TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor's Thesis:
Building Web Application for Mahabad University Graduate Affairs
Afsaneh Nezami Savojbolaghi

Turku University of Applied Sciences
Degree Programme in Business Information Technology
2010

TURKU UVNIVERSITY OF   ABSTRACT

APPLIED SCINCES

Degree Programme : Business Information Technology

Author: Afsaneh Nezami Savojbolaghi

Title: Building web application for Mahabad University Graduate Affairs

Specialization line: Business Information systems Management	Instructor: Minna-Kristiina Paakki

Date: 2010	Total number of pages: 34+9 appendices

This thesis deals with the development of a web application using c# programming language and SQL server database management system. The project was made for Graduate Affairs in Mahabad university which was going to implement it to the student affairs at the university. The web application has four user interfaces for the employees that include the field of study, degree, gender and students' origin i.e. local or non local students.

The main objective of the study was to build a useful database for student statistics. The database was manually developed before it was transferred into proper database program.

The final result was a functional and user friendly application that an employee can create, update and delete data that is stored in the database.

Keywords: SQL, C#, graduate affairs

Deposit at:
Turku University of Applied Sciences Library, Salo

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The task of measuring the benefit of IT is an important task and very complex. Many studies have been done on the concept of the impacts on organizational performance and productivity. Results obtained on many of such studies have been varied. There are divergent opinions or whether or not information technology provides organizational or financial improvement for companies. Currently the new development in IT system is wide spread and could be quite expensive, making this concept even more important. Every organization desires to enjoy for the benefits derivable from these new IT system because a successful implementation of It system for business accomplishment and continuance. (Max Aro 2005, 1)

Development of the technology and database system has changed the way many organizations operate and individuals work over the last few years. The introduction of database system is the most important development in the field of software engineering .The database is now powerful and more intuitive to use. This system led to users creating applications without the necessary knowledge to produce an effective system. (Thomas Connolly &Carolyn Begg, 2004, 3)

This study focuses on the organizational process view and process improvements enabled by IT for large university with many students, that they need to have a student database or student information systems, which handles all the statistics information for issues connected to them. The development was made for graduate affairs which is a part of Mahabad University in northwestern of Iran.

After discussing with the CEO of graduate affairs in university the need for her organization to have a database system for student's statistics, including: Number of males and females; Number of undergraduates, MA students, and P.H.D students; Number of majors; and the issuance places (indigenous and non-indigenous students: Local and non Local).

Then I discussed with her about the installing server on their computer. She answered me that it is Structure query language (SQL) Server 2005. It was better for them to use programming language C# which support by Microsoft visual Studio because Microsoft Visual Studio includes native support data programming with Microsoft SQL Server and it can be used to write and debug code to be executed by SQL CLR. To replace a traditional system we made an information database project. These issues are described in following chapters. Chapter two focuses on database system development and Chapter three goes through university background and its information needs, chapter four of this study will cover an overview of the database system management selection while chapter five outlines the development process.

.

