Course-Teacher

Student-Grades

Department Of Students Report

MAIN MENU

Dept. Course

Departments with Courses

DepartmentId	DepartmentName	CO
▶ ABC	ARCHITECTURE	
MAN	BUSINESS ADMINISTRATION	MA
MAN	BUSINESS ADMINISTRATION	AT
MAN	BUSINESS ADMINISTRATION	MA
MAN	BUSINESS ADMINISTRATION	MA
MAN	BUSINESS ADMINISTRATION	EN
MAN	BUSINESS ADMINISTRATION	EN
MAN	BUSINESS ADMINISTRATION	CI
MAN	BUSINESS ADMINISTRATION	MA
MAN	BUSINESS ADMINISTRATION	MA
CIS	COMPUTER INFORMATION SYSTEMS	MA
CIS	COMPUTER INFORMATION SYSTEMS	CI
CIS	COMPUTER INFORMATION SYSTEMS	CI
CIS	COMPUTER INFORMATION SYSTEMS	CI
CIS	COMPUTER INFORMATION SYSTEMS	CI
CIS	COMPUTER INFORMATION SYSTEMS	CI
CIS	COMPUTER INFORMATION SYSTEMS	MA
CIS	COMPUTER INFORMATION SYSTEMS	CI

Print This Page

Close

Baslat

Document1 - Microsoft ...

mezunyet - Microsoft Word

Project1

MP3 (D:)

10:28

Department of Courses Report:

The screenshot displays a web application interface for Near East University. At the top, a banner features the university's name. Below this, a 'MENU' section on the left lists various reports under 'REPORTS MENU' and a 'MAIN MENU'. The main content area, titled 'Courses According to Department', shows a 'Listing Department's Courses' window. This window includes a dropdown menu to 'Choose A Department' (currently set to 'MAN') and a table of courses. The table has columns for 'CourseId', 'CourseCode', 'DepartmentId', and 'CourseName'. Below the table are 'Print This Page' and 'Close' buttons. The bottom of the screen shows a Windows taskbar with open applications like 'Document1 - Microsoft Word' and 'Project1', along with the system clock showing 10:29.

NEAR EAST UNIVERSITY

MENU

REPORTS MENU

- Department Report
- Course Report
- Teacher Report
- Student Report
- Department-Teachers
- Department Of Teachers Report
- Department-Courses
- Department Of Courses Report
- Course-Teacher
- Student-Grades
- Department Of Students Report

MAIN MENU

Courses According to Department

Listing Department's Courses

Choose A Department: **MAN**

CourseId	CourseCode	DepartmentId	CourseName
7	ENG101	MAN	ENGLISH
8	ENG102	MAN	ENGLISH 2
9	CIS131	MAN	INTRODUCTION
3	MAN303	MAN	MARKETING
4	MAN306	MAN	MARKETING 2
19	MAN403	MAN	MARKETING 3
13	MAN308	MAN	OPERATIONS RESE
2	MAN401	MAN	SMALL BUSINESS
20	ATA101	MAN	TARİH

Print This Page **Close**

Taskbar: Baylat, Document1 - Microsoft Word, mezuniyet - Microsoft Word, Project1, MP3 (D:), 10:29

Course – Teacher:

NEAR EAST UNIVERSITY

MENU

REPORTS MENU

- Department Report
- Course Report
- Teacher Report
- Student Report
- Department-Teachers
- Department Of Teachers Report
- Department-Courses
- Department Of Courses Report
- Course-Teacher
- Student-Grades
- Department Of Students Report

MAIN MENU

Course_Teacher

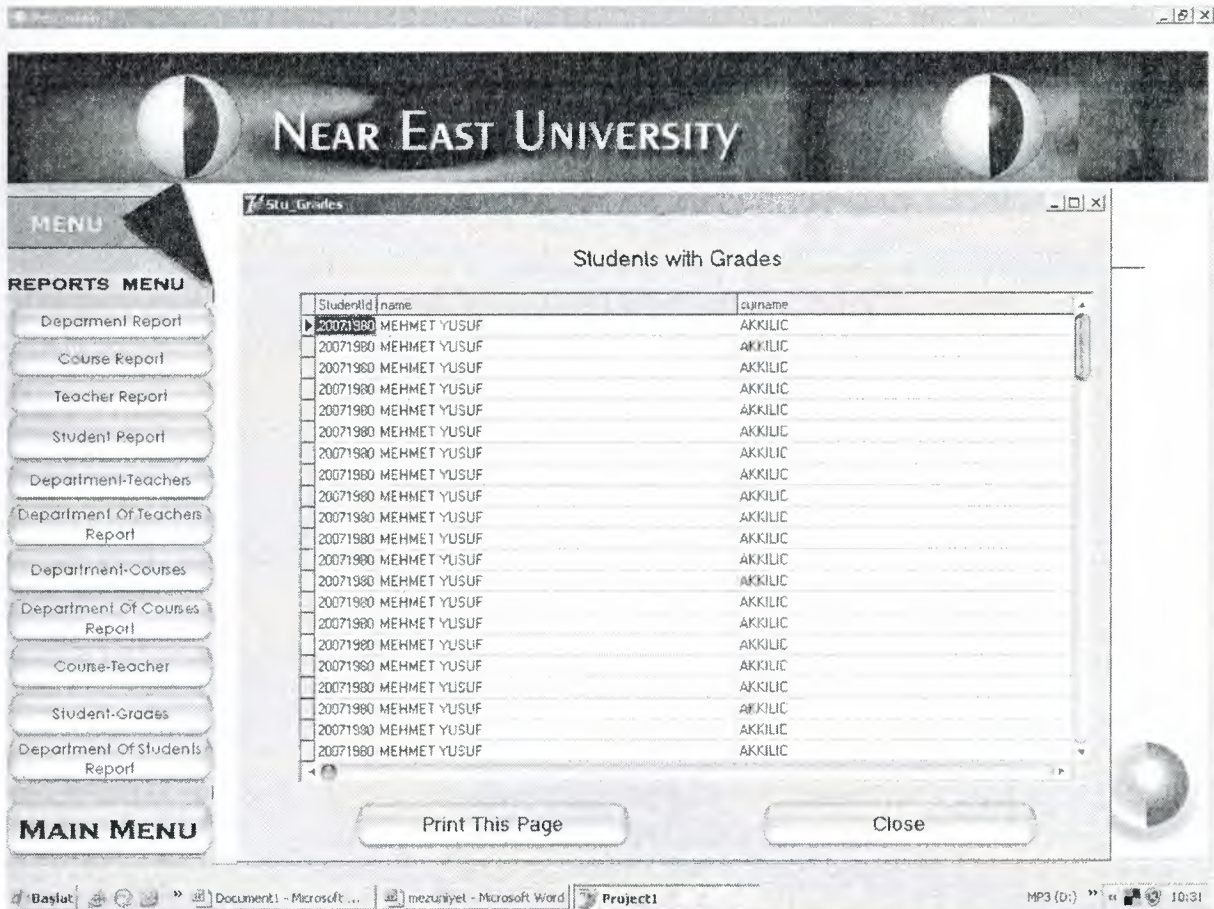
Courses with Teachers

coursecode	coursename	Tname
MAN303	MARKETING	AHMET
MAN306	MARKETING 2	AHMET
	ASD	AHMET
MAN403	MARKETING 3	AHMET
MAN308	OPERATIONS RESEARCH	ALI
CIS456	JAVA	ALI
ATA101	TARİH	CLIMALI
CIS331	SYSTEM ANALYS	ILHAM
CIS305	PROGRAMMING TECHNIQS	ILHAM
CIS244	DATA STRUCTURES	MUSTAFA
CIS151	INT	MUSTAFA
CIS243	DATA STRUCTURE	MUSTAFA
COM318	DATA COMMUNICATION	NADIRE
CIS240	DELPHI	NADIRE
CIS347	HKDAS	TANSEL
CIS301	PROGRAMMING WITH C	TANSEL
MAN401	SMALL BUSINESS	TÜLEN
MAT101	MATEMATICS	UMIT
MAN404	INTRODUCTION	YALCIN
CIS132	INTRO COMPUTER	YALCIN

Print This Page **Close**

Baslat Document1 - Microsoft Word Project1 MP3 (D:) 10:30

Student – Grades:



Department of Student Report:

The screenshot displays a web application interface for Near East University. The main header features the university's name and logo. A left sidebar contains a 'MENU' section with a 'REPORTS MENU' and a 'MAIN MENU'. The 'REPORTS MENU' includes options like 'Department Report', 'Course Report', 'Teacher Report', 'Student Report', 'Department-Teachers', 'Department Of Teachers Report', 'Department-Courses', 'Department Of Courses Report', 'Course-Teacher', 'Student-Grades', and 'Department Of Students Report'. The 'MAIN MENU' is also visible. The main content area shows a window titled 'Students According To Departments' with the subtitle 'Listing Students According to Departments'. It features a 'Choose A Department' dropdown menu set to 'CIS'. Below this is a table listing students:

StudentId	DepartmentId	name
20071981	CIS	CEMIL
20071988	CIS	MARIA
20071980	CIS	MEHMET YUSUF
20071986	CIS	SALIHA

At the bottom of the window are 'Print This Page' and 'Close' buttons. The taskbar at the bottom shows the 'Bağlat' button, several icons, and open applications including 'Document1 - Microsoft ...', 'mezuniyet - Microsoft Word', and 'Project1'. The system clock indicates 'MP3 (D:) 10:32'.

Login Menu:

CHANGE LOGIN INFORMATION

NEAR EAST UNIVERSITY

MENU

LOGIN MENU

MAIN MENU

LOGIN MENU

User Name

Password

Department

Access Level

Admin

Advisor

User

Change Password

Delete User

Change Access Level

Add User

OK CANCEL

Document1 - Microsoft ...

mezuniyet - Microsoft Word

Project1

NP3 [D:] 10:33

APPENDIX II

SOURCE CODE

program Project1;

uses

Forms,

Unit1 in 'Unit1.pas' {frm_main},

Unit2 in 'Unit2.pas' {frm_std},

Unit3 in 'Unit3.pas' {frm_addstd},

Unit4 in 'Unit4.pas' {frm_updstd},

Unit5 in 'Unit5.pas' {frm_teacher},

Unit6 in 'Unit6.pas' {frm_addteacher},

Unit8 in 'Unit8.pas' {frm_course},

Unit9 in 'Unit9.pas' {frm_addcourse},

Unit10 in 'Unit10.pas' {frm_updcourse},

Unit11 in 'Unit11.pas' {frm_dept},

Unit12 in 'Unit12.pas' {frm_adddept},

Unit13 in 'Unit13.pas' {frm_updept},

Unit14 in 'Unit14.pas' {frm_trans},

Unit22 in 'Unit22.pas' {frm_correg},

Unit15 in 'Unit15.pas' {frm_semend},

Unit16 in 'Unit16.pas' {Rep_main},
Unit17 in 'Unit17.pas' {Rep_all_depts},
Unit18 in 'Unit18.pas' {Rep_all_teacs},
Unit19 in 'Unit19.pas' {Rep_teacs_dept},
Unit20 in 'Unit20.pas' {Rep_dept_teacs},
Unit21 in 'Unit21.pas' {Rep_all_courses},
Unit23 in 'Unit23.pas' {Rep_dept_cor},
Unit24 in 'Unit24.pas' {Rep_all_stu},
Unit25 in 'Unit25.pas' {Rep_dept_stu},
Unit40 in 'Unit40.pas' {frm_about},
Unit7 in 'Unit7.pas' {frm_updteacher},
Unit26 in 'Unit26.pas' {frm_password},
Unit27 in 'Unit27.pas' {frm_login},
Unit28 in 'Unit28.pas' {frm_beginsemester},
Unit29 in 'Unit29.pas' {frm_adddrop},
Unit30 in 'Unit30.pas' {Form30},
Unit31 in 'Unit31.pas' {Form31},
Unit32 in 'Unit32.pas' {Form32},
Unit33 in 'Unit33.pas' {Form33};

{SR *.res}

begin

Application.Initialize;

Application.CreateForm(Tfrm_password, frm_password);

Application.CreateForm(Tfrm_main, frm_main);

Application.CreateForm(Tfrm_std, frm_std);

Application.CreateForm(Tfrm_addstd, frm_addstd);

Application.CreateForm(Tfrm_updstd, frm_updstd);

Application.CreateForm(Tfrm_teacher, frm_teacher);

Application.CreateForm(Tfrm_addteacher, frm_addteacher);

Application.CreateForm(Tfrm_course, frm_course);

Application.CreateForm(Tfrm_addcourse, frm_addcourse);

Application.CreateForm(Tfrm_updcourse, frm_updcourse);

Application.CreateForm(Tfrm_dept, frm_dept);

Application.CreateForm(Tfrm_adddept, frm_adddept);

Application.CreateForm(Tfrm_updept, frm_updept);

Application.CreateForm(Tfrm_trans, frm_trans);

Application.CreateForm(Tfrm_correg, frm_correg);

Application.CreateForm(Tfrm_semend, frm_semend);

Application.CreateForm(TRep_main, Rep_main);

Application.CreateForm(TRep_all_depts, Rep_all_depts);

Application.CreateForm(TRep_all_teacs, Rep_all_teacs);

Application.CreateForm(TRep_teacs_dept, Rep_teacs_dept);

```

Application.CreateForm(TRep_dept_teacs, Rep_dept_teacs);
Application.CreateForm(TRep_all_courses, Rep_all_courses);
Application.CreateForm(TRep_dept_cor, Rep_dept_cor);
Application.CreateForm(TRep_all_stu, Rep_all_stu);
Application.CreateForm(TRep_dept_stu, Rep_dept_stu);
Application.CreateForm(Tfrm_about, frm_about);
Application.CreateForm(Tfrm_updteacher, frm_updteacher);
Application.CreateForm(Tfrm_login, frm_login);
Application.CreateForm(Tfrm_beginsemester, frm_beginsemester);
Application.CreateForm(Tfrm_adddrop, frm_adddrop);
Application.CreateForm(TForm30, Form30);
Application.CreateForm(TForm31, Form31);
Application.CreateForm(TForm32, Form32);
Application.CreateForm(TForm33, Form33);
Application.Run;

end.

```

```

unit Unit1;

```

```

interface

```

```

uses

```

```

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

```

```

Dialogs, DB, DBTables, ADOdb, StdCtrls, ExtCtrls, jpeg, ComCtrls;

```

type

```
Tfrm_main = class(TForm)

    Button1: TButton;

    Button2: TButton;

    Button4: TButton;

    Button5: TButton;

    Button7: TButton;

    DataSource1: TDataSource;

    ADOTable1: TADOTable;

    ADOTable2: TADOTable;

    ADOTable3: TADOTable;

    ADOTable5: TADOTable;

    ADOTable6: TADOTable;

    DataSource2: TDataSource;

    DataSource3: TDataSource;

    DataSource5: TDataSource;

    DataSource6: TDataSource;

    ADOQuery1: TADOQuery;

    DataSource7: TDataSource;

    ADOQuery2: TADOQuery;

    DataSource8: TDataSource;

    ADOQuery3: TADOQuery;

    DataSource9: TDataSource;

    ADOQuery4: TADOQuery;

    DataSource10: TDataSource;
```



```
ADOQuery5: TADOQuery;

DataSource11: TDataSource;

ADOQuery6: TADOQuery;

DataSource12: TDataSource;

Button8: TButton;

Button6: TButton;

Button11: TButton;

ADOTable4: TADOTable;

DataSource4: TDataSource;

DataSource13: TDataSource;

Image1: TImage;

Label2: TLabel;

Image2: TImage;

Button10: TButton;

procedure Button7Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button10Click(Sender: TObject);

procedure Button11Click(Sender: TObject);

private
```

```

    { Private declarations }

public

    { Public declarations }

end;

var

    frm_main: Tfrm_main;

implementation

uses Unit2, Unit5, Unit8, Unit11, Unit22, Unit14, Unit15, Unit16, Unit40,

    Unit27, unit26, Unit28;

{$R *.dfm}

procedure Tfrm_main.Button7Click(Sender: TObject);

begin

    //application.Terminate;

    frm_main.Close;

    frm_password.Show;

end;

procedure Tfrm_main.Button1Click(Sender: TObject);

begin

    if level='Admin' then

        begin

            frm_std.show;

            frm_main.hide;

        end

    else

        messagedlg('You are not an accessible person to this part!!',mtinformation,[mbok],0);

```

```

end;

procedure Tfrm_main.Button2Click(Sender: TObject);
begin
    if level='Admin' then

        begin
            frm_teacher.show;

            frm_main.Hide;

        end

    else

        messagedlg('You are not an accessible person to this part!!',mtinformation,[mbok],0);

    end;

procedure Tfrm_main.Button4Click(Sender: TObject);
begin
    if level='Admin' then

        begin
            frm_course.show;

            frm_main.hide;

        end

    else

        messagedlg('You are not an accessible person to this part!!',mtinformation,[mbok],0);

    end;

procedure Tfrm_main.Button5Click(Sender: TObject);
begin
    if level='Admin' then

        begin

```

```

frm_dept.show;

frm_main.Hide;

end

else

    messagedlg('You are not an accessible person to this part!!',mtinformation,[mbok],0);

end;

procedure Tfrm_main.Button8Click(Sender: TObject);

begin

    if (level='Advisor') or (level='Admin') then

        begin

            frm_main.Hide;

            frm_beginsemester.show;

        end

    else

        messagedlg('You are not an accessible person to this part!!',mtinformation,[mbok],0);

    end;

procedure Tfrm_main.Button3Click(Sender: TObject);

begin

    if (level='Admin') or (level='User') then

        begin

            frm_trans.show;

            frm_main.Hide;

        end

    else

        messagedlg('You are not an accessible person to this part!!',mtinformation,[mbok],0);

```


end;

procedure Tfrm_main.Button6Click(Sender: TObject);

begin

if (level='Admin') or (level='User') then

begin

rep_main.show;

frm_main.Hide;

end

else

messaging('You are not an accessible person to this part!!',mtinformation,[mbok],0);

end;

procedure Tfrm_main.Button10Click(Sender: TObject);

begin

if (level='Admin') or (level='User') then

begin

frm_trans.show;

frm_main.hide;

end

else

messaging('You are not an accessible person to this part!!',mtinformation,[mbok],0);

end;

procedure Tfrm_main.Button11Click(Sender: TObject);

begin

 frm_login.show;

 frm_main.Hide;

end;

end.

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Buttons, jpeg, ExtCtrls, StdCtrls;

type

Tfrm_std = class(TForm)

 Image1: TImage;

 Button1: TButton;

 Button2: TButton;

 Button3: TButton;

 Label1: TLabel;

 Image2: TImage;

```

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

private

    { Private declarations }

public

    { Public declarations }

end;

var

    frm_std: Tfrm_std;

implementation

uses Unit1, Unit3, Unit4, unit26;

{$R *.dfm}

procedure Tfrm_std.FormClose(Sender: TObject; var Action: TCloseAction);

begin

    frm_main.show;

end;

procedure Tfrm_std.Button1Click(Sender: TObject);

begin

    frm_addstd.show;

    frm_std.Hide;

end;

procedure Tfrm_std.Button2Click(Sender: TObject);

```

```
begin
    frm_updstd.Show;
    frm_std.Hide;
end;
```

```
procedure Tfrm_std.Button3Click(Sender: TObject);
```

```
begin
    frm_std.close;
    frm_main.Show;
end;
```

```
procedure Tfrm_std.Button4Click(Sender: TObject);
```

```
begin
    frm_main.Close;
    frm_password.Show;
end;
end.
```

unit Unit3;

```
interface
```

```
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, Mask, DBCtrls, ExtCtrls, jpeg, strutils, dateutils,
    ComCtrls;
```

```
type
    Tfrm_addstd = class(TForm)
        Button1: TButton;
        Button2: TButton;
        Edit1: TEdit;
        Edit2: TEdit;
        Edit3: TEdit;
```

```

Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Edit9: TEdit;
Edit10: TEdit;
Edit11: TEdit;
Edit12: TEdit;
Edit13: TEdit;
Memo1: TMemo;
ComboBox1: TComboBox;
Button3: TButton;
Button4: TButton;
Button5: TButton;
ComboBox2: TComboBox;
DateTimePicker1: TDateTimePicker;
Image1: TImage;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label34: TLabel;
Label35: TLabel;
Label36: TLabel;
Label37: TLabel;
Label38: TLabel;
Label39: TLabel;
Label40: TLabel;
Label41: TLabel;
Label23: TLabel;
Label2: TLabel;
Label1: TLabel;
Label3: TLabel;
Image2: TImage;
Label10: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Button6: TButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Button5Click(Sender: TObject);

```



```

procedure Button4Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Edit2KeyPress(Sender: TObject; var Key: Char);
procedure Memo1KeyPress(Sender: TObject; var Key: Char);
procedure ComboBox2KeyPress(Sender: TObject; var Key: Char);

private
  { Private declarations }

public
  { Public declarations }

end;

var
  frm_addstd: Tfrm_addstd;

implementation
uses unit1, Unit2, unit26;

{$R *.dfm}

procedure Tfrm_addstd.Button1Click(Sender: TObject);
begin
  frm_std.show;
  frm_addstd.Close;
end;

procedure Tfrm_addstd.Button2Click(Sender: TObject);
begin
  frm_main.show;
  frm_addstd.hide;

end;

procedure clearing(Sender:Tobject);
begin
  frm_addstd.Edit1.Clear;
  frm_addstd.Edit2.Clear;
  frm_addstd.Edit3.Clear;
  frm_addstd.Edit4.Clear;
  frm_addstd.Edit5.Clear;
  frm_addstd.Edit6.Clear;
  frm_addstd.Edit7.Clear;
  frm_addstd.Edit8.Clear;
  frm_addstd.Edit9.Clear;
  frm_addstd.Edit10.Clear;
  frm_addstd.Edit11.Clear;
  frm_addstd.Edit12.Clear;

```

```
frm_addstd.Edit13.Clear;
frm_addstd.combobox2.Clear;
frm_addstd.combobox2.Items.Clear;
frm_addstd.Memo1.Clear;
frm_addstd.combobox1.Clear;
frm_addstd.combobox1.Items.Clear;
end;
```

```
procedure invisibility(Sender:TObject);
begin
    frm_addstd.Edit2.Visible:=false;
    frm_addstd.Edit3.Visible:=false;
    frm_addstd.Edit4.Visible:=false;
    frm_addstd.Edit5.Visible:=false;
    frm_addstd.Edit6.Visible:=false;
    frm_addstd.Edit7.Visible:=false;
    frm_addstd.Edit8.Visible:=false;
    frm_addstd.Edit9.Visible:=false;
    frm_addstd.Edit10.Visible:=false;
    frm_addstd.Edit11.Visible:=false;
    frm_addstd.Edit12.Visible:=false;
    frm_addstd.Edit13.Visible:=false;
    frm_addstd.DateTimePicker1.Visible:=false;
    frm_addstd.combobox2.Visible:=false;
    frm_addstd.memo1.Visible:=false;
    frm_addstd.combobox1.Visible:=false;

end;
```

```
procedure visibility(Sender:TObject);
begin
    frm_addstd.Edit2.Visible:=true;
    frm_addstd.Edit3.Visible:=true;
    frm_addstd.Edit4.Visible:=true;
    frm_addstd.Edit5.Visible:=true;
    frm_addstd.Edit6.Visible:=true;
    frm_addstd.Edit7.Visible:=true;
    frm_addstd.Edit8.Visible:=true;
    frm_addstd.Edit9.Visible:=true;
    frm_addstd.Edit10.Visible:=true;
    frm_addstd.Edit11.Visible:=true;
    frm_addstd.Edit12.Visible:=true;
    frm_addstd.Edit13.Visible:=true;
    frm_addstd.DateTimePicker1.Visible:=true;
    frm_addstd.memo1.Visible:=true;
    frm_addstd.combobox1.Visible:=true;
    frm_addstd.combobox2.Visible:=true;

end;
```

```

procedure enabling(Sender:Tobject);
begin
    frm_addstd.edit2.enabled:=true;
    frm_addstd.edit3.enabled:=true;
    frm_addstd.edit4.enabled:=true;
    frm_addstd.edit5.enabled:=true;
    frm_addstd.edit6.enabled:=true;
    frm_addstd.edit7.enabled:=true;
    frm_addstd.edit8.enabled:=true;
    frm_addstd.edit9.enabled:=true;
    frm_addstd.edit10.enabled:=true;
    frm_addstd.edit11.enabled:=true;
    frm_addstd.edit12.enabled:=true;
    frm_addstd.edit13.enabled:=true;
    frm_addstd.DateTimePicker1.enabled:=true;
    frm_addstd.combobox1.enabled:=true;
    frm_addstd.combobox2.enabled:=true;
    frm_addstd.memo1.enabled:=true;
    frm_addstd.button3.enabled:=true;
    frm_addstd.button4.enabled:=true;

```

```

end;

```

```

procedure disabling(Sender:Tobject);
begin
    frm_addstd.edit2.enabled:=false;
    frm_addstd.edit3.enabled:=false;
    frm_addstd.edit4.enabled:=false;
    frm_addstd.edit5.enabled:=false;
    frm_addstd.edit6.enabled:=false;
    frm_addstd.edit7.enabled:=false;
    frm_addstd.edit8.enabled:=false;
    frm_addstd.edit9.enabled:=false;
    frm_addstd.edit10.enabled:=false;
    frm_addstd.edit11.enabled:=false;
    frm_addstd.edit12.enabled:=false;
    frm_addstd.edit13.enabled:=false;
    frm_addstd.DateTimePicker1.enabled:=false;
    frm_addstd.combobox1.enabled:=false;
    frm_addstd.combobox2.enabled:=false;
    frm_addstd.memo1.enabled:=false;
    frm_addstd.button3.enabled:=false;
    frm_addstd.button4.enabled:=false;
end;

```

```

procedure Tfrm_addstd.FormShow(Sender: TObject);
begin
    edit1.ReadOnly:=false;

```

```

clearing(sender);
edit1.SetFocus;
disabling(sender);
invisibility(sender);
// frm_addstd.Position:=posscreencenter;
end;

```

```

procedure Tfrm_addstd.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    edit1.ReadOnly:=false;

    // disabling(sender);
    frm_std.Show;
end;

```

```

procedure Tfrm_addstd.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    if not((key in ['0'..'9'])or(key=#8) or (key=#13)) then
    begin
        key:=#0;
        messagedlg('Please Enter a Numeric Character ',mtwarning,[mbok],0);
    end;
    if key=#13 then
        button5.Click;
end;

```

```

procedure Tfrm_addstd.Button5Click(Sender: TObject);
begin
    if edit1.text<>"" then
    begin
        ShortDateFormat := 'yyyy';

        if Ansileftstr(edit1.Text,4)=datetostr(date) then
        begin
            if length(edit1.Text)=8 then
            begin
                with frm_main.ADOQuery1 do
                begin
                    sql.clear;
                    sql.Add('Select StudentId from student where StudentId="'+edit1.text+'"');
                    open;
                end;
                if frm_main.DataSource7.DataSet.RecordCount=0 then
                begin
                    shortdateformat:='m/d/yyyy';
                    enabling(sender);
                    visibility(sender);
                    edit2.SetFocus;
                    combobox2.Items.Add('Male');
                    combobox2.Items.Add('Female');

```

```

edit1.ReadOnly:=true;
with frm_main.ADOQuery1 do
begin
  sql.clear;
  sql.Add('Select DepartmentId from Department');
  open;
end;
frm_main.datasource7.DataSet.First;
while not frm_main.datasource7.DataSet.Eof do
begin
  combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
  frm_main.DataSource7.DataSet.Next;
end;
datetimepicker1.Date:=strtodate('01/01/1985');
end
else
begin
  messagedlg('This Student Number Is Already Recorded !',mtwarning,[mbok],0);
  edit1.SetFocus;
end;
end
else
begin
  messagedlg('Student Number Must Be 8 Digits !',mtwarning,[mbok],0);
  edit1.SetFocus;
end;
end
else
begin
  messagedlg('First 4 digit must be same with current year !',mtwarning,[mbok],0);
  edit1.Clear;
  edit1.SetFocus;
end;
end
else
begin
  messagedlg('Please Enter Number For New Student !',mtwarning,[mbok],0);
  edit1.Clear;
  edit1.SetFocus;
end;
end;
end;

```

```

procedure Tfrm_addstd.Button4Click(Sender: TObject);
begin
  clearing(sender);
  disabling(sender);
  invisibility(sender);
  edit1.SetFocus;
  edit1.Clear;
  edit1.ReadOnly:=false;

```


end;

procedure Tfrm_addstd.Button3Click(Sender: TObject);

begin

if ((edit2.Text<>")and(edit3.Text<>")and(edit8.Text<>")and(combobox1.Text<>")) then

begin

with frm_main.DataSource1 do

begin

edit1.ReadOnly:=false;

dataset.Edit;

dataset.Append;

dataset.FieldValues['StudentId']:=edit1.Text;

if combobox1.text<>" then

dataset.FieldValues['DepartmentId']:=combobox1.Text;

dataset.FieldValues['name']:=edit2.Text;

dataset.FieldValues['surname']:=edit3.Text;

if combobox2.text<>" then

dataset.FieldValues['sex']:=combobox2.Text;

if memo1.Text<>" then

dataset.FieldValues['Address']:=memo1.Text;

if edit7.text<>" then

dataset.FieldValues['City']:=edit7.Text;

if edit8.text<>" then

dataset.FieldValues['Country']:=edit8.Text;

if edit9.text<>" then

dataset.FieldValues['postcode']:=edit9.Text;

if edit11.text<>" then

dataset.FieldValues['Gsm']:=edit11.Text;

if edit6.text<>" then

dataset.FieldValues['FatherName']:=edit6.Text;

if edit5.text<>" then

dataset.FieldValues['MotherName']:=edit5.Text;

if edit4.text<>" then

dataset.FieldValues['BirthPlace']:=edit4.Text;

dataset.FieldValues['BirthDate']:=datetimepicker1.Date;

if edit13.text<>" then

dataset.FieldValues['OsymNo']:=edit13.Text;

if edit12.text<>" then

dataset.FieldValues['Email']:=edit12.Text;

dataset.FieldValues['RegisterDate']:=Date;

if edit10.text<>" then

dataset.FieldValues['Phone']:=edit10.Text;

dataset.FieldValues['ProcessUser']:=username;

dataset.Post;

messagedlg('The student recorded successfully !',mtwarning,[mbok],0);

```

clearing(sender);
disabling(sender);
invisibility(sender);
edit1.SetFocus;
end;

end
else
begin
  messagedlg('Please Fill All The Required Fields !',mtwarning,[mbok],0);
  edit2.SetFocus;

  end;
end;

procedure Tfrm_addstd.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
  if not((upcase(key) in ['A'..'Z'])or(key=#8) or (key=#13) or (key=chr(VK_SPACE)))then
  begin
    key:=#0;
    messagedlg('Please Enter a Character ',mtwarning,[mbok],0);
  end
  else
    key:=upcase(key);

end;

procedure Tfrm_addstd.Memo1KeyPress(Sender: TObject; var Key: Char);
begin
  key:=upcase(key);
end;

procedure Tfrm_addstd.ComboBox2KeyPress(Sender: TObject; var Key: Char);
begin
  key:=#0;
end;

end.