2 DATABASE SYSTEM DEVELOPMENT

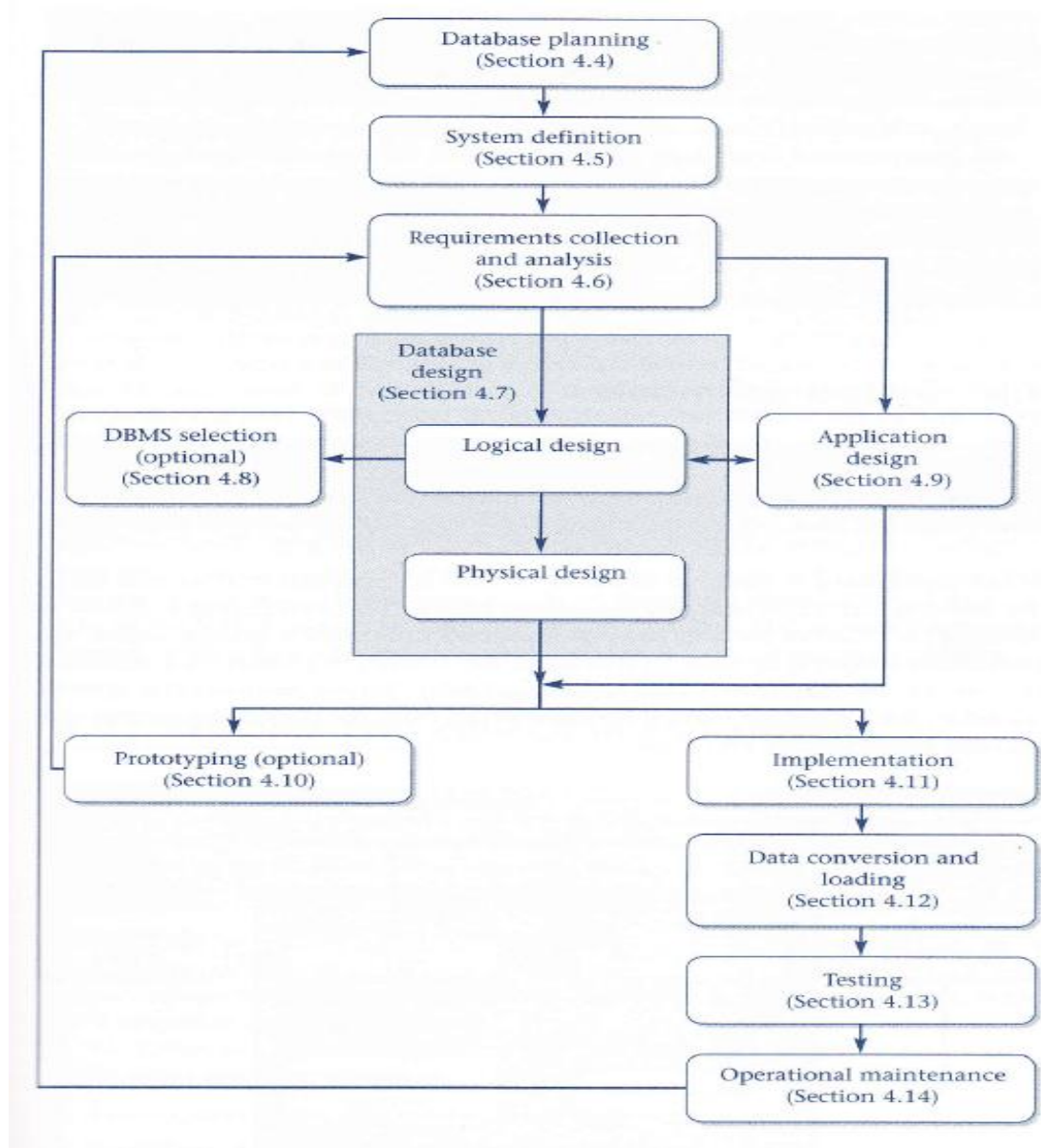I will illustrate the stage of database system development lifecycle. (Thomas Connolly &Carolyn Begg, 2004, 81).

Figure 1: The stage of the database system development lifecycle

In the next section I will describe these stages

2.1 Main stages of system development of database

In the following section I will define the main stages:

2.1.1 System definition

The scope and boundary of the system is very important to identify. It's essential to know how it interfaces with organization's system parts.

2.1.2 Requirements collection and analysis

Using the information of the organization for a new database system and description of the requirements and details of the data is the process of this section. In the our new system for case university a new system helps them for a reliable and fast system for student information management. In order to appreciate the requirements for an student information system it is necessary to know some information that I explain more about it in part three.

2.1.3 DBMS

''The DBMS is the software that interacts with the users, application programs, and the database. Among other things the DBMS allows users to insert, update, delete and retrieve data from the database. Having a central repository for all data and data description allows the DBMS to provide a general inquiry facility to this data, called a query language''. (Thomas Connolly &Carolyn Begg, 2004, 8)

I will discuss the Structure Query Language (SQL) as a main query language for Database management system in our case.