```

unit Unit4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, DBCtrls, ExtCtrls, Mask, jpeg, ComCtrls;

type

Tfrm_updstd = class(TForm)
 Image1: TImage;

Button1: TButton;
Button2: TButton;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Edit9: TEdit;
Edit10: TEdit;
Edit11: TEdit;
Edit12: TEdit;
Edit13: TEdit;
Memo1: TMemo;
ComboBox1: TComboBox;
Button3: TButton;
Button4: TButton;
Button5: TButton;
ComboBox2: TComboBox;
DateTimePicker1: TDateTimePicker;
Label1: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label34: TLabel;
Label35: TLabel;
Label36: TLabel;
Label37: TLabel;
Label38: TLabel;
Label39: TLabel;
Label40: TLabel;
Label41: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Image2: TImage;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label23: TLabel;
Label24: TLabel;
Button7: TButton;

```

procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormShow(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Edit2KeyPress(Sender: TObject; var Key: Char);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Memo1KeyPress(Sender: TObject; var Key: Char);
procedure Button6Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  frm_updstd: Tfrm_updstd;

```

implementation

```

uses Unit1, Unit2, unit26, unit3;

```

```

{$R *.dfm}
procedure clearing(Sender: Tobject);
begin
  frm_updstd.Edit1.Clear;
  frm_updstd.Edit2.Clear;
  frm_updstd.Edit3.Clear;
  frm_updstd.Edit4.Clear;
  frm_updstd.Edit5.Clear;
  frm_updstd.Edit6.Clear;
  frm_updstd.Edit7.Clear;
  frm_updstd.Edit8.Clear;
  frm_updstd.Edit9.Clear;
  frm_updstd.Edit10.Clear;
  frm_updstd.Edit11.Clear;
  frm_updstd.Edit12.Clear;
  frm_updstd.Edit13.Clear;
  frm_updstd.combobox2.Clear;
  frm_updstd.combobox2.Items.Clear;
  frm_updstd.Memo1.Clear;
  frm_updstd.combobox1.Clear;
  frm_updstd.combobox1.Items.Clear;
  frm_updstd.Label23.Caption:="";
  frm_updstd.Label24.Caption:="";
end;

```

```

procedure invisibility(Sender:TObject);
begin
    frm_updstd.Edit2.Visible:=false;
    frm_updstd.Edit3.Visible:=false;
    frm_updstd.Edit4.Visible:=false;
    frm_updstd.Edit5.Visible:=false;
    frm_updstd.Edit6.Visible:=false;
    frm_updstd.Edit7.Visible:=false;
    frm_updstd.Edit8.Visible:=false;
    frm_updstd.Edit9.Visible:=false;
    frm_updstd.Edit10.Visible:=false;
    frm_updstd.Edit11.Visible:=false;
    frm_updstd.Edit12.Visible:=false;
    frm_updstd.Edit13.Visible:=false;
    frm_updstd.DateTimePicker1.Visible:=false;
    frm_updstd.combobox2.Visible:=false;
    frm_updstd.memo1.Visible:=false;
    frm_updstd.combobox1.Visible:=false;

end;

```

```

procedure visibility(Sender:TObject);
begin
    frm_updstd.Edit2.Visible:=true;
    frm_updstd.Edit3.Visible:=true;
    frm_updstd.Edit4.Visible:=true;
    frm_updstd.Edit5.Visible:=true;
    frm_updstd.Edit6.Visible:=true;
    frm_updstd.Edit7.Visible:=true;
    frm_updstd.Edit8.Visible:=true;
    frm_updstd.Edit9.Visible:=true;
    frm_updstd.Edit10.Visible:=true;
    frm_updstd.Edit11.Visible:=true;
    frm_updstd.Edit12.Visible:=true;
    frm_updstd.Edit13.Visible:=true;
    frm_updstd.DateTimePicker1.Visible:=true;
    frm_updstd.memo1.Visible:=true;
    frm_updstd.combobox1.Visible:=true;
    frm_updstd.combobox2.Visible:=true;

end;

```

```

procedure enabling(Sender:TObject);
begin
    frm_updstd.edit2.enabled:=true;
    frm_updstd.edit3.enabled:=true;
    frm_updstd.edit4.enabled:=true;
    frm_updstd.edit5.enabled:=true;
    frm_updstd.edit6.enabled:=true;
    frm_updstd.edit7.enabled:=true;

end;

```



```
frm_updstd.edit8.enabled:=true;
frm_updstd.edit9.enabled:=true;
frm_updstd.edit10.enabled:=true;
frm_updstd.edit11.enabled:=true;
frm_updstd.edit12.enabled:=true;
frm_updstd.edit13.enabled:=true;
frm_updstd.DateTimePicker1.enabled:=true;
frm_updstd.combobox1.enabled:=true;
frm_updstd.combobox2.enabled:=true;
frm_updstd.memo1.enabled:=true;
frm_updstd.button3.enabled:=true;
frm_updstd.button4.enabled:=true;
```

end;

```
procedure disabling(Sender:TObject);
begin
```

```
    frm_updstd.edit2.enabled:=false;
    frm_updstd.edit3.enabled:=false;
    frm_updstd.edit4.enabled:=false;
    frm_updstd.edit5.enabled:=false;
    frm_updstd.edit6.enabled:=false;
    frm_updstd.edit7.enabled:=false;
    frm_updstd.edit8.enabled:=false;
    frm_updstd.edit9.enabled:=false;
    frm_updstd.edit10.enabled:=false;
    frm_updstd.edit11.enabled:=false;
    frm_updstd.edit12.enabled:=false;
    frm_updstd.edit13.enabled:=false;
    frm_updstd.edit13.enabled:=false;
    frm_updstd.DateTimePicker1.enabled:=false;
    frm_updstd.combobox1.enabled:=false;
    frm_updstd.combobox2.enabled:=false;
    frm_updstd.memo1.enabled:=false;
    frm_updstd.button3.enabled:=false;
    frm_updstd.button4.enabled:=false;
```

end;

```
procedure Tfrm_updstd.Button1Click(Sender: TObject);
```

```
begin
    frm_std.show;
    frm_updstd.hide;
end;
```

```
procedure Tfrm_updstd.Button2Click(Sender: TObject);
```

```
begin
```

```
frm_main.Show;  
frm_upstd.hide;  
end;
```

```
procedure Tfrm_upstd.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
    edit1.ReadOnly:=false;
```

```
// disabling(sender);  
    frm_std.Show;  
end;
```

```
procedure Tfrm_upstd.FormShow(Sender: TObject);  
begin  
    edit1.ReadOnly:=false;  
    clearing(sender);  
    edit1.SetFocus;  
    disabling(sender);  
    invisibility(sender);  
end;
```

```
procedure Tfrm_upstd.Button5Click(Sender: TObject);  
begin  
    if edit1.text<>" then  
        begin  
            ShortDateFormat := 'm/dd/yyyy';  
            if length(edit1.Text)=8 then  
                begin  
                    with frm_main.ADOQuery1 do  
                        begin  
                            sql.clear;  
                            sql.Add('Select * from student where StudentId="'+edit1.text+'"');  
                            open;  
                        end;  
                    if frm_main.DataSource7.DataSet.RecordCount<>0 then  
                        begin  
                            enabling(sender);  
                            visibility(sender);  
                            edit2.SetFocus;  
                            combobox2.Items.Add('Male');  
                            combobox2.Items.Add('Female');  
                            edit1.ReadOnly:=true;  
                            with frm_main.ADOQuery1 do  
                                begin  
                                    sql.clear;  
                                    sql.Add('Select DepartmentId from Department');  
                                    open;  
                                end;
```

```
frm_main.datasource7.DataSet.First;
```

```

while not frm_main.datasource7.DataSet.EOF do
begin
    combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
    frm_main.DataSource7.DataSet.Next;
end;

```

```

with frm_main.ADOQuery1 do

```

```

begin
    sql.clear;
    sql.Add('Select * from student where StudentId="'+edit1.text+'"');
    open;
end;

```

```

if frm_main.DataSource7.DataSet.FieldValues['name'] <> null then
    edit2.Text:=frm_main.DataSource7.DataSet.FieldValues['name'];
if frm_main.DataSource7.DataSet.FieldValues['surname'] <> null then
    edit3.Text:=frm_main.DataSource7.DataSet.FieldValues['surname'];
if frm_main.DataSource7.DataSet.FieldValues['BirthPlace'] <> null then
    edit4.Text:=frm_main.DataSource7.DataSet.FieldValues['BirthPlace'];
if frm_main.DataSource7.DataSet.FieldValues['MotherName'] <> null then
    edit5.Text:=frm_main.DataSource7.DataSet.FieldValues['MotherName'];
if frm_main.DataSource7.DataSet.FieldValues['fathername'] <> null then
    edit6.Text:=frm_main.DataSource7.DataSet.FieldValues['fathername'];
if frm_main.DataSource7.DataSet.FieldValues['city'] <> null then
    edit7.Text:=frm_main.DataSource7.DataSet.FieldValues['city'];
if frm_main.DataSource7.DataSet.FieldValues['country'] <> null then
    edit8.Text:=frm_main.DataSource7.DataSet.FieldValues['country'];
if frm_main.DataSource7.DataSet.FieldValues['postcode'] <> null then
    edit9.Text:=frm_main.DataSource7.DataSet.FieldValues['postcode'];
if frm_main.DataSource7.DataSet.FieldValues['phone'] <> null then
    edit10.Text:=frm_main.DataSource7.DataSet.FieldValues['phone'];
if frm_main.DataSource7.DataSet.FieldValues['Gsm'] <> null then
    edit11.Text:=frm_main.DataSource7.DataSet.FieldValues['Gsm'];
if frm_main.DataSource7.DataSet.FieldValues['Email'] <> null then
    edit12.Text:=frm_main.DataSource7.DataSet.FieldValues['Email'];
if frm_main.DataSource7.DataSet.FieldValues['OsymNo'] <> null then
    edit13.Text:=frm_main.DataSource7.DataSet.FieldValues['OsymNo'];
if frm_main.DataSource7.DataSet.FieldValues['DepartmentId'] <> null then
    combobox1.Text:=frm_main.DataSource7.DataSet.FieldValues['DepartmentId'];
if frm_main.DataSource7.DataSet.FieldValues['sex'] <> null then
    combobox2.Text:=frm_main.DataSource7.DataSet.FieldValues['sex'];
if frm_main.DataSource7.DataSet.FieldValues['BirthDate'] <> null then
    datetimerpicker1.Date:=frm_main.DataSource7.DataSet.FieldValues['BirthDate'];
if frm_main.DataSource7.DataSet.FieldValues['Address'] <> null then
    memo1.Text:=frm_main.DataSource7.DataSet.FieldValues['Address'];

```

```

    label23.Caption:='The Student Is Registered At
'+DatetoStr(frm_main.DataSource7.DataSet.FieldValues['RegisterDate'])+' By
'+frm_main.DataSource7.DataSet.FieldValues['ProcessUser'];
    if frm_main.DataSource7.DataSet.FieldValues['UpdateUser'] <> null then

```

```

        label24.Caption:='The Student Record Was last Updated At
'+DatetoStr(frm_main.DataSource7.DataSet.FieldValues['UpdateDate'])+' By
'+frm_main.DataSource7.DataSet.FieldValues['UpdateUser'];

    end
    else
    begin
        messagedlg('Student Number Is Not Found !',mtwarning,[mbok],0);
        edit1.SetFocus;
    end;
    end
    else
    begin
        messagedlg('Student Number Must Be 8 Digits !',mtwarning,[mbok],0);
        edit1.SetFocus;
    end;
    end
    else
    begin
        messagedlg('Please Enter Number For Student To Update !',mtwarning,[mbok],0);
        edit1.Clear;
        edit1.SetFocus;
    end;
end;

```

```

procedure Tfrm_upstd.Button4Click(Sender: TObject);
begin
    clearing(sender);
    disabling(sender);
    invisibility(sender);
    edit1.SetFocus;
    edit1.Clear;
    edit1.ReadOnly:=false;
end;

```

```

procedure Tfrm_upstd.Button3Click(Sender: TObject);
begin
    if ((edit2.Text<>"")and(edit3.Text<>"")and(edit8.Text<>"")and(combobox1.Text<>"")) then
    begin
        with frm_main.ADOQuery1 do
        begin
            sql.clear;
            sql.Add('Select * from student where StudentId="'+edit1.text+'"');
            open;
        end;
        with frm_main.DataSource7 do
        begin
            edit1.ReadOnly:=false;
            dataset.Edit;
        end;
    end;
end;

```



```

if combobox1.text<>" then
    dataset.FieldValues['DepartmentId']:=combobox1.Text;
dataset.FieldValues['name']:=edit2.Text;
dataset.FieldValues['surname']:=edit3.Text;
if combobox2.text<>" then
    dataset.FieldValues['sex']:=combobox2.Text;
if memo1.Text<>" then
    dataset.FieldValues['Address']:=memo1.Text;
if edit7.text<>" then
    dataset.FieldValues['City']:=edit7.Text;
if edit8.text<>" then
    dataset.FieldValues['Country']:=edit8.Text;
if edit9.text<>" then
    dataset.FieldValues['postcode']:=edit9.Text;
if edit11.text<>" then
    dataset.FieldValues['Gsm']:=edit11.Text;
if edit6.text<>" then
    dataset.FieldValues['FatherName']:=edit6.Text;
if edit5.text<>" then
    dataset.FieldValues['MotherName']:=edit5.Text;
if edit4.text<>" then
    dataset.FieldValues['BirthPlace']:=edit4.Text;

dataset.FieldValues['BirthDate']:=datetimepicker1.Date;
if edit13.text<>" then
    dataset.FieldValues['OsymNo']:=edit13.Text;
if edit12.text<>" then
    dataset.FieldValues['Email']:=edit12.Text;

dataset.FieldValues['UpdateDate']:=Date;

if edit10.text<>" then
    dataset.FieldValues['Phone']:=edit10.Text;
datetimepicker1.Enabled:=True;
dataset.FieldValues['UpdateUser']:=username;

dataset.Post;
messagedlg('The student recorded successfully !',mtwarning,[mbok],0);
clearing(sender);
disabling(sender);
invisibility(sender);
edit1.SetFocus;
end;

end
else
begin
    messagedlg('Please Fill All The Required Fields !',mtwarning,[mbok],0);
    edit2.SetFocus;

```



```

end;
end;

procedure Tfrm_updstd.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
if not((upcase(key) in ['A'..'Z'])or(key=#8) or (key=#13)) then
begin
key:=#0;
messagedlg('Please Enter a Character ',mtwarning,[mbok],0);
end
else
key:=upcase(key);
end;

procedure Tfrm_updstd.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
if not((key in ['0'..'9'])or(key=#8) or (key=#13)) then
begin
key:=#0;
messagedlg('Please Enter a Numeric Character ',mtwarning,[mbok],0);
end;
if key=#13 then
button5.Click;
end;

procedure Tfrm_updstd.Memo1KeyPress(Sender: TObject; var Key: Char);
begin
key:=upcase(key);
end;

procedure Tfrm_updstd.Button6Click(Sender: TObject);
begin
frm_updstd.Close;
frm_password.Show;
end;

end.

```

unit Unit5;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Buttons, jpeg, ExtCtrls, StdCtrls;

type

```

Tfrm_teacher = class(TForm)
  Button1: TButton;
  Button2: TButton;

```

```
Button3: TButton;  
Image1: TImage;  
Image2: TImage;  
Label1: TLabel;  
procedure Button3Click(Sender: TObject);  
procedure formClose(Sender: TObject; var Action: TCloseAction);  
procedure Button1Click(Sender: TObject);  
procedure Button2Click(Sender: TObject);  
procedure Button4Click(Sender: TObject);
```

```
private  
  { Private declarations }  
public  
  { Public declarations }  
end;
```

```
var  
  frm_teacher: Tfrm_teacher;
```

```
implementation
```

```
uses Unit1, Unit6, Unit7, Unit26;
```

```
{ $R *.dfm }
```

```
procedure Tfrm_teacher.Button3Click(Sender: TObject);  
begin  
  frm_main.show;  
  frm_teacher.Close;  
end;
```

```
procedure Tfrm_teacher.FormClose(Sender: TObject;  
  var Action: TCloseAction);  
begin  
  frm_main.show;  
end;
```

```
procedure Tfrm_teacher.Button1Click(Sender: TObject);  
begin  
  frm_addteacher.show;  
  frm_teacher.Hide;  
end;
```

```
procedure Tfrm_teacher.Button2Click(Sender: TObject);  
begin  
  frm_updteacher.show;  
  frm_teacher.Hide;  
end;
```

```
procedure Tfrm_teacher.Button4Click(Sender: TObject);
```

```
begin
    frm_teacher.Close;
    frm_main.Show;
end;

end.
```

unit Unit6;

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, DBCtrls, Mask, ComCtrls, jpeg;
```

```
type
```

```
Tfrm_addteacher = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label4: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label17: TLabel;
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Memo1: TMemo;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    Button3: TButton;
    Button4: TButton;
    DateTimePicker1: TDateTimePicker;
    ComboBox3: TComboBox;
    Label15: TLabel;
    Label16: TLabel;
```

```

Label18: TLabel;
Button5: TButton;
Image2: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Edit2KeyPress(Sender: TObject; var Key: Char);
procedure Memo1KeyPress(Sender: TObject; var Key: Char);
procedure ComboBox3KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  frm_addteacher: Tfrm_addteacher;

```

implementation

```

uses Unit5, Unit1,unit26;

```

```

{$R *.dfm}

```

```

procedure clearing(Sender:Tobject);
begin
  frm_addteacher.Edit2.Clear;
  frm_addteacher.Edit3.Clear;
  frm_addteacher.Edit4.Clear;
  frm_addteacher.Edit5.Clear;
  frm_addteacher.Edit6.Clear;
  frm_addteacher.DateTimePicker1.Date:=StrToDate('01/01/1970');
  frm_addteacher.combobox2.Clear;
  frm_addteacher.combobox2.Items.Clear;
  frm_addteacher.Memo1.Clear;
  frm_addteacher.combobox1.Clear;
  frm_addteacher.combobox1.Items.Clear;
  frm_addteacher.combobox3.Clear;
  frm_addteacher.combobox3.Items.Clear;

end;

```

```

procedure invisibility(Sender:Tobject);
begin
  frm_addteacher.Edit2.Visible:=false;
  frm_addteacher.Edit3.Visible:=false;

```

```
frm_addteacher.Edit4.Visible:=false;  
frm_addteacher.Edit5.Visible:=false;  
frm_addteacher.Edit6.Visible:=false;  
frm_addteacher.DateTimePicker1.Visible:=false;  
frm_addteacher.combobox2.Visible:=false;  
frm_addteacher.memo1.Visible:=false;  
frm_addteacher.combobox1.Visible:=false;  
frm_addteacher.combobox3.visible:=false;
```

```
end;
```

```
procedure visibility(Sender:Tobject);
```

```
begin
```

```
    frm_addteacher.Edit2.Visible:=true;  
    frm_addteacher.Edit3.Visible:=true;  
    frm_addteacher.Edit4.Visible:=true;  
    frm_addteacher.Edit5.Visible:=true;  
    frm_addteacher.Edit6.Visible:=true;
```

```
    frm_addteacher.DateTimePicker1.Visible:=true;  
    frm_addteacher.memo1.Visible:=true;  
    frm_addteacher.combobox1.Visible:=true;  
    frm_addteacher.combobox2.Visible:=true;  
    frm_addteacher.combobox3.visible:=true;
```

```
end;
```

```
procedure enabling(Sender:Tobject);
```

```
begin
```

```
    frm_addteacher.edit2.enabled:=true;  
    frm_addteacher.edit3.enabled:=true;  
    frm_addteacher.edit4.enabled:=true;  
    frm_addteacher.edit5.enabled:=true;  
    frm_addteacher.edit6.enabled:=true;
```

```
    frm_addteacher.DateTimePicker1.enabled:=true;  
    frm_addteacher.combobox1.enabled:=true;  
    frm_addteacher.combobox2.enabled:=true;  
    frm_addteacher.combobox3.enabled:=true;  
    frm_addteacher.memo1.enabled:=true;
```

```
end;
```

```
procedure disabling(Sender:Tobject);
```

```
begin
```

```
    frm_addteacher.edit2.enabled:=true;  
    frm_addteacher.edit3.enabled:=true;  
    frm_addteacher.edit4.enabled:=true;  
    frm_addteacher.edit5.enabled:=true;  
    frm_addteacher.edit6.enabled:=true;
```

```
    frm_addteacher.DateTimePicker1.enabled:=false;
```



```

frm_addteacher.combobox1.enabled:=true;
frm_addteacher.combobox2.enabled:=true;
frm_addteacher.combobox3.enabled:=true;
frm_addteacher.memo1.enabled:=true;
end;

```

```

procedure Tfrm_addteacher.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  frm_teacher.show;
end;

```

```

procedure Tfrm_addteacher.Button2Click(Sender: TObject);
begin
  frm_main.show;
  frm_addteacher.Hide;
end;

```

```

procedure Tfrm_addteacher.Button1Click(Sender: TObject);
begin
  frm_teacher.Show;
  frm_addteacher.Hide;
end;

```

```

procedure Tfrm_addteacher.FormShow(Sender: TObject);
begin
  clearing(sender);
  edit1.ReadOnly:=true;
  edit2.SetFocus;

```

```

with frm_main.ADOQuery1 do
begin
  sql.clear;
  sql.Add('Select Max(TeacherId) as id from Teacher');
  open;
end;
edit1.Text:=IntToStr(frm_main.DataSource7.DataSet.FieldValues['id']+1);
combobox1.Items.Add('Male');
combobox1.Items.Add('Female');
combobox2.Items.Add('Prof.Dr. ');
combobox2.Items.Add('Assoc.Prof.Dr. ');
combobox2.Items.Add('Asist.Prof.Dr. ');
combobox2.Items.Add('Teacher');
with frm_main.ADOQuery1 do
begin
  sql.clear;
  sql.Add('Select DepartmentId from Department');
  open;
end;
frm_main.datasource7.DataSet.First;

```

```

while not frm_main.datasource7.DataSet.Eof do
begin
    combobox3.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
    frm_main.DataSource7.DataSet.Next;
end;

end;

procedure Tfrm_addteacher.Button4Click(Sender: TObject);
begin
    clearing(sender);
    edit2.SetFocus;
end;

procedure Tfrm_addteacher.Button3Click(Sender: TObject);
begin
    if ((edit2.Text<>")and(edit3.Text<>")and(combobox2.Text<>")and(combobox3.Text<>"))
    then
    begin
        with frm_main.DataSource3 do
        begin
            dataset.Edit;
            dataset.Append;
            // dataset.FieldValues['TeacherId']:=edit1.Text;
            if combobox1.text<>" then
                dataset.FieldValues['Sex']:=combobox1.Text;

            dataset.FieldValues['Tname']:=edit2.Text;

            dataset.FieldValues['Tsurname']:=edit3.Text;

            if combobox2.text<>" then
                dataset.FieldValues['Title']:=combobox2.Text;

            if combobox3.text<>" then
                dataset.FieldValues['DepartmentId']:=combobox3.Text;
            if memo1.Text<>" then
                dataset.FieldValues['Address']:=memo1.Text;
            if edit5.text<>" then
                dataset.FieldValues['Gsm']:=edit5.Text;
            if edit6.text<>" then
                dataset.FieldValues['Email']:=edit6.Text;
            if edit4.text<>" then
                dataset.FieldValues['Phone']:=edit4.Text;

            dataset.FieldValues['BirthDate']:=datetimepicker1.Date;

```

```

dataset.FieldValues['ProcessDate']:=Date;
dataset.FieldValues['ProcessUser']:=username;

dataset.Post;
messagedlg('The Teacher recorded successfully !',mtwarning,[mbok],0);
clearing(sender);
edit2.SetFocus;
edit1.ReadOnly:=false;
with frm_main.ADOQuery2 do
begin
  sql.clear;
  sql.Add('Select Max(TeacherId) as id from Teacher');
  open;
end;
// frm_main.DataSource7.DataSet:=frm_main.ADOQuery1;
edit1.Text:=IntToStr(frm_main.DataSource8.DataSet.FieldValues['id']+1);
edit1.ReadOnly:=true;
end;
else
begin
  messagedlg('Please Fill All The Required Fields !',mtwarning,[mbok],0);
  edit2.SetFocus;

end;
end;

procedure Tfrm_addteacher.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
  if not((upcase(key) in ['A'..'Z'])or(key=#8) or (key=#13)) then
  begin
    key:=#0;
    messagedlg('Please Enter a Character ',mtwarning,[mbok],0);
  end
  else
    key:=upcase(key);
end;

procedure Tfrm_addteacher.Memo1KeyPress(Sender: TObject; var Key: Char);
begin
  key:=upcase(key);
end;

procedure Tfrm_addteacher.ComboBox3KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if not((key=vk_down)or(key=vk_up)) then
    key:=vk_up;
end;

```

end.

unit Unit7;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, DBCtrls, Mask, ComCtrls, jpeg;

type

Tfrm_updteacher = class(TForm)

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label4: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label16: TLabel;

Label17: TLabel;

Label15: TLabel;

Label18: TLabel;

Label23: TLabel;

Label24: TLabel;

Button1: TButton;

Button2: TButton;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Memo1: TMemo;

ComboBox1: TComboBox;

ComboBox2: TComboBox;

Button3: TButton;

Button4: TButton;

DateTimePicker1: TDateTimePicker;

ComboBox3: TComboBox;

Edit7: TEdit;

Edit8: TEdit;

```

Button5: TButton;
Label19: TLabel;
Label20: TLabel;
Button6: TButton;
Image2: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure FormShow(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Edit2KeyPress(Sender: TObject; var Key: Char);
procedure Memo1KeyPress(Sender: TObject; var Key: Char);

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  frm_updteacher: Tfrm_updteacher;

```

implementation

```

uses Unit1, Unit5, unit26;

```

```

{$R *.dfm}

```

```

procedure clearing(Sender:Tobject);
begin
  frm_updteacher.Edit1.Clear;
  frm_updteacher.Edit2.Clear;
  frm_updteacher.Edit3.Clear;
  frm_updteacher.Edit4.Clear;
  frm_updteacher.Edit5.Clear;
  frm_updteacher.Edit6.Clear;
  frm_updteacher.Edit7.Clear;
  frm_updteacher.Edit8.Clear;
  frm_updteacher.DateTimePicker1.Date:=StrToDate('01/01/1970');
  frm_updteacher.combobox2.Clear;
  frm_updteacher.combobox2.Items.Clear;
  frm_updteacher.Memo1.Clear;
  frm_updteacher.combobox1.Clear;
  frm_updteacher.combobox1.Items.Clear;
  frm_updteacher.combobox3.Clear;
  frm_updteacher.combobox3.Items.Clear;
  frm_updteacher.Label23.Caption:="";

```



```
frm_updteacher.Label24.Caption:="";
```

```
end;
```

```
procedure invisibility(Sender:Tobject);  
begin
```

```
    frm_updteacher.Edit4.Visible:=false;  
    frm_updteacher.Edit5.Visible:=false;  
    frm_updteacher.Edit6.Visible:=false;  
    frm_updteacher.Edit7.Visible:=false;  
    frm_updteacher.Edit8.Visible:=false;  
    frm_updteacher.DateTimePicker1.Visible:=false;  
    frm_updteacher.combobox2.Visible:=false;  
    frm_updteacher.memo1.Visible:=false;  
    frm_updteacher.combobox1.Visible:=false;  
    frm_updteacher.combobox3.visible:=false;
```

```
end;
```

```
procedure visibility(Sender:Tobject);  
begin
```

```
    frm_updteacher.Edit4.Visible:=true;  
    frm_updteacher.Edit5.Visible:=true;  
    frm_updteacher.Edit6.Visible:=true;  
    frm_updteacher.Edit7.Visible:=true;  
    frm_updteacher.Edit8.Visible:=true;  
    frm_updteacher.DateTimePicker1.Visible:=true;  
    frm_updteacher.memo1.Visible:=true;  
    frm_updteacher.combobox1.Visible:=true;  
    frm_updteacher.combobox2.Visible:=true;  
    frm_updteacher.combobox3.visible:=true;
```

```
end;
```

```
procedure enabling(Sender:Tobject);  
begin
```

```
    frm_updteacher.edit4.enabled:=true;  
    frm_updteacher.edit5.enabled:=true;  
    frm_updteacher.edit6.enabled:=true;  
    frm_updteacher.edit7.enabled:=true;  
    frm_updteacher.edit8.enabled:=true;  
    frm_updteacher.DateTimePicker1.enabled:=true;  
    frm_updteacher.combobox1.enabled:=true;  
    frm_updteacher.combobox2.enabled:=true;  
    frm_updteacher.combobox3.enabled:=true;  
    frm_updteacher.memo1.enabled:=true;  
    frm_updteacher.Button3.Enabled:=true;
```

```
frm_updteacher.Button4.Enabled:=true;  
end;
```

```
procedure disabling(Sender:TObject);  
begin
```

```
    frm_updteacher.edit4.enabled:=false;  
    frm_updteacher.edit5.enabled:=false;  
    frm_updteacher.edit6.enabled:=false;  
    frm_updteacher.edit7.enabled:=false;  
    frm_updteacher.edit8.enabled:=false;  
    frm_updteacher.DateTimePicker1.enabled:=false;  
    frm_updteacher.combobox1.enabled:=false;  
    frm_updteacher.combobox2.enabled:=false;  
    frm_updteacher.combobox3.enabled:=false;  
    frm_updteacher.memo1.enabled:=false;  
    frm_updteacher.Button3.Enabled:=false;  
    frm_updteacher.Button4.Enabled:=false;  
end;
```

```
procedure Tfrm_updteacher.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
    frm_main.Show;  
end;
```

```
procedure Tfrm_updteacher.Button1Click(Sender: TObject);  
begin  
    frm_teacher.show;  
    frm_updteacher.Hide;  
end;
```

```
procedure Tfrm_updteacher.Button2Click(Sender: TObject);  
begin  
    frm_updteacher.Hide;  
    frm_main.Show;  
end;
```

```
procedure Tfrm_updteacher.Edit1KeyPress(Sender: TObject; var Key: Char);  
begin  
    if key=#13 then  
        edit1.OnExit(SENDER);  
end;
```

```
procedure Tfrm_updteacher.FormShow(Sender: TObject);  
begin  
    clearing(sender);  
    disabling(sender);  
    invisibility(sender);  
    edit1.SetFocus;  
end;
```

```

procedure Tfrm_updteacher.Button5Click(Sender: TObject);
begin
  if
    ((edit1.Text<>"")and((((edit2.text="")and(edit3.text=""))or((edit2.text<>"")and(edit3.text=""))or((edit2.text="")and(edit3.text<>""))))) then
    begin
      combobox1.Items.Add('Male');
      combobox1.Items.Add('Female');
      combobox2.Items.Add('Prof.Dr. ');
      combobox2.Items.Add('Assoc.Prof.Dr. ');
      combobox2.Items.Add('Asist.Prof.Dr. ');
      combobox2.Items.Add('Teacher');
      with frm_main.ADOQuery1 do
        begin
          sql.clear;
          sql.Add('Select DepartmentId from Department');
          open;
        end;
      frm_main.datasource7.DataSet.First;
      while not frm_main.datasource7.DataSet.Eof do
        begin
          combobox3.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
          frm_main.DataSource7.DataSet.Next;
        end;

      with frm_main.ADOQuery1 do
        begin
          sql.clear;
          sql.Add('Select * from Teacher where teacherId='+edit1.text);
          open;
        end;

      if frm_main.DataSource7.DataSet.RecordCount>0 then
        begin
          visibility(sender);
          enabling(sender);
          if frm_main.DataSource7.DataSet.FieldValues['Tname']<>null then
            edit4.Text:=frm_main.DataSource7.DataSet.FieldValues['Tname'];
          if frm_main.DataSource7.DataSet.FieldValues['Tsurname']<>null then
            edit5.Text:=frm_main.DataSource7.DataSet.FieldValues['Tsurname'];
          if frm_main.DataSource7.DataSet.FieldValues['Phone']<>null then
            edit6.Text:=frm_main.DataSource7.DataSet.FieldValues['phone'];
          if frm_main.DataSource7.DataSet.FieldValues['Gsm']<>null then
            edit7.Text:=frm_main.DataSource7.DataSet.FieldValues['Gsm'];
          if frm_main.DataSource7.DataSet.FieldValues['Email']<>null then
            edit8.Text:=frm_main.DataSource7.DataSet.FieldValues['Email'];
          if frm_main.DataSource7.DataSet.FieldValues['BirthDate']<>null then
            datetimerpicker1.Date:=frm_main.DataSource7.DataSet.FieldValues['BirthDate'];
          if frm_main.DataSource7.DataSet.FieldValues['Sex']<>null then

```

```

    combobox1.Text:=frm_main.DataSource7.DataSet.FieldValues['Sex'];
    if frm_main.DataSource7.DataSet.FieldValues['Title']<>null then
        combobox2.Text:=frm_main.DataSource7.DataSet.FieldValues['Title'];
    if frm_main.DataSource7.DataSet.FieldValues['DepartmentId']<>null then
        combobox3.Text:=frm_main.DataSource7.DataSet.FieldValues['DepartmentId'];
    if frm_main.DataSource7.DataSet.FieldValues['Address']<>null then
        memo1.Text:=frm_main.DataSource7.DataSet.FieldValues['Address'];

    if frm_main.DataSource7.DataSet.FieldValues['ProcessUser']<>null then
        label23.Caption:='The Teacher Is Recorded At
'+DatetoStr(frm_main.DataSource7.DataSet.FieldValues['ProcessDate'])+' By
'+frm_main.DataSource7.DataSet.FieldValues['ProcessUser'];
        if frm_main.DataSource7.DataSet.FieldValues['UpdateUser']<>null then
            label24.Caption:='The Teacher Record Was last Updated At
'+DatetoStr(frm_main.DataSource7.DataSet.FieldValues['UpdateDate'])+' By
'+frm_main.DataSource7.DataSet.FieldValues['UpdateUser'];

        edit4.SetFocus;
        edit1.ReadOnly:=true;
        edit2.ReadOnly:=true;
        edit3.ReadOnly:=true;
    end
else
begin
    messagedlg('Teacher Id is not found!!',mtinformation,[mbok],0);
    clearing(sender);
    edit1.SetFocus;

end;

end
else
begin
    if((edit1.text="")and((edit2.Text<>"")and(edit3.Text<>""))) then
        begin
            combobox1.Items.Add('Male');
            combobox1.Items.Add('Female');
            combobox2.Items.Add('Prof.Dr.');
            combobox2.Items.Add('Assoc.Prof.Dr.');
            combobox2.Items.Add('Asist.Prof.Dr.');
            combobox2.Items.Add('Teacher');
            with frm_main.ADOQuery1 do
                begin
                    sql.clear;
                    sql.Add('Select DepartmentId from Department');
                    open;
                end;
            frm_main.datasource7.DataSet.First;
            while not frm_main.datasource7.DataSet.Eof do
                begin

```



```

        combobox3.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
        frm_main.DataSource7.DataSet.Next;
    end;

    with frm_main.ADOQuery1 do
    begin
        sql.clear;
        sql.Add('Select * from Teacher where Tname="'+edit2.text+'" and
Tsurname="'+edit3.Text+'"');
        open;
    end;

    if frm_main.DataSource7.DataSet.RecordCount>0 then
    begin
        visibility(sender);
        enabling(sender);

        edit1.Text:=IntToStr(frm_main.DataSource7.DataSet.FieldValues['TeacherId']);
        if frm_main.DataSource7.DataSet.FieldValues['Tname']<>null then
            edit4.Text:=frm_main.DataSource7.DataSet.FieldValues['Tname'];
        if frm_main.DataSource7.DataSet.FieldValues['Tsurname']<>null then
            edit5.Text:=frm_main.DataSource7.DataSet.FieldValues['Tsurname'];
        if frm_main.DataSource7.DataSet.FieldValues['Phone']<>null then
            edit6.Text:=frm_main.DataSource7.DataSet.FieldValues['phone'];
        if frm_main.DataSource7.DataSet.FieldValues['Gsm']<>null then
            edit7.Text:=frm_main.DataSource7.DataSet.FieldValues['Gsm'];
        if frm_main.DataSource7.DataSet.FieldValues['Email']<>null then
            edit8.Text:=frm_main.DataSource7.DataSet.FieldValues['Email'];
        if frm_main.DataSource7.DataSet.FieldValues['BirthDate']<>null then
            datetimerpicker1.Date:=frm_main.DataSource7.DataSet.FieldValues['BirthDate'];
        if frm_main.DataSource7.DataSet.FieldValues['Sex']<>null then
            combobox1.Text:=frm_main.DataSource7.DataSet.FieldValues['Sex'];
        if frm_main.DataSource7.DataSet.FieldValues['Title']<>null then
            combobox2.Text:=frm_main.DataSource7.DataSet.FieldValues['Title'];
        if frm_main.DataSource7.DataSet.FieldValues['DepartmentId']<>null then
            combobox3.Text:=frm_main.DataSource7.DataSet.FieldValues['DepartmentId'];
        if frm_main.DataSource7.DataSet.FieldValues['Address']<>null then
            memo1.Text:=frm_main.DataSource7.DataSet.FieldValues['Address'];

        if frm_main.DataSource7.DataSet.FieldValues['ProcessUser']<>null then
            label23.Caption:='The Teacher Is Recorded At
'+DatetoStr(frm_main.DataSource7.DataSet.FieldValues['ProcessDate'])+' By
'+frm_main.DataSource7.DataSet.FieldValues['ProcessUser'];
            if frm_main.DataSource7.DataSet.FieldValues['UpdateUser']<>null then
                label24.Caption:='The Teacher Record Was last Updated At
'+DatetoStr(frm_main.DataSource7.DataSet.FieldValues['UpdateDate'])+' By
'+frm_main.DataSource7.DataSet.FieldValues['UpdateUser'];

        edit4.SetFocus;
        edit1.ReadOnly:=true;

```



```

edit2.ReadOnly:=true;
edit3.ReadOnly:=true;
end
else
begin
messagedlg('Teacher Name and Surname are not found!!',mtinformation,[mbok],0);
clearing(sender);
edit1.SetFocus;
edit1.ReadOnly:=FALSE;
edit2.ReadOnly:=FALSE;
edit3.ReadOnly:=FALSE;
end;

end
else
begin
messagedlg('Please Enter Teacher ID or Teacher Name And Surname
!',mtinformation,[mbok],0);
edit1.SetFocus;
clearing(sender);

end;

end;

end;

procedure Tfrm_updteacher.Button4Click(Sender: TObject);
begin
clearing(sender);
disabling(sender);
invisibility(sender);
edit1.SetFocus;
edit1.ReadOnly:=false;
edit2.ReadOnly:=false;
edit3.ReadOnly:=false;
end;

procedure Tfrm_updteacher.Button3Click(Sender: TObject);
begin
if ((edit4.Text<>")and(edit5.Text<>")and(combobox2.Text<>")and(combobox3.Text<>"))
then
begin
with frm_main.ADOquery1 do
begin
sql.clear;
sql.Add('Select * from Teacher where TeacherId='+edit1.text);
open;
end;
with frm_main.DataSource7 do

```

```

begin
edit1.ReadOnly:=false;
edit2.ReadOnly:=false;
edit3.ReadOnly:=false;

dataset.Edit;

if combobox1.text<>" then
dataset.FieldValues['sex']:=combobox1.Text;
dataset.FieldValues['Tname']:=edit4.Text;
dataset.FieldValues['Tsurname']:=edit5.Text;
if combobox2.text<>" then
dataset.FieldValues['Title']:=combobox2.Text;
if combobox3.text<>" then
dataset.FieldValues['DepartmentId']:=combobox3.Text;
if memo1.Text<>" then
dataset.FieldValues['Address']:=memo1.Text;
if edit7.text<>" then
dataset.FieldValues['Gsm']:=edit7.Text;
if edit8.text<>" then
dataset.FieldValues['Email']:=edit8.Text;

dataset.FieldValues['BirthDate']:=Datetimepicker1.Date;

if edit6.text<>" then
dataset.FieldValues['Phone']:=edit6.Text;
datetimepicker1.Enabled:=True;
dataset.FieldValues['UpdateUser']:=username;
dataset.FieldValues['UpdateDate']:=Date;

dataset.Post;
messagedlg('The Teacher Information Recorded successfully !',mtInformation,[mbok],0);
clearing(sender);
disabling(sender);
invisibility(sender);
edit1.SetFocus;
end;

end
else
begin
messagedlg('Please Fill All The Required Fields !',mtwarning,[mbok],0);
edit4.SetFocus;
end;
end;

procedure Tfrm_updteacher.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
if not((uppercase(key) in ['A'..'Z'])or(key=#8) or (key=#13)) then
begin

```

```

key:=#0;
messagedlg('Please Enter a Character ',mtwarning,[mbok],0);
end
else
key:=upcase(key);
end;

```

```

procedure Tfrm_updteacher.Memo1KeyPress(Sender: TObject; var Key: Char);
begin
key:=upcase(key);
end;
end.

```

unit Unit8;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Buttons, ExtCtrls, StdCtrls, jpeg;

type

Tfrm_course = class(TForm)

Image1: TImage;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Label1: TLabel;
Image2: TImage;

procedure Button3Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

frm_course: Tfrm_course;

implementation

uses Unit1, Unit9, Unit10;

{ \$R *.dfm }

```
procedure Tfrm_course.Button3Click(Sender: TObject);
begin
    frm_main.show;
    frm_course.Close;
end;
```

```
procedure Tfrm_course.Button1Click(Sender: TObject);
begin
    frm_addcourse.show;
    frm_course.Close;
end;
```

```
procedure Tfrm_course.Button2Click(Sender: TObject);
begin
    frm_updcourse.show;
    frm_course.Close;
end;
```

```
end.
```

unit Unit9;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, DBCtrls, StdCtrls, Mask, jpeg, Grids, DBGrids,
ComCtrls, DB, ADODB;

type

```
Tfrm_addcourse = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    ComboBox3: TComboBox;
    Button3: TButton;
```

```

Button4: TButton;
DBGrid1: TDBGrid;
ADOTable1: TADOTable;
DBGrid2: TDBGrid;
DataSource1: TDataSource;
Label8: TLabel;
Image2: TImage;
Label9: TLabel;
Label10: TLabel;
Button5: TButton;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure ComboBox1Exit(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frm_addcourse: Tfrm_addcourse;

implementation

uses Unit8, Unit1, unit26;

{$R *.dfm}

procedure clearing(Sender: TObject);
begin
  frm_addcourse.Edit2.Clear;
  frm_addcourse.Edit3.Clear;
  frm_addcourse.Edit1.Clear;
  frm_addcourse.combobox2.Clear;
  frm_addcourse.combobox2.Items.Clear;

  frm_addcourse.combobox1.Clear;
  frm_addcourse.combobox1.Items.Clear;
  frm_addcourse.combobox3.Clear;
  frm_addcourse.combobox3.Items.Clear;

  frm_addcourse.combobox3.Items.Add(inttostr(0));
  frm_addcourse.combobox3.Items.Add(Inttostr(1));
  frm_addcourse.combobox3.Items.Add(Inttostr(2));
  frm_addcourse.combobox3.Items.Add(Inttostr(3));
  frm_addcourse.combobox3.Items.Add(Inttostr(4));

```



```
frm_addcourse.combobox3.Items.Add(Inttostr(5));  
frm_addcourse.combobox3.Items.Add(Inttostr(6));
```

```
end;
```

```
procedure invisibility(Sender:Tobject);  
begin  
    frm_addcourse.Edit2.Visible:=false;  
    frm_addcourse.Edit3.Visible:=false;  
    frm_addcourse.Edit1.Visible:=false;  
    frm_addcourse.combobox2.Visible:=false;  
    frm_addcourse.combobox1.Visible:=false;  
    frm_addcourse.combobox3.visible:=false;
```

```
end;
```

```
procedure visibility(Sender:Tobject);  
begin  
    frm_addcourse.Edit2.Visible:=true;  
    frm_addcourse.Edit3.Visible:=true;  
    frm_addcourse.Edit1.Visible:=true;  
    frm_addcourse.combobox1.Visible:=true;  
    frm_addcourse.combobox2.Visible:=true;  
    frm_addcourse.combobox3.visible:=true;  
end;
```

```
procedure enabling(Sender:Tobject);  
begin  
    frm_addcourse.edit2.enabled:=true;  
    frm_addcourse.edit3.enabled:=true;  
    frm_addcourse.edit1.enabled:=true;  
    frm_addcourse.combobox1.enabled:=true;  
    frm_addcourse.combobox2.enabled:=true;  
    frm_addcourse.combobox3.enabled:=true;
```

```
end;
```

```
procedure disabling(Sender:Tobject);  
begin  
    frm_addcourse.edit2.enabled:=false;  
    frm_addcourse.edit3.enabled:=false;  
    // frm_addcourse.edit1.enabled:=false;  
    frm_addcourse.combobox1.enabled:=false;  
    frm_addcourse.combobox2.enabled:=false;  
    frm_addcourse.combobox3.enabled:=false;  
    frm_addcourse.Text1.Text:=frm_addcourse.combobox1.Text+";"  
end;
```

```

procedure Tfrm_addcourse.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  frm_course.Show;
end;

```

```

procedure Tfrm_addcourse.Button1Click(Sender: TObject);
begin
  frm_addcourse.Close;
end;

```

```

procedure Tfrm_addcourse.Button2Click(Sender: TObject);
begin
  frm_main.Show;
end;

```

```

procedure Tfrm_addcourse.FormShow(Sender: TObject);
begin
  clearing(sender);
  // disabling(sender);
  edit1.SetFocus;
  with frm_main.ADOQuery1 do
  begin
    sql.clear;
    sql.Add('Select DepartmentId from Department');
    open;
  end;
  frm_main.datasource7.DataSet.First;
  while not frm_main.datasource7.DataSet.Eof do
  begin
    combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
    frm_main.DataSource7.DataSet.Next;
  end;

```

end;

```

procedure Tfrm_addcourse.ComboBox1Exit(Sender: TObject);
begin
  with frm_main.ADOQuery4 do
  begin
    sql.clear;
    sql.Add('Select TeacherId,Tname+" "+Tsurname As [Teacher Name&Surname] from
Teacher where DepartmentId="'+combobox1.Text+'");
    open;

```

end;

```
dbgrid1.DataSource:=frm_main.DataSource10;  
combobox2.Items.Clear;  
frm_main.datasource10.DataSet.First;  
while not frm_main.datasource10.DataSet.Eof do  
begin  
    combobox2.Items.Add(frm_main.datasource10.DataSet.FieldValues["TeacherId"]);  
    frm_main.DataSource10.DataSet.Next;  
end;
```

end;

```
procedure Tfrm_addcourse.Button4Click(Sender: TObject);  
begin  
    clearing(sender);  
    with frm_main.ADOQuery1 do  
    begin  
        sql.clear;  
        sql.Add('Select DepartmentId from Department');  
        open;  
    end;  
    frm_main.datasource7.DataSet.First;  
    while not frm_main.datasource7.DataSet.Eof do  
    begin  
        combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues["DepartmentId"]);  
        frm_main.DataSource7.DataSet.Next;  
    end;  
    edit1.SetFocus;  
    dbgrid1.DataSource:=frm_main.DataSource13;  
    dbgrid1.Repaint;  
end;
```

```
procedure Tfrm_addcourse.Button3Click(Sender: TObject);  
begin  
    with frm_main.ADOQuery3 do  
    begin  
        sql.clear;  
        sql.Add('Select CourseCode from course where coursecode="'+edit1.text+'"');  
        open;  
    end;  
    if frm_main.DataSource9.DataSet.RecordCount=0 then  
    begin  
        if  
        ((edit2.Text<>")and(edit3.Text<>")and(combobox1.Text<>")and(combobox2.Text<>")and(co  
        mbobox3.Text<>")) then  
        begin  
            with frm_main.DataSource6 do  
            begin  
                dataset.Edit;
```

```

dataset.Append;
dataset.FieldValues['Coursecode']:=edit1.Text;
if combobox1.text<>" then
dataset.FieldValues['DepartmentId']:=combobox1.Text;

dataset.FieldValues['Coursename']:=edit2.Text;

dataset.FieldValues['Coursecontent']:=edit3.Text;

if combobox2.text<>" then
dataset.FieldValues['TeacherId']:=StrToint(combobox2.Text);

if combobox3.text<>" then
dataset.FieldValues['Credit']:=Strtoint(combobox3.Text);
dataset.FieldValues['ProcessDate']:=Date;
dataset.FieldValues['ProcessUser']:=username;

dataset.Post;
messagedlg('The Course recorded successfully !',mtwarning,[mbok],0);
clearing(sender);
edit1.SetFocus;

with frm_main.ADOQuery1 do
begin
sql.clear;
sql.Add('Select DepartmentId from Department');
open;
end;
frm_main.datasource7.DataSet.First;
while not frm_main.datasource7.DataSet.Eof do
begin
combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
frm_main.DataSource7.DataSet.Next;
end;
end;
else
begin
messagedlg('Please Fill All The Required Fields !',mtwarning,[mbok],0);
edit2.SetFocus;
end;

end
else
begin
messagedlg('This Course Is Already Recorded !',mtwarning,[mbok],0);
edit1.SetFocus;
end;

end;

```


end.

unit Unit10;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, DBCtrls, ExtCtrls, Mask, Grids, DBGrids, ComCtrls,
jpeg;

type

Tfrm_updcourse = class(TForm)

Image1: TImage;

Label15: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Button1: TButton;

Button2: TButton;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Button3: TButton;

Button4: TButton;

DBGrid1: TDBGrid;

ComboBox1: TComboBox;

ComboBox2: TComboBox;

ComboBox3: TComboBox;

Button5: TButton;

Button6: TButton;

DBGrid2: TDBGrid;

Label1: TLabel;

Label9: TLabel;

Label10: TLabel;

Image2: TImage;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure ComboBox1Exit(Sender: TObject);

procedure ComboBox3Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private


```

{ Private declarations }
public
{ Public declarations }
end;

var
frm_updcourse: Tfrm_updcourse;

implementation

uses Unit1, Unit8,unit26, Unit9;