2.1.4 Application design

 For the process of database system we use the user interface and application design.

To achieve how we can design required functionality we should design user interface in the system. I will explain how to add, align and position controls in the Graduate Affairs application to design and modify the user interface.

2.1.5 Prototyping

A prototyping is a working model to identify the system features and function which allows the users evaluate how the final system will work.

2.1.6 Implementation

Creating the database and application design. In our case the application programs are implemented using the c# programming language which is simple, object oriented and powerful which include support with Microsoft SQL Server.

3 DEVELOPMENT PROJECT DEFENITON

3.1 Brief overview of Mahabad University:

Mahabad University was established in 1986 in an excellent location in Mahabad city in northwestern Iran. The university has 4 faculties include faculty of agriculture, engineering, sciences and literature and natural sciences. It is one of the most attractive universities. They offer education that develops working life and entrepreneurship, research and development services (R&D) and holistic development of organizations. At Mahabad University you can study for Bachelor's or Master's Degree in several different educational fields. The project was made for Graduate Affairs in Mahabad University which was going to implement for the student and up to now has been done traditionally.

Figure 2: Mahabad university web Site. http://www.iau-mahabad.ac.ir

3.2 The need for a student management system

Previously the statistic student information created in spreadsheets. Also due to the size of the SSI, these files where often large and the secretary had difficulties retrieving them for reference or updating them. The Secretary spent plenty of time on information to find out the project real situation. And anyone could access in this kind of file on local workstation and it's not secure.

3.3 The requirements for a new system

A new system helps them for a reliable and fast system for student information management .In order to appreciate the requirements for an student information system for SSI it is necessary to know some information. Including: Number of males and females; Number of undergraduates, MA students, and P.H.D students; Number of majors; and the issuance places (indigenous and non-indigenous students: Local and non Local). This project is an information database to replace a traditional system.

Another requirement was installing server on their computer, that it is Structure query language (SQL) Server 2005. It was better for University to use programming language C# which support by Microsoft visual Studio because Microsoft Visual Studio includes native support data programming with Microsoft SQL Server and it can be used to write and debug code to be executed by SQL CLR.

4 DATABASE MANAGEMENT SYSTEM SELECTIONS

To selection a DBMS a simple way is to check off requirements and features. There is an opportunity to certify that the selection process is well arrangement and organization get real benefits of the system in selecting a new database product. (Thomas Connolly &Carolyn Begg, 2004, 88)

4.1 SQL Server Management Studio

For accessing, configuring, managing, administering, and developing all components of SQL Server there is a graphical user interface tool namely SQL Server Management Studio. It is included group of graphical tools with a number of rich script editors to provide access to SQL Server to developers and administrators of all skill levels.Object Explorer is A central feature of SQL Server Management Studio is the, which users can browse, select, and act within the server.

Microsoft has also introduced a graphical configuration tool called SQL Server Management Studio Express (SSMSE) for SQL Server Express. [online, referenced (10.02.2010)

4.2 Microsoft SQL Server 2005

´´Microsoft SQL Server 2005 is a comprehensive, integrated data management and analysis software that enables organizations to reliably manage mission-critical information and confidently run today's increasingly complex business applications. SQL Server 2005 allows companies to gain greater insight from their business information and achieve faster results for a competitive advantage´´. [online, referenced (10.02.2010)