{$R *.dfm}

procedure clearing(Sender:Tobject);
begin
    frm_updcourse.Edit2.Clear;
    frm_updcourse.Edit3.Clear;
    frm_updcourse.Edit1.Clear;
    frm_updcourse.combobox2.Clear;
    frm_updcourse.combobox2.Items.Clear;
    frm_updcourse.Edit1.readonly:=false;
    frm_updcourse.Button3.Enabled:=false;

    frm_updcourse.combobox1.Clear;
    frm_updcourse.combobox1.Items.Clear;
    frm_updcourse.combobox3.Clear;
    frm_updcourse.combobox3.Items.Clear;

    frm_updcourse.combobox3.Items.Add(inttostr(0));
    frm_updcourse.combobox3.Items.Add(Inttostr(1));
    frm_updcourse.combobox3.Items.Add(Inttostr(2));
    frm_updcourse.combobox3.Items.Add(Inttostr(3));
    frm_updcourse.combobox3.Items.Add(Inttostr(4));
    frm_updcourse.combobox3.Items.Add(Inttostr(5));
    frm_updcourse.combobox3.Items.Add(Inttostr(6));
    frm_updcourse.dbgrid1.DataSource:=frm_main.DataSource13;
    frm_updcourse.dbgrid1.repaint;

end;

procedure invisibility(Sender:Tobject);
begin
    frm_updcourse.Edit2.Visible:=false;
    frm_updcourse.Edit3.Visible:=false;
    frm_updcourse.Edit1.Visible:=false;
    frm_updcourse.combobox2.Visible:=false;
    frm_updcourse.combobox1.Visible:=false;
    frm_updcourse.combobox3.visible:=false;

```

end;

procedure visibility(Sender:TObject);
begin

frm_updcourse.Edit2.Visible:=true;
frm_updcourse.Edit3.Visible:=true;
frm_updcourse.Edit1.Visible:=true;
frm_updcourse.combobox1.Visible:=true;
frm_updcourse.combobox2.Visible:=true;
frm_updcourse.combobox3.visible:=true;

end;

procedure enabling(Sender:TObject);
begin

frm_updcourse.edit2.enabled:=true;
frm_updcourse.edit3.enabled:=true;
frm_updcourse.edit1.enabled:=true;
frm_updcourse.combobox1.enabled:=true;
frm_updcourse.combobox2.enabled:=true;
frm_updcourse.combobox3.enabled:=true;

end;

procedure disabling(Sender:TObject);
begin

frm_updcourse.edit2.enabled:=false;
frm_updcourse.edit3.enabled:=false;
frm_addcourse.edit1.enabled:=TRUE;
frm_updcourse.combobox1.enabled:=false;
frm_updcourse.combobox2.enabled:=false;
frm_updcourse.combobox3.enabled:=false;

end;

procedure Tfrm_updcourse.FormClose(Sender: TObject;
var Action: TCloseAction);

begin
frm_course.show;
end;

procedure Tfrm_updcourse.Button1Click(Sender: TObject);
begin

frm_course.show;
frm_updcourse.Hide;
end;

procedure Tfrm_updcourse.Button2Click(Sender: TObject);

begin

frm_updcourse.Hide;
frm_main.Show;
end;

procedure Tfrm_updcourse.Button4Click(Sender: TObject);
begin

frm_updcourse.Edit2.Clear;
frm_updcourse.Edit3.Clear;
frm_updcourse.Edit1.Clear;
frm_updcourse.combobox2.Clear;
frm_updcourse.combobox2.Items.Clear;
frm_updcourse.Edit1.Enabled:=true;
frm_updcourse.Button3.Enabled:=false;

frm_updcourse.combobox1.Clear;
frm_updcourse.combobox1.Items.Clear;
frm_updcourse.combobox3.Clear;
frm_updcourse.combobox3.Items.Clear;

frm_updcourse.combobox3.Items.Add(inttostr(0));
frm_updcourse.combobox3.Items.Add(Inttostr(1));
frm_updcourse.combobox3.Items.Add(Inttostr(2));
frm_updcourse.combobox3.Items.Add(Inttostr(3));
frm_updcourse.combobox3.Items.Add(Inttostr(4));
frm_updcourse.combobox3.Items.Add(Inttostr(5));
frm_updcourse.combobox3.Items.Add(Inttostr(6));

with frm_main.ADOQuery1 do

begin

sql.clear;
sql.Add('Select DepartmentId from Department');
open;
end;

frm_main.datasource7.DataSet.First;

while not frm_main.datasource7.DataSet.Eof do

begin

combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);

frm_main.DataSource7.DataSet.Next;

end;

edit1.SetFocus;

dbgrid1.DataSource:=frm_main.DataSource13;

dbgrid1.Repaint;

Disabling(sender);

end;

procedure Tfrm_updcourse.FormShow(Sender: TObject);

begin

clearing(sender);

edit1.SetFocus;

Disabling(sender);

```

with frm_main.ADOQuery1 do
begin
  sql.clear;
  sql.Add('Select DepartmentId from Department');
  open;
end;
frm_main.datasource7.DataSet.First;
while not frm_main.datasource7.DataSet.Eof do
begin
  combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
  frm_main.DataSource7.DataSet.Next;
end;
end;

procedure Tfrm_updcourse.Button5Click(Sender: TObject);
begin
  with frm_main.ADOQuery3 do
  begin
    sql.clear;
    sql.Add('Select * from course where coursecode="'+edit1.text+"'");
    open;
  end;
  if frm_main.DataSource9.DataSet.RecordCount<>0 then
  begin
    enabling(sender);
    button3.Enabled:=true;
    with frm_main.ADOQuery4 do
    begin
      sql.clear;
      sql.Add('Select TeacherId,Tname+" "+Tsurname As [Teacher Name&Surname] from
Teacher where
DepartmentId="'+frm_main.DataSource9.DataSet.FieldValues['DepartmentId']+"'");
      open;
    end;

    dbgrid1.DataSource:=frm_main.DataSource10;
    combobox2.Items.Clear;
    frm_main.datasource10.DataSet.First;
    while not frm_main.datasource10.DataSet.Eof do
    begin
      combobox2.Items.Add(frm_main.datasource10.DataSet.FieldValues['TeacherId']);
      frm_main.DataSource10.DataSet.Next;
    end;
    edit1.ReadOnly:=true;
    with frm_main.DataSource9 do
    begin
      edit2.Text:=dataset.FieldValues['Coursename'];
      edit3.Text:=dataset.FieldValues['Coursecontent'];
      combobox1.Text:=dataset.FieldValues['departmentId'];
      combobox2.Text:=inttostr(dataset.FieldValues['TeacherId']);
    end;
  end;
end;

```

```

    combobox3.Text:=inttostr(dataset.FieldValues['credit']);
end;

end
else
begin
    messagedlg('The Course Is Not Found!',mtwarning,[mbok],0);
    clearing(sender);
    edit1.SetFocus;

end;

end;

procedure Tfrm_updcourse.ComboBox1Exit(Sender: TObject);
begin
    with frm_main.ADOQuery4 do
        begin
            sql.clear;
            sql.Add('Select TeacherId,Tname+" "+Tsurname As [Teacher Name&Surname] from
Teacher where DepartmentId="'+combobox1.Text+'"');
            open;
        end;

        dbgrid1.DataSource:=frm_main.DataSource10;
        combobox2.Items.Clear;
        frm_main.datasource10.DataSet.First;
        while not frm_main.datasource10.DataSet.Eof do
            begin
                combobox2.Items.Add(frm_main.datasource10.DataSet.FieldValues['TeacherId']);
                frm_main.DataSource10.DataSet.Next;
            end;

end;

procedure Tfrm_updcourse.ComboBox3Click(Sender: TObject);
begin
    frm_addcourse.combobox3.Items.Add(inttostr(0));
    frm_addcourse.combobox3.Items.Add(inttostr(1));
    frm_addcourse.combobox3.Items.Add(inttostr(2));
    frm_addcourse.combobox3.Items.Add(inttostr(3));
    frm_addcourse.combobox3.Items.Add(inttostr(4));
    frm_addcourse.combobox3.Items.Add(inttostr(5));
    frm_addcourse.combobox3.Items.Add(inttostr(6))
end;

procedure Tfrm_updcourse.Button3Click(Sender: TObject);

```



```

begin
  if
    ((edit2.Text<>"")and(edit3.Text<>"")and(combobox1.Text<>"")and(combobox2.Text<>"")and(co
mbobox3.Text<>"")) then
    begin
      with frm_main.DataSource6 do
        begin
          dataset.Edit;

// dataset.FieldValues['Coursecode']:=edit1.Text;
          if combobox1.text<>" then
            dataset.FieldValues['DepartmentId']:=combobox1.Text;
          dataset.FieldValues['CourseName']:=edit2.Text;
          dataset.FieldValues['CourseContent']:=edit3.Text;

          if combobox2.text<>" then
            dataset.FieldValues['TeacherId']:=StrToint(combobox2.Text);

          if combobox3.text<>" then
            dataset.FieldValues['Credit']:=Strtoint(combobox3.Text);
          dataset.FieldValues['ProcessDate']:=Date;
          dataset.FieldValues['ProcessUser']:=username;

          dataset.Post;
          messagedlg('The Course Updated Successfully !',mtwarning,[mbok],0);
          clearing(sender);
          edit1.SetFocus;

          with frm_main.ADOQuery1 do
            begin
              sql.clear;
              sql.Add('Select DepartmentId from Department');
              open;
            end;
            frm_main.datasource7.DataSet.First;
            while not frm_main.datasource7.DataSet.Eof do
              begin
                combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
                frm_main.DataSource7.DataSet.Next;
              end;
            end;
          end
        else
          begin
            messagedlg('Please Fill All The Required Fields !',mtwarning,[mbok],0);
            edit2.SetFocus;
          end;
        end;
      end;
    end;
  end;
end;

```

end.

unit Unit11;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Buttons, jpeg, ExtCtrls, StdCtrls, ComCtrls;

type

Tfrm_dept = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label1: TLabel;

Image2: TImage;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

frm_dept: Tfrm_dept;

implementation

uses Unit1, Unit12, Unit13, Unit26;

{ \$R *.dfm }

procedure Tfrm_dept.FormClose(Sender: TObject; var Action: TCloseAction);

begin

frm_main.show;

end;

procedure Tfrm_dept.Button1Click(Sender: TObject);

begin

frm_adddept.show;

frm_dept.Hide;

end;

```
procedure Tfrm_dept.Button2Click(Sender: TObject);
```

```
begin  
    frm_updept.show;  
    frm_dept.Hide;  
end;
```

```
procedure Tfrm_dept.Button3Click(Sender: TObject);
```

```
begin  
    frm_dept.Close;  
    frm_main.show;  
end;  
end.
```

unit Unit12;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, ExtCtrls, DBCtrls, Mask, ComCtrls, jpeg, Grids,
 DBGrids, DB, ADODB;

type

```
Tfrm_adddept = class(TForm)  
    Button1: TButton;  
    Button2: TButton;  
    Edit1: TEdit;  
    Edit2: TEdit;  
    Button3: TButton;  
    Button4: TButton;  
    Button5: TButton;  
    Image1: TImage;  
    Label3: TLabel;  
    Label4: TLabel;  
    Label1: TLabel;  
    Image2: TImage;  
    Label2: TLabel;  
    DBGrid1: TDBGrid;  
    Button6: TButton;  
    procedure Button1Click(Sender: TObject);  
    procedure FormClose(Sender: TObject; var Action: TCloseAction);  
    procedure Button2Click(Sender: TObject);  
    procedure FormShow(Sender: TObject);  
    procedure Button5Click(Sender: TObject);  
    procedure Button3Click(Sender: TObject);  
    procedure Button4Click(Sender: TObject);  
private  
    { Private declarations }  
public
```

```
{ Public declarations }  
end;  
  
var  
    frm_adddept: Tfrm_adddept;  
  
implementation  
  
uses Unit1, Unit11, unit26;  
  
{$R *.dfm}  
  
procedure clearing(Sender:Tobject);  
begin  
    frm_adddept.Edit1.Clear;  
    frm_adddept.Edit2.Clear;  
end;  
  
procedure invisibility(Sender:Tobject);  
begin  
    frm_adddept.Edit2.Visible:=false;  
    frm_adddept.button3.Visible:=true;  
    frm_adddept.button4.Visible:=true;  
end;  
  
procedure visibility(Sender:Tobject);  
begin  
    frm_adddept.Edit2.Visible:=true;  
    frm_adddept.button3.Visible:=true;  
    frm_adddept.button4.Visible:=true;  
end;  
  
procedure enabling(Sender:Tobject);  
begin  
    frm_adddept.Edit2.Enabled:=true;  
    frm_adddept.button3.Enabled:=true;  
    frm_adddept.button4.Enabled:=true;  
end;  
  
procedure disabling(Sender:Tobject);  
begin  
    frm_adddept.Edit2.Enabled:=false;  
    frm_adddept.button3.Enabled:=false;  
    frm_adddept.button4.Enabled:=false;  
end;  
  
procedure Tfrm_adddept.Button1Click(Sender: TObject);  
begin  
    frm_adddept.hide;  
    frm_main.show;
```

```

end;

procedure Tfrm_adddept.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  frm_dept.show;
end;

procedure Tfrm_adddept.Button2Click(Sender: TObject);
begin
  frm_adddept.Close;
end;

procedure Tfrm_adddept.FormShow(Sender: TObject);
begin
  clearing(sender);
  disabling(sender);
  invisibility(sender);
  edit1.SetFocus;
end;

procedure Tfrm_adddept.Button5Click(Sender: TObject);
begin
  if edit1.text<>" then
  begin
    with frm_main.ADOQuery1 do
    begin
      sql.clear;
      sql.Add('Select DepartmentId from Department where DepartmentId="'+edit1.text+"'");
      open;
      end;
    if frm_main.DataSource7.DataSet.RecordCount=0 then
    begin
      visibility(sender);
      enabling(sender);
      edit2.SetFocus;
    end
  else
    begin
      messagedlg('This Department ID Is Already Recorded !',mtwarning,[mbok],0);
      edit1.SetFocus;
      edit1.Clear;
      //dbgrid1.Update;
    end;
  end
else
  begin
    messagedlg('Please Enter Department Id For To New Department !',mtwarning,[mbok],0);
    edit1.Clear;
  end

```



```

edit1.SetFocus;

end;
end;

procedure Tfrm_adddept.Button3Click(Sender: TObject);
begin
if ((edit1.Text<>"")and(edit2.Text<>"")) then
begin
with frm_main.DataSource5 do
begin
dataset.Edit;
dataset.Append;
if edit2.text<>" then
dataset.FieldValues['DepartmentId']:=edit1.Text;
if edit2.text<>" then
dataset.FieldValues['DepartmentName']:=edit2.Text;

dataset.FieldValues['ProcessUser']:=username;
dataset.FieldValues['ProcessDate']:=Date;

dataset.Post;
messagedlg('The Department recorded successfully !',mtwarning,[mbok],0);
clearing(sender);
disabling(sender);
invisibility(sender);
edit1.SetFocus;
end;

end
else
begin
messagedlg('Please Fill All Fields !',mtwarning,[mbok],0);
edit1.SetFocus;
end;

dbgrid1.Update;
end;

procedure Tfrm_adddept.Button4Click(Sender: TObject);
begin
clearing(sender);
disabling(sender);
invisibility(sender);
edit1.SetFocus;
end;

end.

```

unit Unit13;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, DBCtrls, Mask, ComCtrls, jpeg, DB, ADODB,
Grids, DBGrids;

type

Tfrm_upddept = class(TForm)

 Button1: TButton;

 Button2: TButton;

 Edit1: TEdit;

 StatusBar1: TStatusBar;

 Button3: TButton;

 Edit2: TEdit;

 Edit3: TEdit;

 Button4: TButton;

 Button5: TButton;

 Image1: TImage;

 Label4: TLabel;

 Label7: TLabel;

 Label8: TLabel;

 Image2: TImage;

 Label1: TLabel;

 Label2: TLabel;

 DBGrid1: TDBGrid;

 Button6: TButton;

 procedure FormClose(Sender: TObject; var Action: TCloseAction);

 procedure Button2Click(Sender: TObject);

 procedure Button1Click(Sender: TObject);

 procedure FormShow(Sender: TObject);

 procedure Button3Click(Sender: TObject);

 procedure Button4Click(Sender: TObject);

 procedure Button5Click(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 frm_upddept: Tfrm_upddept;

implementation

```
uses Unit11, Unit1, unit26;
```

```
{ $R *.dfm }
```

```
procedure Tfrm_updept.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
    frm_dept.show;  
end;
```

```
procedure Tfrm_updept.Button2Click(Sender: TObject);  
begin  
    frm_updept.Close;  
    frm_dept.show;  
end;
```

```
procedure Tfrm_updept.Button1Click(Sender: TObject);  
begin  
    frm_updept.Close;  
    frm_main.Show ;  
end;
```

```
procedure Tfrm_updept.FormShow(Sender: TObject);  
begin
```

```
    edit1.Clear;  
    edit2.Clear;  
    edit3.Clear;  
    edit1.ReadOnly:=false;  
    edit1.SetFocus;  
    edit2.Visible:=false;  
    edit3.Visible:=false;
```

```
end;
```

```
procedure Tfrm_updept.Button3Click(Sender: TObject);  
begin
```

```
    dbgrid1.SelectedField;  
    if (edit1.Text<>") then  
    begin  
        with frm_main.ADOQuery1 do  
        begin  
            sql.clear;  
            sql.Add('Select * from Department where DepartmentId="'+edit1.text+'"');  
            open;  
        end;
```

```
    if frm_main.DataSource7.DataSet.RecordCount>0 then  
    begin  
        edit2.Visible:=true;
```

```

edit3.Visible:=true;
button4.Enabled:=true;
button5.Enabled:=true;

if frm_main.DataSource7.DataSet.FieldValues['DepartmentId']<>null then
    edit2.Text:=frm_main.DataSource7.DataSet.FieldValues['DepartmentId'];
if frm_main.DataSource7.DataSet.FieldValues['DepartmentName']<>null then
    edit3.Text:=frm_main.DataSource7.DataSet.FieldValues['DepartmentName'];

edit1.ReadOnly:=true;

end
else
begin
    messagedlg('Department ID is not found !',mtinformation,[mbok],0);
    edit1.SetFocus;
end;

end
else
begin
    messagedlg('Please Enter Department ID !',mtinformation,[mbok],0);
    edit1.SetFocus;
    edit1.Clear;

end;

end;

end;

procedure Tfrm_upddept.Button4Click(Sender: TObject);
begin
    if edit1.Text<>edit2.Text then
    begin
        with frm_main.ADOQuery2 do
        begin
            sql.clear;
            sql.Add('Select departmentID from Department where DepartmentId="'+edit2.text+'"');
            open;

        end;
        if frm_main.DataSource8.DataSet.RecordCount=0 then
        begin
            with frm_main.DataSource7 do
            begin
                edit1.ReadOnly:=false;
                dataset.Edit;
                //dataset.UpdateRecord;
                if edit2.Text<>" then
                    dataset.FieldValues['DepartmentId']:=edit2.Text;
            end;
        end;
    end;
end;

```

```

dataset.FieldValues['UpdateUser']:=username;
dataset.FieldValues['Updatedate']:=date;
dbgrid1.Update;
if edit3.text<>" then
    dataset.FieldValues['DepartmentName']:=Edit3.Text;
dataset.UpdateRecord;
dataset.post;
dbgrid1.Update;
{ with frm_main.ADOQuery3 do
begin
    sql.clear;
    sql.Add('Select * from TEACHER where DepartmentId="'+edit1.text+'"');
    open;
end;
if frm_main.DataSource9.DataSet.RecordCount>0 then
begin
    with frm_main.DataSource9 do
    begin
        dataset.edit;
        dataset.First;
        while not (dataset.Eof) do
        begin
            dataset.FieldValues['DepartmentId']:=edit2.Text;
            dataset.Next;
        end;
        dataset.Post;
    END;

    messagedlg('The Department Information in All Tables are Updated Successfully
!',mtinformation,[mbok],0);

    edit1.SetFocus;
    edit1.Clear;
    edit2.Clear;
    edit3.Clear;
    edit2.Visible:=false;
    edit3.Visible:=false;
    button4.Visible:=false;
    button5.Visible:=false;
end; }

end;
dbgrid1.Update;
end

else
begin
    messagedlg('Duplicate Department ID is not allowed !',mtinformation,[mbok],0);
    edit2.SetFocus;
    dbgrid1.Update;

```



```

end;
dbgrid1.Update;
end

else
begin
with frm_main.DataSource7 do
begin
dataset.edit;
dataset.FieldValues['UpdateUser']:=username;
dataset.FieldValues['Updatedate']:=date;
if edit3.text<>" then
dataset.FieldValues['DepartmentName']:=Edit3.Text;
dataset.post;
messagedlg('The Department Information Recorded Successfully
!',mtinformation,[mbok],0);
edit1.SetFocus;
edit1.ReadOnly:=false;
edit1.Clear;
edit2.Clear;
edit3.Clear;
edit2.Visible:=false;
edit3.Visible:=false;
button4.Enabled:=false;
button5.Enabled:=false;
dbgrid1.Update;

end;
dbgrid1.Update;
end;

dbgrid1.Update;
dbgrid1.Refresh;
end;
procedure Tfrm_upddept.Button5Click(Sender: TObject);
begin
edit1.SetFocus;
edit1.ReadOnly:=false;
edit1.Clear;
edit2.Clear;
edit3.Clear;
edit2.Visible:=false;
edit3.Visible:=false;
button4.Enabled:=false;
button5.Enabled:=false;
end;

end.

```

unit Unit14;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls, Mask, DBCtrls, Grids, DBGrids, jpeg;

type

Tfrm_trans = class(TForm)

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Label24: TLabel;

Label25: TLabel;

Label26: TLabel;

Label27: TLabel;

Label28: TLabel;

Label29: TLabel;

Label30: TLabel;

Label31: TLabel;

Label15: TLabel;

Label32: TLabel;

Label45: TLabel;

Label46: TLabel;

Edit1: TEdit;

Button1: TButton;

Panel3: TPanel;

Label7: TLabel;

Panel4: TPanel;

Label8: TLabel;

StringGrid2: TStringGrid;

Panel5: TPanel;

Label9: TLabel;

StringGrid3: TStringGrid;

Panel6: TPanel;

Label10: TLabel;

StringGrid4: TStringGrid;

```

Panel7: TPanel;
Label11: TLabel;
StringGrid5: TStringGrid;
Panel8: TPanel;
Label12: TLabel;
StringGrid6: TStringGrid;
Panel9: TPanel;
Label13: TLabel;
StringGrid7: TStringGrid;
Panel10: TPanel;
Label14: TLabel;
StringGrid8: TStringGrid;
Button2: TButton;
Button3: TButton;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
StringGrid1: TStringGrid;
Button4: TButton;
Button5: TButton;
Label33: TLabel;
Label34: TLabel;
Image2: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button3Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Button2Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure FormShow(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frm_trans: Tfrm_trans;

implementation

uses Unit1, Unit16;

{$R *.dfm}

procedure clearing(Sender: TObject);
var i: integer;

```

```
begin
with frm_trans do
begin
Edit1.Clear;
Edit2.Clear;
Edit3.Clear;
Edit4.Clear;
Edit5.Clear;
edit6.Clear;
label7.Caption:="";
label8.Caption:="";
label9.Caption:="";
label10.Caption:="";
label11.Caption:="";
label12.Caption:="";
label13.Caption:="";
label14.Caption:="";
```

```
label16.Caption:="";
label17.Caption:="";
label18.Caption:="";
label19.Caption:="";
label20.Caption:="";
label21.Caption:="";
label22.Caption:="";
label23.Caption:="";
```

```
label15.Caption:="";
```

```
i:=0;
for i:=0 to 6 do
begin
stringgrid1.Cells[0,i]:="";
stringgrid1.Cells[1,i]:="";
end;
```

```
i:=0;
for i:=0 to 6 do
begin
stringgrid2.Cells[0,i]:="";
stringgrid2.Cells[1,i]:="";
end;
```

```
i:=0;
```

```
for i:=0 to 6 do
begin
  stringgrid3.Cells[0,i]:="";
  stringgrid3.Cells[1,i]:="";
end;
```

```
i:=0;
for i:=0 to 6 do
begin
  stringgrid4.Cells[0,i]:="";
  stringgrid4.Cells[1,i]:="";
end;
```

```
i:=0;
for i:=0 to 6 do
begin
  stringgrid5.Cells[0,i]:="";
  stringgrid5.Cells[1,i]:="";
end;
```

```
i:=0;
for i:=0 to 6 do
begin
  stringgrid6.Cells[0,i]:="";
  stringgrid6.Cells[1,i]:="";
end;
```

```
i:=0;
for i:=0 to 6 do
begin
  stringgrid7.Cells[0,i]:="";
  stringgrid7.Cells[1,i]:="";
end;
```

```
i:=0;
for i:=0 to 6 do
begin
  stringgrid8.Cells[0,i]:="";
  stringgrid8.Cells[1,i]:="";
end;
```

```
i:=0;
for i:=0 to 6 do
begin

end;
```

```
i:=0;
for i:=0 to 6 do
begin
```



```

end;

i:=0;
for i:=0 to 6 do
begin
end;

i:=0;
for i:=0 to 6 do
begin
end;
edit1.SetFocus;
end;
end;

procedure Tfrm_trans.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    frm_main.show;
end;

procedure Tfrm_trans.Button3Click(Sender: TObject);
begin
    frm_trans.Close;

end;

procedure Tfrm_trans.Button1Click(Sender: TObject);
var i,count,sumcredit,sumgencredit,varcorId:integer;
var sum,sumgrade,sumgengrade:real;
var realdate:Tdate;
var regyear,regmonth,mpart,ypart,season,t:string;

begin
    i:=0;
    frm_main.DataSource7.DataSet:=frm_main.ADOQuery1;

    if (edit1.text<>null) then
    begin
        with frm_main.ADOQuery1 do
        begin
            SQL.Clear;
            sql.add('SELECT * FROM Student WHERE StudentID="'+edit1.text+'"');
            open;
            end;

        if frm_main.DataSource7.DataSet.RecordCount=0 then
        begin
            messagedlg('Student is not found !!',mtwarning,[mbok],0);
            clearing(sender);
        end;
    end;

```

```

    exit;
end
else
begin
    edit2.Text:=frm_main.DataSource7.DataSet.FieldValues['StudentId'];
    edit3.Text:=frm_main.DataSource7.DataSet.FieldValues['name'];
    edit4.Text:=frm_main.DataSource7.DataSet.FieldValues['Surname'];
    edit5.Text:=datetostr(frm_main.DataSource7.DataSet.FieldValues['Birthdate']);
    edit6.Text:=frm_main.DataSource7.DataSet.FieldValues['DepartmentId'];

    // to write semester names

    with frm_main.ADOQuery1 do
    begin
        SQL.Clear;
        sql.add('SELECT t.courseregdate,t.grade,c.coursecode,c.credit FROM transcript as t
inner join course as c on t.courseId=c.courseId WHERE t.StudentID="'+edit1.text+'" and
t.dropdate is null and t.grade is not null order by t.transId');
        open;
    end;

    frm_main.DataSource7.DataSet.First;
    label45.Caption:=inttostr(frm_main.DataSource7.DataSet.RecordCount);

    realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
    shortdateformat:=('yyyy');
    regyear:=datetostr(realdate);
    ypart:=regyear;
    shortdateformat:=('mm');
    regmonth:=datetostr(realdate);
    mpart:=regmonth;
    i:=0;
    sumcredit:=0;
    sumgrade:=0;

    while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
    begin

        if
        ((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
            season:='Fall'
        else
            season:='Spring';

        stringgrid1.Cells[0,i]:=frm_main.DataSource7.DataSet.FieldValues['Coursecode'];

```

```

stringgrid1.Cells[1,i]:=frm_main.DataSource7.DataSet.FieldValues['grade'];
i:=i+1;
shortdateformat:=('yyyy');
label7.Caption:=REGYEAR+' - '+season+' Term';
shortdateformat:=('dd/mm/yyyy');

sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

end;
if ((sumcredit>0)) then
  begin
    str((sumgrade/sumcredit):3:2,t);
    label16.Caption:=t;
  end;
// end of first stringgrid
// start of second stringgrid

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

```

```

mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin

    if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
        season:='Fall'
    else
        season:='Spring';

    stringgrid2.Cells[0,i]:=frm_main.DataSource7.DataSet.FieldValues['Coursecode'];
    stringgrid2.Cells[1,i]:=frm_main.DataSource7.DataSet.FieldValues['grade'];
    i:=i+1;
    shortdateformat:='(yyyy)';
    label8.Caption:=REGYEAR+' - '+season+' Term';
    shortdateformat:='(dd/mm/yyyy)';

    sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

    if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
    if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
    if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
    if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
    if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
    if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
    if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
    if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
    if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

    frm_main.DataSource7.DataSet.Next;
    realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
    shortdateformat:='(yyyy)';
    regyear:=datetostr(realdate);

```



```

shortdateformat:=('mm');
regmonth:=datetostr(realdate);

end;
if ((sumcredit>0)) then
begin
str((sumgrade/sumcredit):3:2,t);
label17.Caption:=t;
end;

// end of second stringgrid
// start of 3. stringgrid

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

while not(((abs(strtoint(regmonth)-
strtoint(mpart))))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin

if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
season:='Fall'
else
season:='Spring';

stringgrid3.Cells[0,i]:=frm_main.DataSource7.DataSet.FieldValues['Coursecode'];
stringgrid3.Cells[1,i]:=frm_main.DataSource7.DataSet.FieldValues['grade'];
i:=i+1;
shortdateformat:=('yyyy');
label9.Caption:=REGYEAR+ ' - '+season+' Term';
shortdateformat:=('dd/mm/yyyy');

sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then

```



```

sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

```

```

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

```

```

end;
if ((sumcredit>0)) then
begin
str((sumgrade/sumcredit):3:2,t);
label18.Caption:=t;
end;

```

// end of 3. stringgrid

// start of 4 stringgrid

```

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

```

```

while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin

```

```

if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
    season:='Fall'
else
    season:='Spring';

```

```

stringgrid4.Cells[0,i]:=frm_main.DataSource7.DataSet.FieldValues['Coursecode'];
stringgrid4.Cells[1,i]:=frm_main.DataSource7.DataSet.FieldValues['grade'];
i:=i+1;
shortdateformat:='yyyy';
label10.Caption:=REGYEAR+' - '+season+' Term';
shortdateformat:='dd/mm/yyyy';

```

```

sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

```

```

while 1
if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

```

```

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:='yyyy';
regyear:=datetostr(realdate);
shortdateformat:='mm';
regmonth:=datetostr(realdate);

```

```

end;
if ((sumcredit>0)) then
begin
    str((sumgrade/sumcredit):3:2,t);
    label19.Caption:=t;
end;

```

```
// end of 4. stringgrid
```

```
// start of 5. stringgrid
```

```
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;
```

```
while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin
```

```
if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
    season:='Fall'
else
    season:='Spring';
```

```
stringgrid5.Cells[0,i]:=frm_main.DataSource7.DataSet.FieldValues['Coursecode'];
stringgrid5.Cells[1,i]:=frm_main.DataSource7.DataSet.FieldValues['grade'];
i:=i+1;
shortdateformat:=('yyyy');
label11.Caption:=REGYEAR+' - '+season+' Term';
shortdateformat:=('dd/mm/yyyy');
```

```
sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];
```

```
if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
```



```

sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

```

```

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

```

```

end;
if ((sumcredit>0)) then
begin
str((sumgrade/sumcredit):3:2,t);
label20.Caption:=t;
end;

```

// end of 5. stringgrid

// start of 6. stringgrid

```

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

```

```

while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin

```

```

if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
season:='Fall'
else
season:='Spring';

```

```

stringgrid6.Cells[0,i]:=frm_main.DataSource7.DataSet.FieldValues['Coursecode'];
stringgrid6.Cells[1,i]:=frm_main.DataSource7.DataSet.FieldValues['grade'];
i:=i+1;
shortdateformat:=('yyyy');
label12.Caption:=REGYEAR+' - '+season+' Term';

```

```

shortdateformat:=('dd/mm/yyyy');

sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
  sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

end;
if ((sumcredit>0)) then
  begin
    str((sumgrade/sumcredit):3:2,t);
    label21.Caption:=t;
  end;

// end of 6. stringgrid

// start of 7. stringgrid

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;

```



```

sumcredit:=0;
sumgrade:=0;

while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin
if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
season:='Fall'
else
season:='Spring';

stringgrid7.Cells[0,i]:=frm_main.DataSource7.DataSet.FieldValues['Coursecode'];
stringgrid7.Cells[1,i]:=frm_main.DataSource7.DataSet.FieldValues['grade'];
i:=i+1;
shortdateformat:='yyyy';
label13.Caption:=REGYEAR+' - '+season+' Term';
shortdateformat:='(dd/mm/yyyy)';

sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:='yyyy';
regyear:=datetostr(realdate);
shortdateformat:='mm';
regmonth:=datetostr(realdate);

end;

```

```

if ((sumcredit>0)) then
begin
  str((sumgrade/sumcredit):3:2,t);
  label22.Caption:=t;
end;

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin

  if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
    season:='Fall'
  else
    season:='Spring';

  stringgrid8.Cells[0,i]:=frm_main.DataSource7.DataSet.FieldValues['Coursecode'];
  stringgrid8.Cells[1,i]:=frm_main.DataSource7.DataSet.FieldValues['grade'];
  i:=i+1;
  shortdateformat:=('yyyy');
  label14.Caption:=REGYEAR+ ' - '+season+' Term';
  shortdateformat:=('dd/mm/yyyy');

  sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

  if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
  if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
  if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
  if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
  if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
  if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
  if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then

```

```

sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

```

```

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

```

```

end;
if ((sumcredit>0)) then
begin
str((sumgrade/sumcredit):3:2,t);
label23.Caption:=t;
end;

```

// end of 8. stringgrid

// start of 9. stringgrid

```

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

```

```

while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin

```

```

if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
season:='Fall'
else
season:='Spring';

```



```

i:=i+1;
shortdateformat:=('yyyy');
shortdateformat:=('dd/mm/yyyy');

sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

```

```

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

```

```

end;
if ((sumcredit>0)) then
    begin
        str((sumgrade/sumcredit):3:2,t);
    end;

```

// end of 9. stringgrid

// start of 10. stringgrid

```

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

```

```

mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin

    if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
        season:='Fall'
    else
        season:='Spring';

    i:=i+1;
    shortdateformat:=('yyyy');

    shortdateformat:=('dd/mm/yyyy');

    sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

    if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
    if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
    if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
    if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
    if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
    if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
    if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
    if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
    if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
        sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

    frm_main.DataSource7.DataSet.Next;
    realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
    shortdateformat:=('yyyy');
    regyear:=datetostr(realdate);
    shortdateformat:=('mm');

```



```

regmonth:=datetostr(realdate);

end;
if ((sumcredit>0)) then
begin
str((sumgrade/sumcredit):3:2,t);

end;

// end of 10. stringgrid

// start of 11. stringgrid

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

while not(((abs(strtoint(regmonth)-
strtoint(mpart)))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
begin

if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
season:='Fall'
else
season:='Spring';

i:=i+1;
shortdateformat:=('yyyy');

shortdateformat:=('dd/mm/yyyy');

sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;

```

```

if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

```

```

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

```

```

end;
if ((sumcredit>0)) then
    begin
        str((sumgrade/sumcredit):3:2,t);
    end;

```

// end of 11. stringgrid

// start of 12. stringgrid

```

realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
ypart:=regyear;
shortdateformat:=('mm');
regmonth:=datetostr(realdate);
mpart:=regmonth;
i:=0;
sumcredit:=0;
sumgrade:=0;

```

```

while not(((abs(strtoint(regmonth)-
strtoint(mpart))))>2)or(regyear<>ypart)or(frm_main.DataSource7.DataSet.eof))do
    begin

```

```

if
((regmonth='01')or(regmonth='12')or(regmonth='11')or(regmonth='10')or(regmonth='09')or(re
gmonth='08')) then
    season:='Fall'
else
    season:='Spring';

```

```

i:=i+1;
shortdateformat:=('yyyy');
shortdateformat:=('dd/mm/yyyy');

```

```

sumcredit:=sumcredit+frm_main.DataSource7.dataset.FieldValues['credit'];

```

```

if frm_main.DataSource7.dataset.FieldValues['grade']='AA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*4;
if frm_main.DataSource7.dataset.FieldValues['grade']='BA' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='BB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*3;
if frm_main.DataSource7.dataset.FieldValues['grade']='CB' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='CC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*2;
if frm_main.DataSource7.dataset.FieldValues['grade']='DC' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='DD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*1;
if frm_main.DataSource7.dataset.FieldValues['grade']='FD' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0.5;
if frm_main.DataSource7.dataset.FieldValues['grade']='FF' then
    sumgrade:=sumgrade+frm_main.DataSource7.dataset.FieldValues['credit']*0;

```

```

frm_main.DataSource7.DataSet.Next;
realdate:=frm_main.DataSource7.DataSet.FieldValues['courseregdate'];
shortdateformat:=('yyyy');
regyear:=datetostr(realdate);
shortdateformat:=('mm');
regmonth:=datetostr(realdate);

```

```

end;
if ((sumcredit>0)) then
begin
    str((sumgrade/sumcredit):3:2,t);
end;

```

```

// end of 12. stringgrid

```

```

// start of cgpa
sumgencredit:=0;
sumgengrade:=0;

with frm_main.ADOQuery4 do
begin
sql.Clear;
sql.add('Select t.GradeDate,t.studentId,t.grade,t.courseId,c.credit from transcript as t
inner join course as c on t.courseId=c.courseId where t.studentId="'+edit1.Text+'" and t.grade
is not null');
open;
end;

frm_main.DataSource10.DataSet.First;

while not(frm_main.DataSource10.DataSet.Eof) do
begin
with frm_main.DataSource10 do
begin
varcorId:=dataset.fieldvalues['courseId'];
label46.caption:=inttostr(varcorId);
with frm_main.ADOQuery6 do
begin
sql.Clear;
sql.add('Select t.GradeDate,t.studentId,t.grade,t.courseID,c.credit from transcript as t
inner join course as c on t.courseID=c.courseID where t.studentId="'+edit1.Text+'" and
t.courseId='+label46.caption+' order by gradedate desc');
open;
end;

frm_main.DataSource12.DataSet.First;

if
dataset.FieldValues['gradedate']=frm_main.DataSource12.DataSet.FieldValues['gradedate']
then
begin
sumgencredit:=sumgencredit+dataset.FieldValues['credit'];

if dataset.FieldValues['grade']='AA' then
sumgengrade:=sumgengrade+dataset.FieldValues['credit']*4;
if dataset.FieldValues['grade']='BA' then
sumgengrade:=sumgengrade+dataset.FieldValues['credit']*3.5;
if dataset.FieldValues['grade']='BB' then
sumgengrade:=sumgengrade+dataset.FieldValues['credit']*3;
if dataset.FieldValues['grade']='CB' then
sumgengrade:=sumgengrade+dataset.FieldValues['credit']*2.5;
if dataset.FieldValues['grade']='CC' then
sumgengrade:=sumgengrade+dataset.FieldValues['credit']*2;

```



```

    if dataset.FieldValues['grade']='DC' then
        sumgengrade:=sumgengrade+dataset.FieldValues['credit']*1.5;
    if dataset.FieldValues['grade']='DD' then
        sumgengrade:=sumgengrade+dataset.FieldValues['credit']*1;
    if dataset.FieldValues['grade']='FD' then
        sumgengrade:=sumgengrade+dataset.FieldValues['credit']*0.5;
    if dataset.FieldValues['grade']='FF' then
        sumgengrade:=sumgengrade+dataset.FieldValues['credit']*0;
    dataset.Next;
end
else
begin
    dataset.Next;
end;    // end of if

```

```

end;    //end of with

```

```

end;    //end of while

```

```

if ((sumgencredit>0) and (sumgengrade>0)) then
begin
    str((sumgengrade/sumgencredit):3:2,t);
    label15.Caption:=t;
end;

```

```

shortdateformat:=('dd/mm/yyyy');

```

```

// end of cgpa....

```

```

end;
end
else
begin
    messagedlg('Please Enter student number',mtwarning,[mbok],0);
    clearing(sender);
    exit;
end;

```

```

{*****}

```

```

{

```

```

}

```

```

end;

```

```

procedure Tfrm_trans.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    if not((key in ['0'..'9'])) then
        key:=#0;

```



```

end;

procedure Tfrm_trans.Button2Click(Sender: TObject);
var i,j:integer;
begin
    frm_trans.Print;
    //clearing(sender);

end;

procedure Tfrm_trans.Button5Click(Sender: TObject);
begin
    frm_main.show;
    frm_trans.hide;
end;

procedure Tfrm_trans.Button4Click(Sender: TObject);
begin
    clearing(sender);
    edit1.SetFocus;
end;

procedure Tfrm_trans.FormShow(Sender: TObject);
begin
    clearing(sender);

end;

end.

```

unit Unit15;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, DBCtrls, Grids, DBGrids, DB, ADODB, jpeg, ExtCtrls;

type

```

Tfrm_semend = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;

```

```

Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Edit1: TEdit;
Button1: TButton;
ComboBox1: TComboBox;
ComboBox2: TComboBox;
ComboBox3: TComboBox;
ComboBox4: TComboBox;
ComboBox5: TComboBox;
ComboBox6: TComboBox;
ComboBox7: TComboBox;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Button5: TButton;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
ADOQuery1: TADOQuery;
Button6: TButton;
Label20: TLabel;
Label21: TLabel;
Image2: TImage;
procedure FormShow(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frm_semend: Tfrm_semend;

implementation

uses Unit1,unit26, Unit22, Unit28;

```

```
{SR *.dfm}
```

```
procedure Tfrm_semend.FormShow(Sender: TObject);  
begin
```

```
    edit1.SetFocus;  
    edit1.Clear;  
    edit2.Clear;  
    edit3.Clear;  
    edit4.Clear;  
    edit1.ReadOnly:=false;
```

```
    label12.Caption:="";
```

```
    label2.Visible:=false;  
    label2.Caption:="";  
    label3.Visible:=false;  
    label3.Caption:="";  
    label4.Visible:=false;  
    label4.Caption:="";  
    label5.Visible:=false;  
    label5.Caption:="";  
    label6.Visible:=false;  
    label6.Caption:="";  
    label7.Visible:=false;  
    label7.Caption:="";  
    label8.Visible:=false;  
    label8.Caption:="";
```

```
    combobox1.Text:="";  
    combobox1.Visible:=false;  
    combobox2.Text:="";  
    combobox2.Visible:=false;  
    combobox3.Text:="";  
    combobox3.Visible:=false;  
    combobox4.Text:="";  
    combobox4.Visible:=false;  
    combobox5.Text:="";  
    combobox5.Visible:=false;  
    combobox6.Text:="";  
    combobox6.Visible:=false;  
    combobox7.Text:="";  
    combobox7.Visible:=false;
```

```
end;
```

```
procedure Tfrm_semend.Button1Click(Sender: TObject);  
var i:integer;  
var ypart:integer;  
var mpart:string;
```

```

var ShortMonthNames:string;
begin
if edit1.text<>" then
begin
ShortDateFormat := 'm/dd/yyyy';
if length(edit1.Text)=8 then
begin
with frm_main.ADOQuery1 do
begin
sql.clear;
sql.Add('Select * from student where StudentId="'+edit1.text+'"');
open;
end;
if frm_main.DataSource7.DataSet.RecordCount<>0 then
begin

edit1.ReadOnly:=true;
edit2.Text:= frm_main.DataSource7.DataSet.FieldValues['name'];
edit3.Text:= frm_main.DataSource7.DataSet.FieldValues['surname'];
edit4.Text:= frm_main.DataSource7.DataSet.FieldValues['departmentID'];

// term caption
shortdateformat:=('mm');
ShortMonthNames:=datetostr(date);
// label12.Caption:=ShortMonthNames;
if
((ShortMonthNames='01')or(ShortMonthNames='12')or(ShortMonthNames='11')or(ShortMonthNames='10')or(ShortMonthNames='09')or(ShortMonthNames='08')) then
mpart:='Fall'
else
mpart:='Spring';

shortdateformat:=('yyyy');
// ypart:=date;
label12.Caption:=datetostr(date)+' - '+mpart+' Term registration ';

shortdateformat:=('dd/mm/yyyy');

with frm_main.ADOQuery2 do //Find if already registered
begin
sql.Clear;
sql.Add('SELECT c.coursecode,c.courseId,t.grade,t.gradedate,t.updateuser FROM
transcript as t inner join course as c on t.courseId=c.courseId where
t.StudentId="'+edit1.Text+'" and t.grade is null and t.dropdate is null');
open;
end;

if frm_main.DataSource8.DataSet.RecordCount=0 then
begin

```

```
messagedlg('This Student Has Not Been Registered Yet, Use Registration Menu To  
Make Course Registration ...!',mtwarning,[mbok],0);
```

```
Edit1.Clear;
```

```
Edit2.Clear;
```

```
Edit3.Clear;
```

```
Edit4.Clear;
```

```
edit1.ReadOnly:=False;
```

```
exit;
```

```
end
```

```
else
```

```
begin
```

```
with frm_main.DataSource8 do
```

```
begin
```

```
dataset.First;
```

```
i:=1;
```

```
if not(dataset.Eof) then
```

```
begin
```

```
label2.Visible:=true;
```

```
combobox1.Visible:=true;
```

```
label2.Caption:=dataset.FieldValues['Coursecode'];
```

```
label13.Caption:=dataset.FieldValues['CourseId'];
```

```
dataset.Next;
```

```
end;
```

```
if not(dataset.Eof) then
```

```
begin
```

```
label3.Visible:=true;
```

```
combobox2.Visible:=true;
```

```
label3.Caption:=dataset.FieldValues['Coursecode'];
```

```
label14.Caption:=dataset.FieldValues['CourseId'];
```

```
dataset.Next;
```

```
end;
```

```
if not(dataset.Eof) then
```

```
begin
```

```
label4.Visible:=true;
```

```
combobox3.Visible:=true;
```

```
label4.Caption:=dataset.FieldValues['Coursecode'];
```

```
label15.Caption:=dataset.FieldValues['CourseId'];
```

```
dataset.Next;
```

```
end;
```

```
if not(dataset.Eof) then
```

```
begin
```

```
label5.Visible:=true;
```

```
combobox4.Visible:=true;
```

```
label5.Caption:=dataset.FieldValues['Coursecode'];
```

```
label16.Caption:=dataset.FieldValues['CourseId'];
```

```
dataset.Next;
```

```
end;
```



```

if not(dataset.Eof) then
begin
label6.Visible:=true;
combobox5.Visible:=true;
label6.Caption:=dataset.FieldValues['Coursecode'];
label17.Caption:=dataset.FieldValues['CourseId'];
dataset.Next;
end;

if not(dataset.Eof) then
begin
label7.Visible:=true;
combobox6.Visible:=true;
label7.Caption:=dataset.FieldValues['Coursecode'];
label18.Caption:=dataset.FieldValues['CourseId'];
dataset.Next;
end;

if not(dataset.Eof) then
begin
label8.Visible:=true;
combobox7.Visible:=true;
label8.Caption:=dataset.FieldValues['Coursecode'];
label19.Caption:=dataset.FieldValues['CourseId'];
end;
end;
end;

with frm_main.ADOQuery1 do
begin
sql.clear;
sql.Add('Select * from student where StudentId="'+edit1.text+'"');
open;
end;

end
else
begin
messagedlg('Student Number Is Not Found !',mtwarning,[mbok],0);
edit1.SetFocus;
end;
end
else
begin
messagedlg('Student Number Must Be 8 Digits !',mtwarning,[mbok],0);
edit1.SetFocus;
end;
end
else

```

```

begin
    messagedlg('Please Enter Number For Student To Update ! ',mtwarning,[mbok],0);
    edit1.Clear;
    edit1.SetFocus;
end;
end;

```

```

procedure Tfrm_semend.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    frm_beginsemester.show;
end;

```

```

procedure Tfrm_semend.Button4Click(Sender: TObject);
begin
    frm_semend.Close;
end;

```

```

procedure Tfrm_semend.Button5Click(Sender: TObject);
begin
    frm_main.Show;
end;

```

```

procedure Tfrm_semend.Button3Click(Sender: TObject);
begin
    frm_semend.Show;
end;

```

```

procedure Tfrm_semend.Button2Click(Sender: TObject);
begin

```

```

{ with frm_main.ADOQuery2 do      //Find if already registered
begin
    sql.Clear;
    sql.Add('SELECT t.grade,t.studentId,t.gradedate,t.updateuser FROM transcript as t inner
join course as c on t.courseId=c.courseId where t.StudentId="'+edit1.Text+'" and
c.coursecode="'+label1.caption+'" and t.grade is null and t.dropdate is null');
    open;
end;
}

```

```

with frm_main.ADOQuery2 do      //Find if already registered
begin
    sql.Clear;
    sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label2.Caption+'" ) order by courseregdate desc');
    open;
end;

```

```

with frm_main.DataSource8 do
begin
// dataset.First;
label19.caption:=inttostr(DataSet.RecordCount);
if not((DataSet.RecordCount=0)and((combobox1.text=")or(combobox1.Text=null))) then
begin
dataset.Edit;
DataSet.fieldvalues['grade']:=combobox1.Text;
DataSet.FieldValues['gradedate']:=date;
DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label13.Caption);
DataSet.post;

end;
end;

with frm_main.ADOQuery2 do      //Find if already registered
begin
sql.Clear;
sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label3.Caption+'" ) order by courseregdate desc');
open;
end;

with frm_main.DataSource8 do
begin
// dataset.First;

if not((DataSet.RecordCount=0)and((combobox2.text=")or(combobox2.Text=null))) then
begin
dataset.Edit;
DataSet.FieldValues['grade']:=combobox2.Text;
DataSet.FieldValues['gradedate']:=date;
DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label14.Caption);
DataSet.post;
//dataset.Next;
end;
end;

with frm_main.ADOQuery2 do      //Find if already registered
begin
sql.Clear;
sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label4.Caption+'" ) order by courseregdate desc');

```

```

open;
end;

with frm_main.DataSource8 do
begin
// dataset.First;

if not((DataSet.RecordCount=0)and((combobox3.text=")or(combobox3.Text=null))) then
begin
dataset.Edit;
DataSet.FieldValues['grade']:=combobox4.Text;
DataSet.FieldValues['gradedate']:=date;
DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label15.Caption);
DataSet.post;
// dataset.Next;
end;
end;

with frm_main.ADOQuery2 do      //Find if already registered
begin
sql.Clear;
sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+" and courseId=(SELECT courseId from course where
coursecode='"+label5.Caption+"') order by courseregdate desc');
open;
end;

with frm_main.DataSource8 do
begin
// dataset.First;

if not((DataSet.RecordCount=0)and((combobox4.text=")or(combobox4.Text=null))) then
begin
dataset.Edit;
DataSet.FieldValues['grade']:=combobox4.Text;
DataSet.FieldValues['gradedate']:=date;
DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label16.Caption);
DataSet.post;
// dataset.Next;
end;
end;

with frm_main.ADOQuery2 do      //Find if already registered
begin
sql.Clear;

```



```

sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label6.Caption+'" ) order by courseregdate desc');
open;
end;

```

```

// frm_main.ADOQuery2.DataSource.DataSet.FieldValues

```

```

with frm_main.DataSource8 do
begin
// dataset.First;

if not((DataSet.RecordCount=0)and(((combobox5.text=")or(combobox5.Text=null))) then
begin
dataset.Edit;
DataSet.FieldValues['grade']:=combobox5.Text;
DataSet.FieldValues['gradedate']:=date;
DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label17.Caption);
DataSet.post;
// dataset.Next;
end;
end;

```

```

with frm_main.ADOQuery2 do      //Find if already registered
begin
sql.Clear;
sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label7.Caption+'" ) order by courseregdate desc');
open;
end;

```

```

with frm_main.DataSource8 do
begin
// dataset.First;
// label19.caption:=inttostr(DataSet.RecordCount);
if not((DataSet.RecordCount=0)and(((combobox6.text=")or(combobox6.Text=null))) then
begin
dataset.Edit;
DataSet.fieldvalues['grade']:=combobox6.Text;
DataSet.FieldValues['gradedate']:=date;
DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label18.Caption);
DataSet.post;

end;

```



```

end;

with frm_main.ADOQuery2 do      //Find if already registered
begin
    sql.Clear;
    sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label8.Caption+'" ) order by courseregdate desc');
    open;
end;

with frm_main.DataSource8 do
begin
// dataset.First;
// label19.caption:=inttostr(DataSet.RecordCount);
if not((DataSet.RecordCount=0)and((combobox7.text=")or(combobox7.Text=null))) then
begin
    dataset.Edit;
    DataSet.fieldvalues['grade']:=combobox7.Text;
    DataSet.FieldValues['gradedate']:=date;
    DataSet.FieldValues['updateuser']:=username;
    DataSet.FieldValues['StudentId']:=edit1.Text;
    DataSet.FieldValues['courseId']:=strtoint(label19.Caption);
    DataSet.post;

end;
end;

{with frm_main.ADOQuery2 do      //Find if already registered
begin
    sql.Clear;
    sql.Add('UPDATE transcript (');//SELECT top 1
grade,studentId,gradedate,updateuser,courseId FROM transcript where
studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label6.Caption+'" ) order by courseregdate desc');
    open;
end;
{ with frm_main.ADOQuery2 do      //Find if already registered
begin
    sql.Clear;
    sql.Add('SELECT t.grade,t.gradedate,t.updateuser,t.studentId FROM transcript as t inner
join course as c on t.courseId=c.courseId where t.StudentId="'+edit1.Text+'" and
c.coursecode="'+label1.caption+'" and t.grade is null and t.dropdate is null');
    open;
end;

{ with frm_main.DataSource8 do
begin
    dataset.Edit;

```

```

dataset.First;
if not((dataset.Eof)and(combobox6.text<>"")and(combobox6.Visible)) then
begin
  dataset.FieldValues['grade']:=combobox6.Text;
  dataset.FieldValues['gradedate']:=date;
  dataset.FieldValues['updateuser']:=username;
  dataset.FieldValues['StudentId']:=edit1.Text;
  dataset.post;
  // dataset.Next;
end;
end;

with frm_main.ADOQuery2 do      //Find if already registered
begin
  sql.Clear;
  sql.Add('SELECT t.grade,t.gradedate,t.updateuser,t.StudentId FROM transcript as t inner
join course as c on t.courseId=c.courseId where t.StudentId="'+edit1.Text+'" and
c.coursecode="'+label1.caption+'" and t.grade is null and t.dropdate is null');
  open;
end;

with frm_main.DataSource8 do
begin
  dataset.Edit;
  dataset.First;
  if not((dataset.Eof)and(combobox7.text<>"")and(combobox7.Visible)) then
  begin
    dataset.FieldValues['grade']:=combobox7.Text;
    dataset.FieldValues['gradedate']:=date;
    dataset.FieldValues['updateuser']:=username;
    dataset.FieldValues['StudentId']:=edit1.Text;
    dataset.post;
  //  dataset.Next;
  end;
end;
}
end;

end.

```

unit Unit15;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, DBCtrls, Grids, DBGrids, DB, ADODB, jpeg, ExtCtrls;

type

```

Tfrm_semend = class(TForm)
  Image1: TImage;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label11: TLabel;
  Label12: TLabel;
  Label13: TLabel;
  Label14: TLabel;
  Label15: TLabel;
  Label16: TLabel;
  Label17: TLabel;
  Label18: TLabel;
  Label19: TLabel;
  Edit1: TEdit;
  Button1: TButton;
  ComboBox1: TComboBox;
  ComboBox2: TComboBox;
  ComboBox3: TComboBox;
  ComboBox4: TComboBox;
  ComboBox5: TComboBox;
  ComboBox6: TComboBox;
  ComboBox7: TComboBox;
  Button2: TButton;
  Button3: TButton;
  Button4: TButton;
  Button5: TButton;
  Edit2: TEdit;
  Edit3: TEdit;
  Edit4: TEdit;
  ADOQuery1: TADOQuery;
  Button6: TButton;
  Label20: TLabel;
  Label21: TLabel;
  Image2: TImage;
  procedure FormShow(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure Button4Click(Sender: TObject);
  procedure Button5Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
private

```

```

    { Private declarations }
public
    { Public declarations }
end;

var
    frm_send: Tfrm_send;

implementation

uses Unit1, unit26, Unit22, Unit28;

{$R *.dfm}

procedure Tfrm_send.FormShow(Sender: TObject);
begin
    edit1.SetFocus;
    edit1.Clear;
    edit2.Clear;
    edit3.Clear;
    edit4.Clear;
    edit1.ReadOnly:=false;

    label12.Caption:="";

    label2.Visible:=false;
    label2.Caption:="";
    label3.Visible:=false;
    label3.Caption:="";
    label4.Visible:=false;
    label4.Caption:="";
    label5.Visible:=false;
    label5.Caption:="";
    label6.Visible:=false;
    label6.Caption:="";
    label7.Visible:=false;
    label7.Caption:="";
    label8.Visible:=false;
    label8.Caption:="";

    combobox1.Text:="";
    combobox1.Visible:=false;
    combobox2.Text:="";
    combobox2.Visible:=false;
    combobox3.Text:="";
    combobox3.Visible:=false;
    combobox4.Text:="";
    combobox4.Visible:=false;
    combobox5.Text:="";
    combobox5.Visible:=false;

```

```

combobox6.Text:="";
combobox6.Visible:=false;
combobox7.Text:="";
combobox7.Visible:=false;

```

```

end;

```

```

procedure Tfrm_semend.Button1Click(Sender: TObject);
var i:integer;
var ypart:integer;
var mpart:string;
var ShortMonthNames:string;
begin
  if edit1.text<>" then
  begin
    ShortDateFormat := 'm/dd/yyyy';
    if length(edit1.Text)=8 then
    begin
      with frm_main.ADOQuery1 do
      begin
        sql.clear;
        sql.Add('Select * from student where StudentId="'+edit1.text+'"');
        open;
      end;
      if frm_main.DataSource7.DataSet.RecordCount<>0 then
      begin

        edit1.ReadOnly:=true;
        edit2.Text:= frm_main.DataSource7.DataSet.FieldValues['name'];
        edit3.Text:= frm_main.DataSource7.DataSet.FieldValues['surname'];
        edit4.Text:= frm_main.DataSource7.DataSet.FieldValues['departmentID'];

        // term caption
        shortdateformat:=('mm');
        ShortMonthNames:=datetostr(date);
        // label12.Caption:=ShortMonthNames;
        if
        ((ShortMonthNames='01')or(ShortMonthNames='12')or(ShortMonthNames='11')or(ShortMonthNames='10')or(ShortMonthNames='09')or(ShortMonthNames='08')) then
          mpart:='Fall'
        else
          mpart:='Spring';

        shortdateformat:=('yyyy');
        // ypart:=date;
        label12.Caption:=datetostr(date)+' - '+mpart+' Term registration ';

        shortdateformat:=('dd/mm/yyyy');

        with frm_main.ADOQuery2 do      //Find if already registered

```



```

begin
    sql.Clear;
    sql.Add('SELECT c.coursecode,c.courseId,t.grade,t.gradedate,t.updateuser FROM
transcript as t inner join course as c on t.courseId=c.courseId where
t.StudentId="'+edit1.Text+'" and t.grade is null and t.dropdate is null');
    open;
end;

```

```

if frm_main.DataSource8.DataSet.RecordCount=0 then
begin
    messagedlg('This Student Has Not Been Registered Yet, Use Registration Menu To
Make Course Registration ...!',mtwarning,[mbok],0);
    Edit1.Clear;
    Edit2.Clear;
    Edit3.Clear;
    Edit4.Clear;
    edit1.ReadOnly:=False;
    exit;
end
else
begin
    with frm_main.DataSource8 do
    begin
        dataset.First;
        i:=1;
        if not(dataset.Eof) then
        begin
            label2.Visible:=true;
            combobox1.Visible:=true;
            label2.Caption:=dataset.FieldValues['Coursecode'];
            label13.Caption:=dataset.FieldValues['CourseId'];
            dataset.Next;
        end;
        if not(dataset.Eof) then
        begin
            label3.Visible:=true;
            combobox2.Visible:=true;
            label3.Caption:=dataset.FieldValues['Coursecode'];
            label14.Caption:=dataset.FieldValues['CourseId'];
            dataset.Next;
        end;

```

```

        if not(dataset.Eof) then
        begin
            label4.Visible:=true;
            combobox3.Visible:=true;
            label4.Caption:=dataset.FieldValues['Coursecode'];
            label15.Caption:=dataset.FieldValues['CourseId'];
            dataset.Next;

```

end;

```
if not(dataset.Eof) then
begin
label5.Visible:=true;
combobox4.Visible:=true;
label5.Caption:=dataset.FieldValues['Coursecode'];
label16.Caption:=dataset.FieldValues['CourseId'];
dataset.Next;
end;
```

```
if not(dataset.Eof) then
begin
label6.Visible:=true;
combobox5.Visible:=true;
label6.Caption:=dataset.FieldValues['Coursecode'];
label17.Caption:=dataset.FieldValues['CourseId'];
dataset.Next;
end;
```

```
if not(dataset.Eof) then
begin
label7.Visible:=true;
combobox6.Visible:=true;
label7.Caption:=dataset.FieldValues['Coursecode'];
label18.Caption:=dataset.FieldValues['CourseId'];
dataset.Next;
end;
```

```
if not(dataset.Eof) then
begin
label8.Visible:=true;
combobox7.Visible:=true;
label8.Caption:=dataset.FieldValues['Coursecode'];
label19.Caption:=dataset.FieldValues['CourseId'];
end;
```

end;

end;

```
with frm_main.ADOQuery1 do
begin
sql.clear;
sql.Add('Select * from student where StudentId="'+edit1.text+'"');
open;
end;
```

```

end
else
begin
    messagedlg('Student Number Is Not Found !',mtwarning,[mbok],0);
    edit1.SetFocus;
end;
end
else
begin
    messagedlg('Student Number Must Be 8 Digits !',mtwarning,[mbok],0);
    edit1.SetFocus;
end;
end
else
begin
    messagedlg('Please Enter Number For Student To Update ! ',mtwarning,[mbok],0);
    edit1.Clear;
    edit1.SetFocus;
end;
end;

procedure Tfrm_semend.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    frm_beginsemester.show;
end;

procedure Tfrm_semend.Button4Click(Sender: TObject);
begin
    frm_semend.Close;
end;

procedure Tfrm_semend.Button5Click(Sender: TObject);
begin
    frm_main.Show;
end;

procedure Tfrm_semend.Button3Click(Sender: TObject);
begin
    frm_semend.Show;
end;

procedure Tfrm_semend.Button2Click(Sender: TObject);
begin
    { with frm_main.ADOQuery2 do      //Find if already registered
    begin
        sql.Clear;
        sql.Add('SELECT t.grade,t.studentId,t.gradedate,t.updateuser FROM transcript as t inner
        join course as c on t.courseId=c.courseId where t.StudentId="'+edit1.Text+'" and
        c.coursecode="'+label1.caption+'" and t.grade is null and t.dropdate is null');
    end;
    end;
    
```

```
open;  
end;  
}
```

```
with frm_main.ADOQuery2 do      //Find if already registered  
begin  
    sql.Clear;  
    sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript  
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where  
coursecode="'+label2.Caption+'" ) order by courseregdate desc');  
    open;  
end;
```

```
with frm_main.DataSource8 do  
begin  
    // dataset.First;  
    label19.caption:=inttostr(DataSet.RecordCount);  
    if not((DataSet.RecordCount=0)and((combobox1.text=")or(combobox1.Text=null))) then  
    begin  
        dataset.Edit;  
        DataSet.fieldvalues['grade']:=combobox1.Text;  
        DataSet.FieldValues['gradedate']:=date;  
        DataSet.FieldValues['updateuser']:=username;  
        DataSet.FieldValues['StudentId']:=edit1.Text;  
        DataSet.FieldValues['courseId']:=strtoint(label13.Caption);  
        DataSet.post;  
  
    end;  
end;
```

```
with frm_main.ADOQuery2 do      //Find if already registered  
begin  
    sql.Clear;  
    sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript  
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where  
coursecode="'+label3.Caption+'" ) order by courseregdate desc');  
    open;  
end;
```

```
with frm_main.DataSource8 do  
begin  
    // dataset.First;  
  
    if not((DataSet.RecordCount=0)and((combobox2.text=")or(combobox2.Text=null))) then  
    begin  
        dataset.Edit;  
        DataSet.FieldValues['grade']:=combobox2.Text;  
        DataSet.FieldValues['gradedate']:=date;
```

```

DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label14.Caption);
DataSet.post;
//dataset.Next;
end;
end;

```

```

with frm_main.ADOQuery2 do      //Find if already registered
begin
  sql.Clear;
  sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label4.Caption+'") order by courseregdate desc');
  open;
end;

```

```

with frm_main.DataSource8 do
begin
//  dataset.First;

if not((DataSet.RecordCount=0)and((combobox3.text=")or(combobox3.Text=null))) then
begin
  dataset.Edit;
  DataSet.FieldValues['grade']:=combobox4.Text;
  DataSet.FieldValues['gradedate']:=date;
  DataSet.FieldValues['updateuser']:=username;
  DataSet.FieldValues['StudentId']:=edit1.Text;
  DataSet.FieldValues['courseId']:=strtoint(label15.Caption);
  DataSet.post;
//  dataset.Next;
end;
end;

```

```

with frm_main.ADOQuery2 do      //Find if already registered
begin
  sql.Clear;
  sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label5.Caption+'") order by courseregdate desc');
  open;
end;

```

```

with frm_main.DataSource8 do
begin
//  dataset.First;

if not((DataSet.RecordCount=0)and((combobox4.text=")or(combobox4.Text=null))) then

```



```

begin
    dataset.Edit;
    DataSet.FieldValues['grade']:=combobox4.Text;
    DataSet.FieldValues['gradedate']:=date;
    DataSet.FieldValues['updateuser']:=username;
    DataSet.FieldValues['StudentId']:=edit1.Text;
    DataSet.FieldValues['courseId']:=strtoint(label16.Caption);
    DataSet.post;
//    dataset.Next;
end;
end;

with frm_main.ADOQuery2 do      //Find if already registered
begin
    sql.Clear;
    sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label6.Caption+'" ) order by courseregdate desc');
    open;
end;

//    frm_main.ADOQuery2.DataSource.DataSet.FieldValues
with frm_main.DataSource8 do
begin
//    dataset.First;

if not((DataSet.RecordCount=0)and((combobox5.text=")or(combobox5.Text=null))) then
begin
    dataset.Edit;
    DataSet.FieldValues['grade']:=combobox5.Text;
    DataSet.FieldValues['gradedate']:=date;
    DataSet.FieldValues['updateuser']:=username;
    DataSet.FieldValues['StudentId']:=edit1.Text;
    DataSet.FieldValues['courseId']:=strtoint(label17.Caption);
    DataSet.post;
//    dataset.Next;
end;
end;

with frm_main.ADOQuery2 do      //Find if already registered
begin
    sql.Clear;
    sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label7.Caption+'" ) order by courseregdate desc');
    open;
end;

```

```

with frm_main.DataSource8 do
begin
// dataset.First;
// label19.caption:=inttostr(DataSet.RecordCount);
if not((DataSet.RecordCount=0)and((combobox6.text=")or(combobox6.Text=null))) then
begin
dataset.Edit;
DataSet.fieldvalues['grade']:=combobox6.Text;
DataSet.FieldValues['gradedate']:=date;
DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label18.Caption);
DataSet.post;

end;
end;

```

```

with frm_main.ADOQuery2 do      //Find if already registered
begin
sql.Clear;
sql.Add('SELECT top 1 grade,studentId,gradedate,updateuser,courseId FROM transcript
where studentId="'+edit1.Text+'" and courseId=(SELECT courseId from course where
coursecode="'+label8.Caption+'" ) order by courseregdate desc');
open;
end;

```

```

with frm_main.DataSource8 do
begin
// dataset.First;
// label19.caption:=inttostr(DataSet.RecordCount);
if not((DataSet.RecordCount=0)and((combobox7.text=")or(combobox7.Text=null))) then
begin
dataset.Edit;
DataSet.fieldvalues['grade']:=combobox7.Text;
DataSet.FieldValues['gradedate']:=date;
DataSet.FieldValues['updateuser']:=username;
DataSet.FieldValues['StudentId']:=edit1.Text;
DataSet.FieldValues['courseId']:=strtoint(label19.Caption);
DataSet.post;

end;
end;
{with frm_main.ADOQuery2 do      //Find if already registered
begin
sql.Clear;
sql.Add('UPDATE transcript ( )//SELECT top 1
grade,studentId,gradedate,updateuser,courseId FROM transcript where

```

```
studentId="" + edit1.Text + "" and courseId=(SELECT courseId from course where  
coursecode="" + label6.Caption + "") order by courseregdate desc');
```

```
open;
```

```
end;
```

```
{ with frm_main.ADOQuery2 do          //Find if already registered
```

```
begin
```

```
sql.Clear;
```

```
sql.Add('SELECT t.grade,t.gradedate,t.updateuser,t.studentId FROM transcript as t inner  
join course as c on t.courseId=c.courseId where t.StudentId="" + edit1.Text + "" and  
c.coursecode="" + label1.caption + "" and t.grade is null and t.dropdate is null');
```

```
open;
```

```
end;
```

```
{ with frm_main.DataSource8 do
```

```
begin
```

```
dataset.Edit;
```

```
dataset.First;
```

```
if not((dataset.Eof)and(combobox6.text<>"")and(combobox6.Visible)) then
```

```
begin
```

```
dataset.FieldValues['grade']:=combobox6.Text;
```

```
dataset.FieldValues['gradedate']:=date;
```

```
dataset.FieldValues['updateuser']:=username;
```

```
dataset.FieldValues['StudentId']:=edit1.Text;
```

```
dataset.post;
```

```
// dataset.Next;
```

```
end;
```

```
end;
```

```
with frm_main.ADOQuery2 do          //Find if already registered
```

```
begin
```

```
sql.Clear;
```

```
sql.Add('SELECT t.grade,t.gradedate,t.updateuser,t.StudentId FROM transcript as t inner  
join course as c on t.courseId=c.courseId where t.StudentId="" + edit1.Text + "" and  
c.coursecode="" + label1.caption + "" and t.grade is null and t.dropdate is null');
```

```
open;
```

```
end;
```

```
with frm_main.DataSource8 do
```

```
begin
```

```
dataset.Edit;
```

```
dataset.First;
```

```
if not((dataset.Eof)and(combobox7.text<>"")and(combobox7.Visible)) then
```

```
begin
```

```
dataset.FieldValues['grade']:=combobox7.Text;
```

```
dataset.FieldValues['gradedate']:=date;
```

```
dataset.FieldValues['updateuser']:=username;
```

```
dataset.FieldValues['StudentId']:=edit1.Text;
```

```
dataset.post;
```

```

// dataset.Next;
    end;
end;
}
end;
end.

```

unit Unit16;

interface

uses

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

type

TRep_main = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

Button8: TButton;

Button9: TButton;

Button11: TButton;

Button12: TButton;

Button13: TButton;

Image2: TImage;

Label1: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button9Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Button10Click(Sender: TObject);

procedure Button11Click(Sender: TObject);

procedure Button12Click(Sender: TObject);

procedure Button13Click(Sender: TObject);

procedure Button14Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;


```
var
  Rep_main: TRep_main;
implementation
uses Unit17, Unit18, Unit19, Unit20, Unit21, Unit23, Unit24, Unit25, Unit1,
  Unit14, Unit30, Unit31, Unit32, Unit33;
```

```
{ $R *.dfm }
```

```
procedure TRep_main.Button1Click(Sender: TObject);
begin
```

```
  Rep_all_depts.show;
end;
```

```
procedure TRep_main.Button2Click(Sender: TObject);
begin
  rep_main.close;
end;
```

```
procedure TRep_main.Button3Click(Sender: TObject);
begin
```

```
  rep_all_teacs.show;
```

```
end;
```

```
procedure TRep_main.Button4Click(Sender: TObject);
begin
  Rep_teacs_dept.show;
end;
```

```
procedure TRep_main.Button5Click(Sender: TObject);
begin
  Rep_dept_teacs.show;
end;
```

```
procedure TRep_main.Button6Click(Sender: TObject);
begin
  Rep_all_courses.show;
end;
```

```
procedure TRep_main.Button7Click(Sender: TObject);
begin
  Rep_dept_cor.show;
end;
```

```
procedure TRep_main.Button8Click(Sender: TObject);
begin
  rep_all_stu.show;
end;
```



```

procedure TRep_main.Button9Click(Sender: TObject);
begin
  Rep_dept_stu.show;
end;

procedure TRep_main.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  frm_main.show;
end;

procedure TRep_main.Button10Click(Sender: TObject);
begin
  frm_trans.show;
  rep_main.hide;
end;

procedure TRep_main.Button11Click(Sender: TObject);
begin
  Form30.show;
end;

procedure TRep_main.Button12Click(Sender: TObject);
begin
  Form31.show;
end;

procedure TRep_main.Button13Click(Sender: TObject);
begin
  Form32.Show;
end;

procedure TRep_main.Button14Click(Sender: TObject);
begin
  form33.show;
end;

end.

```

unit Unit17;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, Grids, DBGrids, StdCtrls, jpeg, ExtCtrls;

type

TRep_all_depts = class(TForm)
 Image1: TImage;

```

Label1: TLabel;
DBGrid1: TDBGrid;
Button1: TButton;
Button2: TButton;
ADOQuery1: TADOQuery;
DataSource1: TDataSource;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  Rep_all_depts: TRep_all_depts;

```

```

implementation

```

```

uses Unit16;

```

```

{$R *.dfm}

```

```

procedure TRep_all_depts.FormCreate(Sender: TObject);
begin
  with ADOQuery1 do
    begin
      SQL.Clear;
      sql.add('SELECT * FROM department');
      open;
    end;
end;
procedure TRep_all_depts.Button1Click(Sender: TObject);
begin
  Rep_all_depts.close;
end;
procedure TRep_all_depts.Button2Click(Sender: TObject);
begin
  Rep_all_depts.Print;
end;
procedure TRep_all_depts.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  rep_main.show;
end;
end.

```

unit Unit18;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;

type

```
TRep_all_teacs = class(TForm)
  ADOQuery1: TADOQuery;
  DataSource1: TDataSource;
  Label1: TLabel;
  DBGrid1: TDBGrid;
  Button1: TButton;
  Button2: TButton;
  procedure Button2Click(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

Rep_all_teacs: TRep_all_teacs;

implementation

uses Unit16;

{ \$R *.dfm }

```
procedure TRep_all_teacs.Button2Click(Sender: TObject);
begin
  Rep_all_teacs.close;
end;
```

```
procedure TRep_all_teacs.Button1Click(Sender: TObject);
begin
  Rep_all_teacs.Print;
end;
```

```
procedure TRep_all_teacs.FormCreate(Sender: TObject);
begin
  with ADOQuery1 do
```

```
begin
  SQL.Clear;
  sql.add('SELECT * FROM teacher');
  open;
end;
```

```
end;
```

```
procedure TRep_all_teacs.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  rep_main.show;
end;
```

```
end.
```

unit Unit19;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;

type

```
TRep_teacs_dept = class(TForm)
  ADOQuery1: TADOQuery;
  DataSource1: TDataSource;
  DBGrid1: TDBGrid;
  Button1: TButton;
  Button2: TButton;
  Label1: TLabel;
  procedure FormCreate(Sender: TObject);
  procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Rep_teacs_dept: TRep_teacs_dept;
```

implementation

```
{ $R *.dfm }
```

```
procedure TRep_teacs_dept.FormCreate(Sender: TObject);
begin
  with ADOQuery1 do
```

```

begin
    SQL.Clear;
    sql.add('SELECT tname,departmentId FROM teacher order by departmentId');
    open;
end;

```

end;

```

procedure TRep_teacs_dept.Button2Click(Sender: TObject);
begin
    Rep_teacs_dept.Free;
end;

```

end.

unit Unit20;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, DBCtrls, DB, ADODB, Grids, DBGrids;

type

```

TRep_dept_teacs = class(TForm)
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    Label1: TLabel;
    Button1: TButton;
    Button2: TButton;
    DBGrid1: TDBGrid;
    DataSource2: TDataSource;
    ADOQuery2: TADOQuery;
    ComboBox1: TComboBox;
    Label2: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);

    procedure ComboBox1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

```

var

Rep_dept_teacs: TRep_dept_teacs;

implementation


```
uses unit26;  
{SR *.dfm}
```

```
procedure TRep_dept_teacs.FormCreate(Sender: TObject);  
begin
```

```
    with ADOQuery1 do
```

```
    begin
```

```
        SQL.Clear;
```

```
        sql.add('SELECT departmentId FROM department order by departmentId');
```

```
        open;
```

```
    end;
```

```
    DataSource1.DataSet.First;
```

```
    while not (DataSource1.DataSet.Eof) do
```

```
    begin
```

```
        combobox1.Items.add(DataSource1.DataSet.FieldValues['DepartmentID']);
```

```
        DataSource1.DataSet.Next;
```

```
    end;
```

```
end;
```

```
procedure TRep_dept_teacs.Button1Click(Sender: TObject);
```

```
begin
```

```
    Rep_dept_teacs.Close;
```

```
end;
```

```
procedure TRep_dept_teacs.ComboBox1Click(Sender: TObject);
```

```
begin
```

```
    with ADOQuery2 do
```

```
    begin
```

```
        SQL.Clear;
```

```
        sql.add('SELECT Tname,Tsurname FROM teacher where  
departmentID="'+combobox1.Text+'" order by tname,tsurname');
```

```
        open;
```

```
    end;
```

```
end;
```

```
end.
```

```
unit Unit21;
```

```
interface
```

```
;
```

```
uses
```

```
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
    Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;
```

```

type
  TRep_all_courses = class(TForm)
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

```

var
  Rep_all_courses: TRep_all_courses;

```

```

implementation

```

```

uses Unit16,unit26;

```

```

{$R *.dfm}

```

```

procedure TRep_all_courses.Button2Click(Sender: TObject);
begin
  Rep_all_courses.Close;
end;

```

```

procedure TRep_all_courses.Button1Click(Sender: TObject);
begin
  Rep_all_courses.Print;
end;

```

```

procedure TRep_all_courses.FormCreate(Sender: TObject);
begin
  with ADOQuery1 do
    begin
      SQL.Clear;
      sql.add('SELECT * FROM course order by coursename ');
      open;
    end;
end;

```

```

end;

```

```

procedure TRep_all_courses.FormClose(Sender: TObject;

```

```
    var Action: TCloseAction);  
begin  
    rep_main.show;  
end;  
  
end.
```

unit Unit22;

```
interface
```

```
uses
```

```
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
    Dialogs, StdCtrls, ExtCtrls, strutils, ExtDlgs, jpeg;
```

```
type
```

```
    Tfrm_correg = class(TForm)
```

```
        Image1: TImage;  
        Label1: TLabel;  
        Label2: TLabel;  
        Label3: TLabel;  
        Label4: TLabel;  
        Label7: TLabel;  
        Label8: TLabel;  
        Label9: TLabel;  
        Label10: TLabel;  
        Label11: TLabel;  
        Label12: TLabel;  
        Label13: TLabel;  
        Label14: TLabel;  
        Label15: TLabel;  
        Label16: TLabel;  
        Label17: TLabel;  
        Label18: TLabel;  
        Edit1: TEdit;  
        Button1: TButton;  
        Edit2: TEdit;  
        Edit3: TEdit;  
        Edit4: TEdit;  
        Edit5: TEdit;  
        Edit6: TEdit;  
        Button4: TButton;  
        Button5: TButton;  
        Button6: TButton;  
        Button7: TButton;  
        Button8: TButton;  
        Label19: TLabel;  
        Label20: TLabel;  
        Button2: TButton;  
        ListBox1: TListBox;
```

```

ListBox2: TListBox;
Button3: TButton;
Image2: TImage;
procedure Button6Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frm_correg: Tfrm_correg;

implementation

uses Unit1, ADODB,unit26, Unit28;

{$R *.dfm}

procedure Tfrm_correg.Button6Click(Sender: TObject);
begin
  frm_main.show;
  frm_correg.hide;
end;

procedure Tfrm_correg.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  frm_main.Show;
end;

procedure Tfrm_correg.Button1Click(Sender: TObject);
var sumcredit:integer;
var sumgrade,gpa:real;
var sumgencredit:integer;
var sumgengrade,cgpa:real;
var cid:integer;

var ypart:integer;
var mpart,t:string;
var ShortMonthNames:string;

begin

```

```

if (edit1.text<>null) then
begin
with frm_main.ADOQuery1 do
begin
sql.Clear;
sql.Add('SELECT * FROM student WHERE StudentId="'+edit1.Text+'"');
open;
end;
if not(frm_main.DataSource7.DataSet.eof) then
begin
edit2.Text:=frm_main.DataSource7.DataSet.FieldValues['name'];
edit3.Text:=frm_main.DataSource7.DataSet.FieldValues['surname'];
edit4.Text:=frm_main.DataSource7.DataSet.FieldValues['departmentID'];

edit1.ReadOnly:=true;

shortdateformat:=('mm');
ShortMonthNames:=datetostr(date);
label12.Caption:=ShortMonthNames;
if
((ShortMonthNames='01')or(ShortMonthNames='12')or(ShortMonthNames='11')or(ShortMonthNames='10')or(ShortMonthNames='09')or(ShortMonthNames='08')) then
mpart:='Fall'
else
mpart:='Spring';

shortdateformat:=('yyyy');
// ypart:=date;
label9.Caption:=datetostr(date)+' - '+mpart+' Term registration ';

shortdateformat:=('dd/mm/yyyy');

// To calculate gpa.....

sumcredit:=0;
sumgrade:=0;
gpa:=0;
sumgencredit:=0;
sumgengrade:=0;
cgpa:=0;

with frm_main.ADOQuery4 do
begin
sql.Clear;
sql.add('Select t.GradeDate,t.studentId,t.grade,t.courseID,c.credit from transcript as t
inner join course as c on t.courseID=c.courseID where t.studentId="'+edit1.Text+'" and
t.grade is not null');
open;
end;

```



```

with frm_main.ADOQuery5 do
begin
  sql.Clear;
  sql.add('Select MAX(Gradedate) as maxdate from transcript WHERE
StudentId="'+edit1.text+'"');
  open;
end;

```

```

frm_main.DataSource10.DataSet.First;
if not(frm_main.DataSource11.DataSet.Eof) then
begin
  while not(frm_main.DataSource10.DataSet.Eof) do
  begin
    with frm_main.DataSource10 do
    begin
      if
(dataset.FieldValues['GradeDate']>=frm_main.DataSource11.DataSet.FieldValues['maxdate'])
then
      begin
        sumcredit:=sumcredit+dataset.FieldValues['credit'];

        if dataset.FieldValues['grade']='AA' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*4;
        if dataset.FieldValues['grade']='BA' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*3.5;
        if dataset.FieldValues['grade']='BB' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*3;
        if dataset.FieldValues['grade']='CB' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*2.5;
        if dataset.FieldValues['grade']='CC' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*2;
        if dataset.FieldValues['grade']='DC' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*1.5;
        if dataset.FieldValues['grade']='DD' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*1;
        if dataset.FieldValues['grade']='FD' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*0.5;
        if dataset.FieldValues['grade']='FF' then
          sumgrade:=sumgrade+dataset.FieldValues['credit']*0;

        dataset.Next;
      end
    else
    begin
      dataset.Next;
    end;
  end; // with

end; //while
if ((sumcredit>0) and (sumgrade>0)) then

```

```

begin
str((sumgrade/sumcredit):3:2,t);
edit5.Text:=t;
end;
// edit6.Text:=floattostr(sumgrade);
end
else
begin
messagedlg('This Student Never Registered Before...!',mtinformation,[mbok],0);
gpa:=0;
end;

//end of gpa

// start of cgpa
sumgencredit:=0;
sumgengrade:=0;
cgpa:=0;

with frm_main.ADOQuery4 do
begin
sql.Clear;
sql.add('Select t.GradeDate,t.studentId,t.grade,t.courseID,c.credit from transcript as t
inner join course as c on t.courseID=c.courseID where t.studentId="'+edit1.Text+'" and
t.grade is not null');
open;
end;

frm_main.DataSource10.DataSet.First;

while not(frm_main.DataSource10.DataSet.Eof) do
begin
with frm_main.DataSource10 do
begin
label10.Caption:=frm_main.ADOQuery4.FieldValues['courseId'];
with frm_main.ADOQuery6 do
begin
sql.Clear;
sql.add('Select t.GradeDate,t.studentId,t.grade,t.courseID,c.credit from transcript as t
inner join course as c on t.courseID=c.courseID where t.studentId="'+edit1.Text+'" and
t.courseId='+label10.Caption+' order by gradedate desc');
open;
end;

frm_main.DataSource12.DataSet.First;

if
dataset.FieldValues['gradedate']=frm_main.DataSource12.DataSet.FieldValues['gradedate']
then
begin

```

```

sumgencredit:=sumgencredit+dataset.FieldValues['credit'];

if dataset.FieldValues['grade']='AA' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*4;
if dataset.FieldValues['grade']='BA' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*3.5;
if dataset.FieldValues['grade']='BB' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*3;
if dataset.FieldValues['grade']='CB' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*2.5;
if dataset.FieldValues['grade']='CC' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*2;
if dataset.FieldValues['grade']='DC' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*1.5;
if dataset.FieldValues['grade']='DD' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*1;
if dataset.FieldValues['grade']='FD' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*0.5;
if dataset.FieldValues['grade']='FF' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*0;
dataset.Next;

end
else
begin
    dataset.Next;
end;    // end of if

end;    //end of with

end;    //end of while

if ((sumgencredit>0) and (sumgengrade>0)) then
begin
    str((sumgengrade/sumgencredit):3:2,t);
    edit6.Text:=t;
end;

label10.Caption:=Inttostr(sumgencredit);
label11.Caption:=floattostr(sumgengrade);

// end of cgpa....
with frm_main.ADOQuery2 do    //Find if already registered
begin
    sql.Clear;
    sql.Add('SELECT grade FROM transcript where StudentId="'+edit1.Text+'" and grade
is null');

```

```

open;
end;

if frm_main.DataSource8.DataSet.RecordCount>0 then
begin
    messagedlg('This Student Has Already Been Registered, Use Add/Drop Menu To update
Course Registration ..!',mtwarning,[mbok],0);
    Edit1.Clear;
    Edit2.Clear;
    Edit3.Clear;
    Edit4.Clear;
    Edit5.Clear;
    Edit6.Clear;
    Listbox1.Items.Clear;
    listbox2.Items.Clear;
    edit1.ReadOnly:=false;
    edit1.SetFocus;
    exit;

end;

with frm_main.ADOQuery2 do      //Find failed courses
begin
    sql.Clear;
    sql.Add('SELECT c.coursecode,t.grade,t.courseId FROM course as c inner join
transcript as t');
    sql.add('on c.courseId=t.courseId WHERE StudentId="'+edit1.Text+'"'); // and ((t.grade
is null) or(t.grade="FF") or');
    // sql.Add('(t.grade="FD"))');
    open;
end;

frm_main.DataSource8.DataSet.first;
while not(frm_main.DataSource8.DataSet.eof) do
begin
    if ((frm_main.DataSource8.DataSet.FieldValues['grade']='FF') or
(frm_main.DataSource8.DataSet.FieldValues['grade']='FD')) then
begin
    label15.Caption:=inttostr(frm_main.DataSource8.DataSet.FieldValues['courseId']);
    with frm_main.ADOQuery5 do      //Find if passed
begin
        sql.Clear;
        sql.Add('SELECT grade FROM transcript WHERE StudentId="'+edit1.Text+'" and
courseId='+label15.caption+' order by gradedate desc');
        open;
end;

        if
((frm_main.DataSource11.DataSet.FieldValues['grade']='FF')OR(frm_main.DataSource11.Da
taSet.FieldValues['grade']='FD')) THEN
begin

```



```

        listbox1.Font.Color:=clred;
        listbox1.Items.Add(' '+frm_main.DataSource8.DataSet.FieldValues['coursecode']+' -->
'+frm_main.DataSource8.DataSet.FieldValues['grade']);
        frm_main.DataSource8.DataSet.next;
    end
else
begin
    frm_main.DataSource8.DataSet.next;

    end;
end
else
begin
    frm_main.DataSource8.DataSet.Next;
end;

end;
with frm_main.ADOQuery2 do      //find unregistered courses
begin
    sql.Clear;
    sql.Add('SELECT coursecode FROM course ');
    sql.add('WHERE courseId not in (SELECT c.courseID FROM course as c inner join
transcript as t');
    sql.Add('on c.courseId=t.courseId WHERE StudentId="'+edit1.Text+'")'); // and ((t.grade
is null) or(t.grade="FF") or(t.grade="FD"))));
    open;
end;

frm_main.DataSource8.DataSet.first;
while not(frm_main.DataSource8.DataSet.eof) do
begin
    listbox1.Font.Color:=clblack;
    listbox1.Items.Add(' '+frm_main.DataSource8.DataSet.FieldValues['coursecode']);
    frm_main.DataSource8.DataSet.next;
end ;
with frm_main.ADOQuery2 do      //Already passed courses
begin
    sql.Clear;
    sql.Add('SELECT c.coursecode,t.grade FROM course as c inner join transcript as t');
    sql.add('on c.courseId=t.courseId WHERE StudentId="'+edit1.Text+'")'); // and ((t.grade
is null) or(t.grade="FF") or');
    // sql.Add('(t.grade="FD"))');
    open;
end;

frm_main.DataSource8.DataSet.first;
while not(frm_main.DataSource8.DataSet.eof) do
begin
    if not((frm_main.DataSource8.DataSet.FieldValues['grade']='FF') or
(frm_main.DataSource8.DataSet.FieldValues['grade']='FD')) then

```



```

begin
  listbox1.Font.Color:=clred;
  listbox1.Items.Add(' '+frm_main.DataSource8.DataSet.FieldValues['coursecode']+' -->
'+frm_main.DataSource8.DataSet.FieldValues['grade']+' ->PASSED');
  frm_main.DataSource8.DataSet.next;
end
else
begin

```

```

  frm_main.DataSource8.DataSet.Next;
end;
end;

```

```

end
else
begin
  messagedlg('Student ID is not Found !!',mtinformation,[mbok],0);
  edit1.SetFocus;
end;
end;
end;

```

```

procedure Tfrm_correg.Button2Click(Sender: TObject);
var i:integer;
begin
  if ((listbox2.Items.Count<7) and (listbox1.ItemIndex>=0)) then
  begin
    i:=listbox1.ItemIndex;
    listbox2.Items.Add(' '+listbox1.Items.ValueFromIndex[listbox1.itemindex]);//
listbox1..SelCount.ValueFromIndex[listbox1.Items.)
    listbox1.Items.Delete(listbox1.ItemIndex);

```

```

end
else
begin
  messagedlg('You can not register any more course !!!',mtwarning,[mbok],0);
  button5.setfocus;
end;
end;

```

```

procedure Tfrm_correg.FormShow(Sender: TObject);
begin
  Edit1.Clear;
  Edit2.Clear;
  Edit3.Clear;
  Edit4.Clear;
  Edit5.Clear;
  Edit6.Clear;
  Listbox1.Items.Clear;

```

```
listbox2.Items.Clear;
edit1.ReadOnly:=false;
edit1.SetFocus;
```

```
end;
```

```
procedure Tfrm_correg.Button3Click(Sender: TObject);
```

```
begin
  if listbox2.ItemIndex>=0 then
    begin
      listbox1.Items.Add(' '+listbox2.Items.ValueFromIndex[listbox2.itemindex]);//
      listbox1..SelCount.ValueFromIndex[listbox1.Items.)
      listbox2.Items.Delete(listbox2.ItemIndex);
    end;
  end;
```

```
procedure Tfrm_correg.Button4Click(Sender: TObject);
var i:integer;
```

```
begin
  i:=0;
  while (i<(listbox2.Count)) do
    begin
      with frm_main do
        begin

          Datasource2.Edit;
          datasource2.DataSet.Append;
          datasource2.DataSet.FieldValues['StudentId']:=edit1.Text;
          // adotable2.FieldValues['StudentId']:=edit1.Text;
          // adotable2.FieldValues['coursecode']:=listbox2.Items.ValueFromIndex[i];
          if pos(' ',listbox2.Items.ValueFromIndex[i])>0 then
            begin
              with frm_main.ADOQuery4 do          //find unregistered courses
                begin
                  sql.Clear;
                  sql.Add('SELECT courseId FROM course where
coursecode="'+(midstr(listbox2.items.ValueFromIndex[i],1,(pos('-
',ansireplacestr(listbox2.items.ValueFromIndex[i],',')))))+"'');
                  open;
                end;

          datasource2.DataSet.FieldValues['courseid']:=datasource10.DataSet.FieldValues['courseid'];
          end
        else
          begin
            with frm_main.ADOQuery4 do          //find unregistered courses
              begin
```

```

    sql.Clear;
    sql.Add('SELECT courseId FROM course where
coursecode="'+listbox2.Items.ValueFromIndex[i]+'");
    open;
end;

datasource2.DataSet.FieldValues['courseid']:=datasource10.DataSet.FieldValues['courseid'];
end;
datasource2.DataSet.FieldValues['coursestartdate']:=date;
datasource2.DataSet.FieldValues['processuser']:=username;
datasource2.DataSet.FieldValues['processdate']:=date;
datasource2.DataSet.Post;
end;
i:=i+1;
end;
if listbox2.Count>0 then
begin
    messagedlg(' Course Registration completed successfully!!',mtinformation,[mbok],0);
    Edit1.Clear;
    Edit2.Clear;
    Edit3.Clear;
    Edit4.Clear;
    Edit5.Clear;
    Edit6.Clear;
    Listbox1.Items.Clear;
    listbox2.Items.Clear;
    edit1.SetFocus;
    edit1.ReadOnly:=false;
end
else
    messagedlg('Please Select courses to register!!',mtwarning,[mbok],0);
end;

procedure Tfrm_correg.Button7Click(Sender: TObject);
begin
    frm_correg.Hide;
    frm_beginsemester.show;
end;
end.

```

unit Unit23;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;

type
 TRep_dept_cor = class(TForm)

```

ADOQuery1: TADOQuery;
DataSource1: TDataSource;
ComboBox1: TComboBox;
Label1: TLabel;
Label2: TLabel;
DataSource2: TDataSource;
ADOQuery2: TADOQuery;
DBGrid1: TDBGrid;
Button1: TButton;
Button2: TButton;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure ComboBox1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Rep_dept_cor: TRep_dept_cor;

implementation

uses Unit16,unit26;

{$R *.dfm}

procedure TRep_dept_cor.FormCreate(Sender: TObject);
begin
  with ADOQuery1 do
    begin
      SQL.Clear;
      sql.add('SELECT departmentId FROM department order by departmentId');
      open;
    end;

    DataSource1.DataSet.First;
    while not (DataSource1.DataSet.Eof) do
      begin
        combobox1.Items.add(DataSource1.DataSet.FieldValues['DepartmentID']);

        DataSource1.DataSet.Next;
      end;
    end;
end;

```

```

procedure TRep_dept_cor.Button1Click(Sender: TObject);
begin
  Rep_dept_cor.print;
end;

```

```

procedure TRep_dept_cor.Button2Click(Sender: TObject);
begin
  rep_dept_cor.Close;
end;

```

```

• procedure TRep_dept_cor.ComboBox1Click(Sender: TObject);
begin
  with ADOQuery2 do
  begin
    SQL.Clear;
    sql.add('SELECT * FROM course where departmentID="'+combobox1.Text+'" order by
coursename');
    open;
  end;

```

```

end;

```

```

procedure TRep_dept_cor.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  rep_main.show;
end;
end.

```

unit Unit24;

```

interface

```

```

uses

```

```

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;

```

```

type

```

```

  TRep_all_stu = class(TForm)

```

```

    Label1: TLabel;

```

```

    ADOQuery1: TADOQuery;

```

```

    DataSource1: TDataSource;

```

```

    DBGrid1: TDBGrid;

```

```

    Button1: TButton;

```

```

    Button2: TButton;

```

```

    procedure Button2Click(Sender: TObject);

```

```

    procedure Button1Click(Sender: TObject);

```

```

    procedure FormCreate(Sender: TObject);

```

```

    procedure FormClose(Sender: TObject; var Action: TCloseAction);

```

```

  private

```

```

    { Private declarations }

```

```

  public

```



```

    { Public declarations }
end;

var
    Rep_all_stu: TRep_all_stu;
implementation
uses Unit16,unit26;
{$R *.dfm}
procedure TRep_all_stu.Button2Click(Sender: TObject);
begin
    Rep_all_stu.Close;
end;
procedure TRep_all_stu.Button1Click(Sender: TObject);
begin
    Rep_all_stu.Print;
end;
procedure TRep_all_stu.FormCreate(Sender: TObject);
begin
    with ADOQuery1 do
    begin
        SQL.Clear;
        sql.add('SELECT * FROM student');
        open;
    end;
end;

end;
procedure TRep_all_stu.FormClose(Sender: TObject;
    var Action: TCloseAction);
begin
    rep_main.show;
end;
end.

```

unit Unit25;

interface

uses

Windows, Messages; SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls;

type

```

TRep_dept_stu = class(TForm)
    Button1: TButton;
    Button2: TButton;
    ComboBox1: TComboBox;
    Label1: TLabel;
    Label2: TLabel;
    ADOQuery1: TADOQuery;
    ADOQuery2: TADOQuery;

```

```

DataSource1: TDataSource;
DataSource2: TDataSource;
DBGrid1: TDBGrid;
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ComboBox1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Rep_dept_stu: TRep_dept_stu;

implementation

uses Unit16,unit26;

{$R *.dfm}

procedure TRep_dept_stu.Button2Click(Sender: TObject);
begin
    Rep_dept_stu.Close;
end;

procedure TRep_dept_stu.Button1Click(Sender: TObject);
begin
    //Rep_dept_stu.Print;
end;

procedure TRep_dept_stu.FormCreate(Sender: TObject);
begin
    with ADOQuery1 do
        begin
            SQL.Clear;
            sql.add('SELECT departmentId FROM department order by departmentId');
            open;
        end;

    DataSource1.DataSet.First;
    while not (DataSource1.DataSet.Eof) do
        begin
            combobox1.Items.add(DataSource1.DataSet.FieldValues['DepartmentID']);

            DataSource1.DataSet.Next;
        end;end;
end;end;

```

```

procedure TRep_dept_stu.ComboBox1Click(Sender: TObject);
begin
    with ADOQuery2 do
    begin
        SQL.Clear;
        sql.add('SELECT * FROM student where departmentID="'+combobox1.Text+'" order by
name,surname');
        open;
    end;
end;

procedure TRep_dept_stu.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
    rep_main.show;
end;
end.

```

unit Unit26;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, DB, ADODB, jpeg, ExtCtrls, WinSkinData;

type

```

Tfrm_password = class(TForm)
    Button1: TButton;
    EXIT: TButton;
    ADOTable1: TADOTable;
    ADOQuery1: TADOQuery;
    DataSource2: TDataSource;
    SkinData1: TSkinData;
    Image2: TImage;
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Image4: TImage;
    Image5: TImage;
    Label3: TLabel;
    Label6: TLabel;
    Image3: TImage;
    Image6: TImage;
    Edit2: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    Edit1: TEdit;
    DataSource1: TDataSource;

```

```

procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure EXITClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frm_password: Tfrm_password;
  level:string;
  username:string;
  trial:integer;
implementation

uses Unit1;

{$R *.dfm}

procedure Tfrm_password.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  application.Terminate;
end;

procedure Tfrm_password.EXITClick(Sender: TObject);
begin
  frm_password.Close;
end;

procedure Tfrm_password.FormCreate(Sender: TObject);
begin
  trial:=0;
  //datasource1.DataSet.First;
  /*while not datasource1.DataSet.Eof do
  //begin
  //combobox1.Items.Add(datasource1.DataSet.FieldValues['LoginName']);
  //datasource1.DataSet.Next;
  //end;
end;

procedure Tfrm_password.Button1Click(Sender: TObject);
begin
  with adoquery1 do
  begin
    sql.Clear;
    sql.Add('Select * from login where LoginName="'+Edit1.Text+'"');
    open;

```

```

end;

begin
if edit2.Text=datasource2.DataSet.FieldValues['password'] then
begin
level:=datasource2.DataSet.FieldValues['AccessLevel'];
username:=datasource2.DataSet.FieldValues['LoginName'];
edit2.Text:="";
edit1.Text:="";
label4.Visible:=false;
label5.Visible:=false;
edit1.SetFocus;
frm_password.Hide;
frm_main.Show;
end
else
begin
edit2.Text:="";
trial:=trial+1;
label4.Show;
label5.Show;
//MessageDlg('Wrong Password. Please Enter again. '+IntToStr(3-trial)+' Trials Left',
mtInformation,[mbOk], 0);
edit1.SetFocus;
end;
end;
if trial=3 then
begin
MessageDlg('You Are Not An Accessible Person To This System', mtInformation,[mbOk],
0);
application.Terminate;
end;
end;
end.

```

unit Unit27;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Buttons, StdCtrls, ExtCtrls, ComCtrls, jpeg;

type

```

Tfrm_login = class(TForm)
  Image1: TImage;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;

```



```

Edit1: TEdit;
Edit2: TEdit;
ComboBox1: TComboBox;
RadioGroup1: TRadioGroup;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Button5: TButton;
Button6: TButton;
ComboBox2: TComboBox;
Button7: TButton;
Label4: TLabel;
Label5: TLabel;
Image2: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure BitBtn1Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure ComboBox2Select(Sender: TObject);
procedure ComboBox2KeyPress(Sender: TObject; var Key: Char);
procedure ComboBox1KeyPress(Sender: TObject; var Key: Char);
procedure Button7Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frm_login: Tfrm_login;
process:string;
implementation

uses Unit1,unit26;

{$R *.dfm}

procedure Tfrm_login.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  frm_main.show;
  combobox2.Visible:=false;
end;

```

```
procedure Tfrm_login.BitBtn1Click(Sender: TObject);
begin
    frm_login.Close;
end;
```

```
procedure Tfrm_login.Button6Click(Sender: TObject);
begin
    frm_login.Close;
end;
```

```
procedure Tfrm_login.FormShow(Sender: TObject);
begin
    edit1.Clear;
    edit1.Visible:=true;
    edit2.Clear;
    combobox1.Clear;
    combobox2.Clear;
    combobox2.Visible:=false;
    edit1.Enabled:=false;
    edit2.Enabled:=false;
    combobox1.Enabled:=false;
    radiogroup1.Enabled:=false;
    button3.Enabled:=false;
    button4.Enabled:=false;
    button5.Enabled:=false;
    if level='Admin' then
    begin
        button3.Enabled:=true;
        button4.Enabled:=true;
        button5.Enabled:=true;
    end;
end;
```

```
procedure Tfrm_login.Button3Click(Sender: TObject);
begin
    edit1.Enabled:=true;
    edit1.Visible:=true;
    combobox2.Visible:=false;
    edit1.Clear;
    edit2.Clear;
    combobox1.Clear;
    combobox2.Clear;
    combobox2.Enabled:=false;
    with frm_main.ADOQuery1 do
    begin
        sql.clear;
        sql.Add('Select DepartmentId from Department');
        open;
    end;
    frm_main.datasource7.DataSet.First;
```

```
while not frm_main.datasource7.DataSet.Eof do
```

```
begin
```

```
    combobox1.Items.Add(frm_main.datasource7.DataSet.FieldValues['DepartmentId']);
```

```
    frm_main.DataSource7.DataSet.Next;
```

```
end;
```

```
edit1.Enabled:=true;
```

```
edit2.Enabled:=true;
```

```
combobox1.Enabled:=true;
```

```
combobox2.Enabled:=false;
```

```
edit1.SetFocus;
```

```
radiogroup1.Enabled:=true;
```

```
edit1.Clear;
```

```
edit2.Clear;
```

```
radiogroup1.ItemIndex:=2;
```

```
process:='add'; //to inform save button
```

```
end;
```

```
procedure Tfrm_login.FormCreate(Sender: TObject);
```

```
begin
```

```
    process:="";
```

```
end;
```

```
procedure Tfrm_login.Button2Click(Sender: TObject);
```

```
begin
```

```
    if (process='add') then
```

```
    begin
```

```
        if
```

```
((trim(edit1.text)<>"")and(trim(edit2.text)<>"")and(trim(combobox1.text)<>"")and(radiogroup1.  
ItemIndex>=0)) then
```

```
    begin
```

```
        if process='add' then
```

```
        begin
```

```
            with frm_main.ADOQuery2 do
```

```
            begin
```

```
                sql.Clear;
```

```
                sql.Add('select loginname from login where loginname="'+edit1.Text+'"');
```

```
                open;
```

```
            end;
```

```
//    frm_main.DataSource8.DataSet.First;
```

```
    if frm_main.DataSource8.DataSet.RecordCount>0 then
```

```
    begin
```

```
        messagedlg('Please select another user name!!',mtinformation,[mbok],0);
```

```
        edit1.Clear;
```

```
        edit2.Clear;
```

```
        exit;
```

```
    end
```

```
    else
```

```
        begin
```

```
            with frm_password.DataSource1 do
```

```

begin
    dataset.Edit;
    dataset.Append;
    dataset.FieldValues['LoginName']:=edit1.Text;
    dataset.FieldValues['Password']:=edit2.Text;
    dataset.FieldValues['departmentId']:=combobox1.text;
    case radiogroup1.ItemIndex of
        0:dataset.FieldValues['AccessLevel']:='Admin';
        1:dataset.FieldValues['AccessLevel']:='Advisor';
        2:dataset.FieldValues['AccessLevel']:='User';
    end;
    dataset.post;
    edit1.Clear;
    edit2.Clear;
    combobox1.Clear;
    combobox2.Clear;
    edit1.Enabled:=false;
    edit2.Enabled:=false;
    combobox1.Enabled:=false;
    radiogroup1.Enabled:=false;
    process:="";
end;
end;
end;
end
else
begin
    messagedlg('Please Fill All Fields And Select Access Level',mtinformation,[mbok],0);
end;
end;

if process='delete' then
begin
    if combobox1.text="" then
    begin
        edit1.Visible:=true;
        combobox2.Visible:=false;
        combobox2.Enabled:=false;
        process:="";
    end
    else
    begin
        edit1.Enabled:=true;
        edit2.Enabled:=true;
        combobox1.Enabled:=true;
        combobox2.Visible:=false;
        combobox2.Enabled:=false;

        frm_password.DataSource2.DataSet.Delete;
        messagedlg('The record successfully deleted',mtinformation,[mbok],0);
    end
    end
end

```



```

edit1.Clear;
edit2.Clear;
combobox1.Clear;
combobox2.Clear;
edit1.Visible:=true;
edit1.Enabled:=false;
edit2.Enabled:=false;
combobox1.Enabled:=false;
combobox2.Enabled:=false;
combobox2.Visible:=false;
//  button3.Enabled:=false;
//  button4.Enabled:=false;
//  button5.Enabled:=false;
end;
end;
if process='changepassword' then
begin
if trim(edit2.text)<>" then
begin
frm_password.DataSource2.DataSet.edit;
frm_password.DataSource2.DataSet.FieldValues['password']:=edit2.Text;
frm_password.DataSource2.DataSet.Post;
edit1.clear;
edit2.Clear;
combobox1.Clear;
process:="";
end
else
messagedlg('Blank Password Is Not Allowed...!!',mtinformation,[mbok],0);
edit2.SetFocus;
end;

if process='changeaccess' then
begin
with frm_password.ADOquery1 do
begin
sql.clear;
sql.Add('Select * from Login where Loginname="'+username+'");
open;
end;
frm_password.DataSource2.DataSet.edit;
case radiogroup1.ItemIndex of
0:frm_password.DataSource2.dataset.FieldValues['AccessLevel']:='Admin';
1:frm_password.DataSource2.dataset.FieldValues['AccessLevel']:='Advisor';
2:frm_password.DataSource2.dataset.FieldValues['AccessLevel']:='User';
end;
frm_password.DataSource2.DataSet.Post;
edit1.clear;
edit2.Clear;

```



```
combobox1.Clear;
combobox2.Items.Clear;
combobox2.Clear;
combobox2.Visible:=false;
edit1.Visible:=true;
radiogroup1.Enabled:=false;
process:="";
end;
```

```
end;
procedure Tfrm_login.Button4Click(Sender: TObject);
begin
    edit1.Clear;
    edit2.Clear;
    combobox1.Clear;
    combobox2.Clear;
    edit1.Enabled:=false;
    combobox2.Enabled:=true;
    with frm_password.ADOQuery1 do
    begin
        sql.clear;
        sql.Add('Select LoginName from Login');
        open;
    end;
    frm_password.datasource2.DataSet.First;
    while not frm_password.datasource2.DataSet.Eof do
    begin
        combobox2.Items.Add(frm_password.datasource2.DataSet.FieldValues['LoginName']);
        frm_password.DataSource2.DataSet.Next;
    end;
    // edit2.Enabled:=true;
    // combobox1.Enabled:=true;
    edit1.Visible:=false;
    combobox2.Visible:=true;
    edit2.Enabled:=false;

    process:='delete'; //to inform save button
end;
```

```
procedure Tfrm_login.ComboBox2Select(Sender: TObject);
begin
    with frm_password.ADOQuery1 do
    begin
        sql.clear;
        sql.Add('Select * from Login where loginname="'+combobox2.text+'"');
        open;
    end;
    combobox1.Text:=frm_password.DataSource2.DataSet.FieldValues['DepartmentId'];
    combobox1.Enabled:=false;
```

```

if frm_password.DataSource2.DataSet.FieldValues['AccessLevel']='Admin' then
    radiogroup1.ItemIndex:=0;
if frm_password.DataSource2.DataSet.FieldValues['AccessLevel']='Advisor' then
    radiogroup1.ItemIndex:=1;
if frm_password.DataSource2.DataSet.FieldValues['AccessLevel']='User' then
    radiogroup1.ItemIndex:=2;
if process<>'changeaccess' then
    radiogroup1.Enabled:=false;

end;

procedure Tfrm_login.ComboBox2KeyPress(Sender: TObject; var Key: Char);
begin
    KEY:=#0;
end;

procedure Tfrm_login.ComboBox1KeyPress(Sender: TObject; var Key: Char);
begin
    KEY:=#0;
end;

procedure Tfrm_login.Button7Click(Sender: TObject);
begin
    edit1.Clear;
    edit2.Clear;
    combobox1.Text:="";
    combobox2.Text:="";
    combobox1.Items.Clear;
    combobox2.Items.Clear;
    radiogroup1.Enabled:=false;
    process:="";
end;

procedure Tfrm_login.Button1Click(Sender: TObject);
begin
    process:='changepassword';
    combobox2.Visible:=false;
    edit1.Visible:=true;
    combobox2.Enabled:=false;
    edit1.Enabled:=false;
    edit2.Enabled:=true;
    edit2.Visible:=true;
    combobox1.Enabled:=false;
    combobox2.Text:=username;
    edit1.Text:=username;
    edit2.SetFocus;

    with frm_password.ADOQuery1 do
    begin

```

```

    sql.clear;
    sql.Add('Select * from Login where Loginname="'+username+'"');
    open;
end;
combobox1.Text:=frm_password.DataSource2.DataSet.FieldValues['DepartmentId'];

// radiogroup1.ItemIndex:=frm_password.DataSource2.DataSet.FieldValues['AccessLevel'];
end;

procedure Tfrm_login.Button5Click(Sender: TObject);
begin
    process:='changeaccess';
    combobox2.Visible:=true;
    combobox2.Enabled:=true;
    edit1.Enabled:=false;
    edit2.Enabled:=true;
    edit2.Visible:=true;
    combobox1.Enabled:=false;
    combobox2.Text:=username;

    with frm_password.ADOQuery1 do
    begin
        sql.clear;
        sql.Add('Select * from Login where Loginname="'+username+'"');
        open;
    end;
    radiogroup1.Enabled:=true;
    if frm_password.DataSource2.DataSet.FieldValues['AccessLevel']='Admin' then
        radiogroup1.ItemIndex:=0;
    if frm_password.DataSource2.DataSet.FieldValues['AccessLevel']='Advisor' then
        radiogroup1.ItemIndex:=1;
    if frm_password.DataSource2.DataSet.FieldValues['AccessLevel']='User' then
        radiogroup1.ItemIndex:=2;
    combobox1.Text:=frm_password.DataSource2.DataSet.FieldValues['DepartmentId'];

    with frm_password.ADOQuery1 do
    begin
        sql.clear;
        sql.Add('Select * from Login');
        open;
    end;
    frm_password.datasource2.DataSet.First;
    while not frm_password.datasource2.DataSet.Eof do
    begin
        combobox2.Items.Add(frm_password.datasource2.DataSet.FieldValues['LoginName']);
        frm_password.DataSource2.DataSet.Next;
    end;
end;
end.

```

unit Unit28;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Buttons, StdCtrls, jpeg, ExtCtrls;

type

Tfrm_beginsemester = class(TForm)

Image1: TImage;

SpeedButton1: TSpeedButton;

SpeedButton3: TSpeedButton;

SpeedButton4: TSpeedButton;

Label2: TLabel;

Label1: TLabel;

Image2: TImage;

procedure SpeedButton3Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure SpeedButton1Click(Sender: TObject);

procedure SpeedButton2Click(Sender: TObject);

procedure SpeedButton4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

frm_beginsemester: Tfrm_beginsemester;

implementation

uses Unit1, Unit22, Unit29, Unit15;

{ \$R *.dfm }

procedure Tfrm_beginsemester.SpeedButton3Click(Sender: TObject);

begin

frm_beginsemester.Close;

end;

procedure Tfrm_beginsemester.FormClose(Sender: TObject;

var Action: TCloseAction);

begin

frm_main.show;

end;


```
procedure Tfrm_beginsemester.SpeedButton1Click(Sender: TObject);
begin
    frm_correg.show;
    frm_beginsemester.Hide;
end;
```

```
procedure Tfrm_beginsemester.SpeedButton2Click(Sender: TObject);
begin
    frm_adddrop.show;
    frm_beginsemester.Hide;
end;
```

```
procedure Tfrm_beginsemester.SpeedButton4Click(Sender: TObject);
begin
    frm_beginsemester.Hide;
    frm_semend.show;
end;
end.
```

unit Unit29;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, strutils, jpeg;

type

```
Tfrm_adddrop = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    Label18: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    Edit2: TEdit;
    Edit3: TEdit;
```



```

Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Button4: TButton;
Button5: TButton;
Button6: TButton;
Button7: TButton;
Button8: TButton;
Label19: TLabel;
Label20: TLabel;
Button2: TButton;
ListBox1: TListBox;
ListBox2: TListBox;
Button3: TButton;
Image2: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button4Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  frm_adddrop: Tfrm_adddrop;

```

implementation

```

uses Unit28, Unit1, unit26;

```

```

{$R *.dfm}

```

```

procedure Tfrm_adddrop.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  frm_beginsemester.show;
end;

procedure Tfrm_adddrop.Button6Click(Sender: TObject);
begin
  frm_adddrop.Hide;
  frm_main.show;
end;

```

```

procedure Tfrm_adddrop.Button7Click(Sender: TObject);
begin
    frm_adddrop.Close;
end;

```

```

procedure Tfrm_adddrop.Button1Click(Sender: TObject);
var sumcredit:integer;
var sumgrade,gpa:real;
var sumgencredit:integer;
var sumgengrade,cgpa:real;
//var cid:integer;

```

```

//var ypart:integer;
var mpart:string;
var ShortMonthNames:string;

```

```

begin
    if edit1.text<>null then
        begin
            with frm_main.ADOQuery1 do
                begin
                    sql.Clear;
                    sql.Add('SELECT * FROM student WHERE StudentId="'+edit1.Text+'"');
                    open;
                end;
            if not(frm_main.DataSource7.DataSet.eof) then
                begin
                    edit2.Text:=frm_main.DataSource7.DataSet.FieldValues['name'];
                    edit3.Text:=frm_main.DataSource7.DataSet.FieldValues['surname'];
                    edit4.Text:=frm_main.DataSource7.DataSet.FieldValues['departmentID'];

                    edit1.ReadOnly:=true;

                    shortdateformat:=('mm');
                    ShortMonthNames:=datetostr(date);
                    label12.Caption:=ShortMonthNames;
                    if
                        ((ShortMonthNames='01')or(ShortMonthNames='12')or(ShortMonthNames='11')or(ShortMonthNames='10')or(ShortMonthNames='09')or(ShortMonthNames='08')) then
                            mpart:='Fall'
                        else
                            mpart:='Spring';

                    shortdateformat:=('yyyy');
                    // ypart:=date;
                    label9.Caption:=datetostr(date)+' - '+mpart+' Term registration ';

                    shortdateformat:=('dd/mm/yyyy');

```

// To calculate gpa.....

```
sumcredit:=0;
sumgrade:=0;
gpa:=0;
sumgencredit:=0;
sumgengrade:=0;
cgpa:=0;
```

```
with frm_main.ADOQuery4 do
begin
  sql.Clear;
  sql.add('Select t.GradeDate,t.studentId,t.grade,t.courseID,c.credit from transcript as t
inner join course as c on t.courseID=c.courseID where t.studentId="'+edit1.Text+'" and
t.grade is not null');
  open;
end;
```

```
with frm_main.ADOQuery5 do
begin
  sql.Clear;
  sql.add('Select MAX(Gradedate) as maxdate from transcript');
  open;
end;
```

```
frm_main.DataSource10.DataSet.First;
if not(frm_main.DataSource11.DataSet.Eof) then
begin
  while not(frm_main.DataSource10.DataSet.Eof) do
  begin
    with frm_main.DataSource10 do
    begin
      if
then
  (dataset.FieldValues['GradeDate']>=frm_main.DataSource11.DataSet.FieldValues['maxdate'])
then
```

```
begin
  sumcredit:=sumcredit+dataset.FieldValues['credit'];
```

```
if dataset.FieldValues['grade']='AA' then
  sumgrade:=sumgrade+dataset.FieldValues['credit']*4;
if dataset.FieldValues['grade']='BA' then
  sumgrade:=sumgrade+dataset.FieldValues['credit']*3.5;
if dataset.FieldValues['grade']='BB' then
  sumgrade:=sumgrade+dataset.FieldValues['credit']*3;
if dataset.FieldValues['grade']='CB' then
  sumgrade:=sumgrade+dataset.FieldValues['credit']*2.5;
if dataset.FieldValues['grade']='CC' then
  sumgrade:=sumgrade+dataset.FieldValues['credit']*2;
if dataset.FieldValues['grade']='DC' then
  sumgrade:=sumgrade+dataset.FieldValues['credit']*1.5;
```

```

    if dataset.FieldValues['grade']='DD' then
        sumgrade:=sumgrade+dataset.FieldValues['credit']*1;
    if dataset.FieldValues['grade']='FD' then
        sumgrade:=sumgrade+dataset.FieldValues['credit']*0.5;
    if dataset.FieldValues['grade']='FF' then
        sumgrade:=sumgrade+dataset.FieldValues['credit']*0;

    dataset.Next;
end
else
begin
    dataset.Next;
end;
end; // with

end; //while
if ((sumcredit>0) and (sumgrade>0)) then
    edit5.Text:=floattostr(sumgrade/sumcredit);
//    edit6.Text:=floattostr(sumgrade);
end
else
begin
    messagedlg('This Student Never Registered Before...!',mtinformation,[mbok],0);
    gpa:=0;
end;

//end of gpa

// start of cgpa
sumgencredit:=0;
sumgengrade:=0;
cgpa:=0;

with frm_main.ADOQuery4 do
begin
    sql.Clear;
    sql.add('Select t.GradeDate,t.studentId,t.grade,t.courseID,c.credit from transcript as t
inner join course as c on t.courseID=c.courseID where t.studentId="'+edit1.Text+'" and
t.grade is not null');
    open;
end;

frm_main.DataSource10.DataSet.First;

while not(frm_main.DataSource10.DataSet.Eof) do
begin
    with frm_main.DataSource10 do
begin
        label10.Caption:=frm_main.ADOQuery4.FieldValues['courseId'];
        with frm_main.ADOQuery6 do

```



```

begin
    sql.Clear;
    sql.add('Select t.GradeDate,t.studentId,t.grade,t.courseID,c.credit from transcript as t
inner join course as c on t.courseID=c.courseID where t.studentId="'+edit1.Text+"' and
t.courseId='"+label10.Caption+"' order by gradedate desc');
    open;
end;

```

```

frm_main.DataSource12.DataSet.First;

```

```

if
dataset.FieldValues['gradedate']=frm_main.DataSource12.DataSet.FieldValues['gradedate']
then

```

```

begin
    sumgencredit:=sumgencredit+dataset.FieldValues['credit'];

```

```

if dataset.FieldValues['grade']='AA' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*4;
if dataset.FieldValues['grade']='BA' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*3.5;
if dataset.FieldValues['grade']='BB' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*3;
if dataset.FieldValues['grade']='CB' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*2.5;
if dataset.FieldValues['grade']='CC' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*2;
if dataset.FieldValues['grade']='DC' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*1.5;
if dataset.FieldValues['grade']='DD' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*1;
if dataset.FieldValues['grade']='FD' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*0.5;
if dataset.FieldValues['grade']='FF' then
    sumgengrade:=sumgengrade+dataset.FieldValues['credit']*0;
dataset.Next;

```

```

end

```

```

else

```

```

begin

```

```

    dataset.Next;

```

```

end;    // end of if

```

```

end;    //end of with

```

```

end;    //end of while

```

```

if ((sumgencredit>0) and (sumgengrade>0)) then
    edit6.Text:=floattostr(sumgengrade/sumgencredit);

```

```

label10.Caption:=Inttostr(sumgencredit);

```

```

label11.Caption:=floattostr(sumgengrade);

```

```

// end of cgpa....

```

```

with frm_main.ADOQuery2 do    //Find if already registered

```



```

begin
  sql.Clear;
  sql.Add('SELECT c.coursecode FROM transcript as t inner join course as c on
t.courseId=c.courseId where t.StudentId="'+edit1.Text+'" and t.grade is null');
  open;
end;

```

```

if frm_main.DataSource8.DataSet.RecordCount=0 then
  begin
    messagedlg('This Student Has Not Been Registered Yet, Use Registration Menu To
Make Course Registration ..!',mtwarning,[mbok],0);
    Edit1.Clear;
    Edit2.Clear;
    Edit3.Clear;
    Edit4.Clear;
    Edit5.Clear;
    Edit6.Clear;
    Listbox1.Items.Clear;
    listbox2.Items.Clear;
    edit1.ReadOnly:=false;
    edit1.SetFocus;
    exit;
  end
else
  begin
    with frm_main.DataSource8 do
      begin
        dataset.First;
        while not(dataset.Eof) do
          begin
            listbox2.Items.Add(' '+dataset.FieldValues['Coursecode']);
            dataset.Next;
          end;
        end;
      end;

      with frm_main.ADOQuery2 do      //Find failed courses
        begin
          sql.Clear;
          sql.Add('SELECT c.coursecode,t.grade,t.courseId FROM course as c inner join
transcript as t');
          sql.add('on c.courseId=t.courseId WHERE StudentId="'+edit1.Text+'"'); // and ((t.grade
is null) or(t.grade="FF") or');
          // sql.Add('(t.grade="FD"))');
          open;
        end;

        frm_main.DataSource8.DataSet.first;
        while not(frm_main.DataSource8.DataSet.eof) do
          begin

```

```

if ((frm_main.DataSource8.DataSet.FieldValues['grade']='FF') or
(frm_main.DataSource8.DataSet.FieldValues['grade']='FD')) then
begin
label15.Caption:=inttostr(frm_main.DataSource8.DataSet.FieldValues['courseId']);
with frm_main.ADOQuery5 do      //Find if passed
begin
sql.Clear;
sql.Add('SELECT grade FROM transcript WHERE StudentId="'+edit1.Text+'" and
courseId='+label15.caption+' order by gradedate desc');
open;
end;

if
((frm_main.DataSource11.DataSet.FieldValues['grade']='FF')OR(frm_main.DataSource11.DataSet.FieldValues['grade']='FD')) THEN
begin
listbox1.Font.Color:=clred;
listbox1.Items.Add(' '+frm_main.DataSource8.DataSet.FieldValues['coursecode']+' --
--- >'+frm_main.DataSource8.DataSet.FieldValues['grade']);
frm_main.DataSource8.DataSet.next;
end
else
begin
frm_main.DataSource8.DataSet.next;

end;
end
else
begin
frm_main.DataSource8.DataSet.Next;
end;

end;
with frm_main.ADOQuery2 do      //find unregistered courses
begin
sql.Clear;
sql.Add('SELECT coursecode FROM course ');
sql.add('WHERE courseId not in (SELECT c.courseID FROM course as c inner join
transcript as t');
sql.Add('on c.courseId=t.courseId WHERE StudentId="'+edit1.Text+'"'));// and ((t.grade
is null) or(t.grade="FF") or(t.grade="FD"))));
open;
end;

frm_main.DataSource8.DataSet.first;
while not(frm_main.DataSource8.DataSet.eof) do
begin
listbox1.Font.Color:=clblack;
listbox1.Items.Add(' '+frm_main.DataSource8.DataSet.FieldValues['coursecode']);
frm_main.DataSource8.DataSet.next;

```

```

end ;
with frm_main.ADOQuery2 do      //Already passed courses
begin
  sql.Clear;
  sql.Add('SELECT c.coursecode,t.grade FROM course as c inner join transcript as t');
  sql.add('on c.courseId=t.courseId WHERE StudentId="'+edit1.Text+'"'); // and ((t.grade
is null) or(t.grade="FF") or');
//   sql.Add('(t.grade="FD"))');
  open;
end;

frm_main.DataSource8.DataSet.first;
while not(frm_main.DataSource8.DataSet.eof) do
begin
  if
not((frm_main.DataSource8.DataSet.FieldValues['grade']=null)or(frm_main.DataSource8.Dat
aSet.FieldValues['grade']='FF') or
(frm_main.DataSource8.DataSet.FieldValues['grade']='FD')) then
  begin
    listbox1.Font.Color:=clred;
    listbox1.Items.Add(' '+frm_main.DataSource8.DataSet.FieldValues['coursecode']+' ----
- >'+frm_main.DataSource8.DataSet.FieldValues['grade']+' --- PASSED');
    frm_main.DataSource8.DataSet.next;
  end
  else
  begin

    frm_main.DataSource8.DataSet.Next;
  end;
end;

end
else
begin
  messagedlg('Student ID is not Found !!',mtinformation,[mbok],0);
  edit1.SetFocus;
end;

end;

end;

procedure Tfrm_adddrop.Button2Click(Sender: TObject);

begin
  if listbox2.Items.Count<7 then
  begin

    listbox2.Items.Add(' '+listbox1.Items.ValueFromIndex[listbox1.itemindex]); //
listbox1..SelCount.ValueFromIndex[listbox1.Items.)

```

```

listbox1.Items.Delete(listbox1.ItemIndex);
end
else
begin
messagedlg('You can not register any more course !!!',mtwarning,[mbok],0);
button5.setfocus;
end;
end;

```

```

procedure Tfrm_adddrop.Button3Click(Sender: TObject);
begin
if listbox2.ItemIndex>=0 then
begin
listbox1.Items.Add(' '+listbox2.Items.ValueFromIndex[listbox2.itemindex]);//
listbox1..SelCount.ValueFromIndex[listbox1.Items.)
listbox2.Items.Delete(listbox2.ItemIndex);
end;
end;

```

```

procedure Tfrm_adddrop.Button5Click(Sender: TObject);
begin
Edit1.Clear;
Edit2.Clear;
Edit3.Clear;
Edit4.Clear;
Edit5.Clear;
Edit6.Clear;
Listbox1.Items.Clear;
listbox2.Items.Clear;
edit1.ReadOnly:=false;
edit1.SetFocus;
end;

```

```

procedure Tfrm_adddrop.FormShow(Sender: TObject);
begin
Edit1.Clear;
Edit2.Clear;
Edit3.Clear;
Edit4.Clear;
Edit5.Clear;
Edit6.Clear;
Listbox1.Items.Clear;
listbox2.Items.Clear;
edit1.ReadOnly:=false;
edit1.SetFocus;
end;

```

```

procedure Tfrm_adddrop.Button4Click(Sender: TObject);
var i,t:integer;
var a:string;

```

```

begin
with frm_main.ADOQuery2 do      //Find if already registered
begin
sql.Clear;
sql.Add('SELECT c.coursecode FROM transcript as t inner join course as c on
t.courseId=c.courseId where t.StudentId="'+edit1.Text+'" and t.grade is null and t.dropdate is
null order by t.courseregdate desc');

open;
end;
a:=' ';

frm_main.DataSource8.DataSet.First;
while not (frm_main.DataSource8.DataSet.Eof) do
begin
i:=0;
t:=0;
while (i<=(listbox2.Count-1)) do
begin
if
(frm_main.DataSource8.DataSet.FieldValues['coursecode']=leftstr(listbox2.Items.ValueFromIndex[i],ansiindexstr(listbox2.Items.ValueFromIndex[i],a))) then
begin
frm_main.DataSource8.DataSet.Edit;
frm_main.DataSource8.DataSet.FieldValues['ProcessDate']:=date;
frm_main.DataSource8.DataSet.FieldValues['ProcessUser']:=username;
frm_main.DataSource8.DataSet.Post;
t:=1;
i:=i+1;          //unchanged course
end
else
begin
i:=i+1;
end;
end; //end of 2nd while

if t=0 then      //dropped course
begin
frm_main.DataSource8.DataSet.Edit;
frm_main.DataSource8.DataSet.FieldValues['ProcessDate']:=date;
frm_main.DataSource8.DataSet.FieldValues['ProcessUser']:=username;
frm_main.DataSource8.DataSet.FieldValues['dropdate']:=date;
frm_main.DataSource8.DataSet.Post;

end;
frm_main.DataSource8.DataSet.Next;

end; // end of while

//////////7
i:=0;

```



```

frm_main.DataSource8.DataSet.First;
while (i<=(listbox2.Count-1)) do
begin
t:=0;
frm_main.DataSource8.DataSet.First;
while not (frm_main.DataSource8.DataSet.Eof) do
begin
if
(frm_main.DataSource8.DataSet.FieldValues['coursecode']=leftstr(listbox2.Items.ValueFromIndex[i],ansiindexstr(listbox2.Items.ValueFromIndex[i],a))) then
begin
frm_main.DataSource8.DataSet.Next;
t:=1;           //unchanged course
end
else
begin
frm_main.DataSource8.DataSet.Next;
end;
end; //end of 2nd while

```

```

if t=0 then    //add course
begin
frm_main.DataSource2.DataSet.Edit;
frm_main.DataSource2.DataSet.Append;
frm_main.DataSource8.DataSet.FieldValues['StudentId']:=edit1.Text;
frm_main.DataSource8.DataSet.FieldValues['courseregdate']:=date;
frm_main.DataSource8.DataSet.FieldValues['adddate']:=date;
frm_main.DataSource8.DataSet.FieldValues['ProcessUser']:=username;
frm_main.DataSource8.DataSet.FieldValues['processdate']:=date;

```

```

with frm_main.ADOQuery4 do    //Find courseID
begin
sql.Clear;
sql.Add('SELECT courseId FROM course where
coursecode="'+leftstr(listbox2.Items.ValueFromIndex[i],ansiindexstr(listbox2.Items.ValueFromIndex[i],a))+"'');
open;
end;

```

```

frm_main.DataSource8.DataSet.FieldValues['courseId']:=frm_main.DataSource10.DataSet.FieldValues['courseId'];
frm_main.DataSource8.DataSet.post;
end;
frm_main.DataSource8.DataSet.Next;
end; // end of while
end;

```

end.

unit Unit30;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;

type

TForm30 = class(TForm)

Label1: TLabel;

ADOTable1: TADOTable;

DataSource1: TDataSource;

DBGrid1: TDBGrid;

Button1: TButton;

Button2: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form30: TForm30;

implementation

uses Unit16,unit26;

{ \$R *.dfm }

procedure TForm30.Button1Click(Sender: TObject);

begin

Form30.Print;

end;

procedure TForm30.Button2Click(Sender: TObject);

begin

Form30.Close;

end;

end.

unit Unit31;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;

type

TForm31 = class(TForm)

Label1: TLabel;

ADOTable1: TADOTable;

DataSource1: TDataSource;

DBGrid1: TDBGrid;

Button1: TButton;

Button2: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form31: TForm31;

implementation

uses Unit16,unit26;

{ \$R *.dfm }

procedure TForm31.Button1Click(Sender: TObject);

begin

form31.Print;

end;

procedure TForm31.Button2Click(Sender: TObject);

begin

form31.Close;

end;

end.

unit Unit32;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;

type

TForm32 = class(TForm)

Label1: TLabel;

ADOTable1: TADOTable;

DataSource1: TDataSource;

DBGrid1: TDBGrid;

Button1: TButton;

Button2: TButton;

procedure Button1Click(Sender: TObject);

```

    procedure Button2Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form32: TForm32;
implementation
    uses Unit16,unit26;
    {$R *.dfm}
    procedure TForm32.Button1Click(Sender: TObject);
    begin
        form32.Print;
    end;
    procedure TForm32.Button2Click(Sender: TObject);
    begin
        form32.Close;
    end;
end.

```

unit Unit33;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB;

type

```

TForm33 = class(TForm)
    ComboBox1: TComboBox;
    ADOQuery1: TADOQuery;
    ADOQuery2: TADOQuery;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    DBGrid1: TDBGrid;
    Label1: TLabel;
    Label2: TLabel;
    Button1: TButton;
    Button2: TButton;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure ComboBox1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
    { Private declarations }
public
    { Public declarations }
end;

```

```
var  
  Form33: TForm33;
```

```
implementation
```

```
uses Unit23, Unit16;
```

```
{SR *.dfm}
```

```
procedure TForm33.FormCreate(Sender: TObject);  
begin
```

```
  with ADOQuery1 do
```

```
    begin
```

```
      SQL.Clear;
```

```
      sql.add('SELECT StudentId FROM Transcript order by StudentId');
```

```
      open;
```

```
    end;
```

```
    DataSource1.DataSet.First;
```

```
    while not (DataSource1.DataSet.Eof) do
```

```
    begin
```

```
      combobox1.Items.add(DataSource1.DataSet.FieldValues['StudentId']);
```

```
      DataSource1.DataSet.Next;
```

```
    end;
```

```
end;
```

```
procedure TForm33.Button1Click(Sender: TObject);
```

```
begin
```

```
  Form33.print;
```

```
end;
```

```
procedure TForm33.Button2Click(Sender: TObject);
```

```
begin
```

```
  Form33.Close;
```

```
end;
```

```
procedure TForm33.ComboBox1Click(Sender: TObject);
```

```
begin
```

```
  with ADOQuery2 do
```

```
    begin
```

```
      SQL.Clear;
```

```
      sql.add('SELECT t.*,c.* FROM transcript as t inner join course as c on t.CourseId =  
c.CourseId where t.studentID="'+combobox1.Text+'" order by t.studentId');
```

```
      open;
```

```
    end;
```

```
end;
```



```
procedure TForm33.FormClose(Sender: TObject;  
  var Action: TCloseAction);  
begin  
  rep_main.show;  
end;  
end.
```

unit Unit40;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls, Buttons;

type

```
Tfrm_about = class(TForm)  
  Image1: TImage;  
  Panel1: TPanel;  
  SpeedButton1: TSpeedButton;  
  procedure SpeedButton1Click(Sender: TObject);  
  procedure FormClose(Sender: TObject; var Action: TCloseAction);  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;
```

var

```
frm_about: Tfrm_about;
```

implementation

uses Unit1,unit26;

```
{ $R *.dfm }
```

```
procedure Tfrm_about.SpeedButton1Click(Sender: TObject);  
begin  
  frm_about.Close;  
end;
```

```
procedure Tfrm_about.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
  frm_main.Show;  
end;  
  
end.
```