4.3 Create a New Registered Server:

In the New Server Registration dialog box, I have, typed the name of the server that in our case server name isKurdish-de735d4 and server type is Data base Engine. Under Authentication, I can accept the default of Windows Authentication, or click SQLServer Authentication, and complete the User name and Password box.
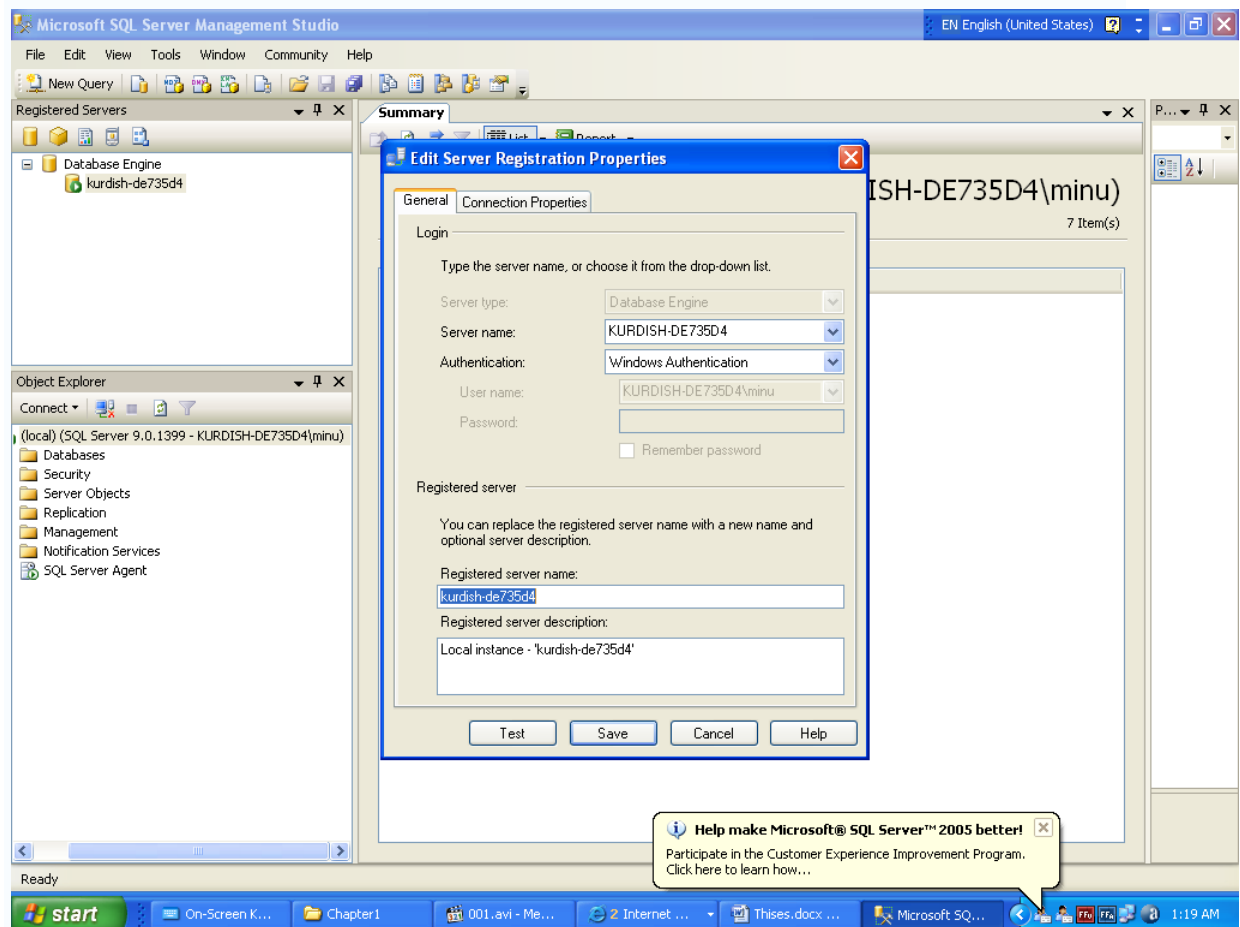

Figure3: Create a New Registered Serve

4.4 Connected to server

The Connect to Server dialog box on the Object Explorer toolbar we should Click Connect, and click the type of server. In the Server Name box, I typed the name of the SQL Server.
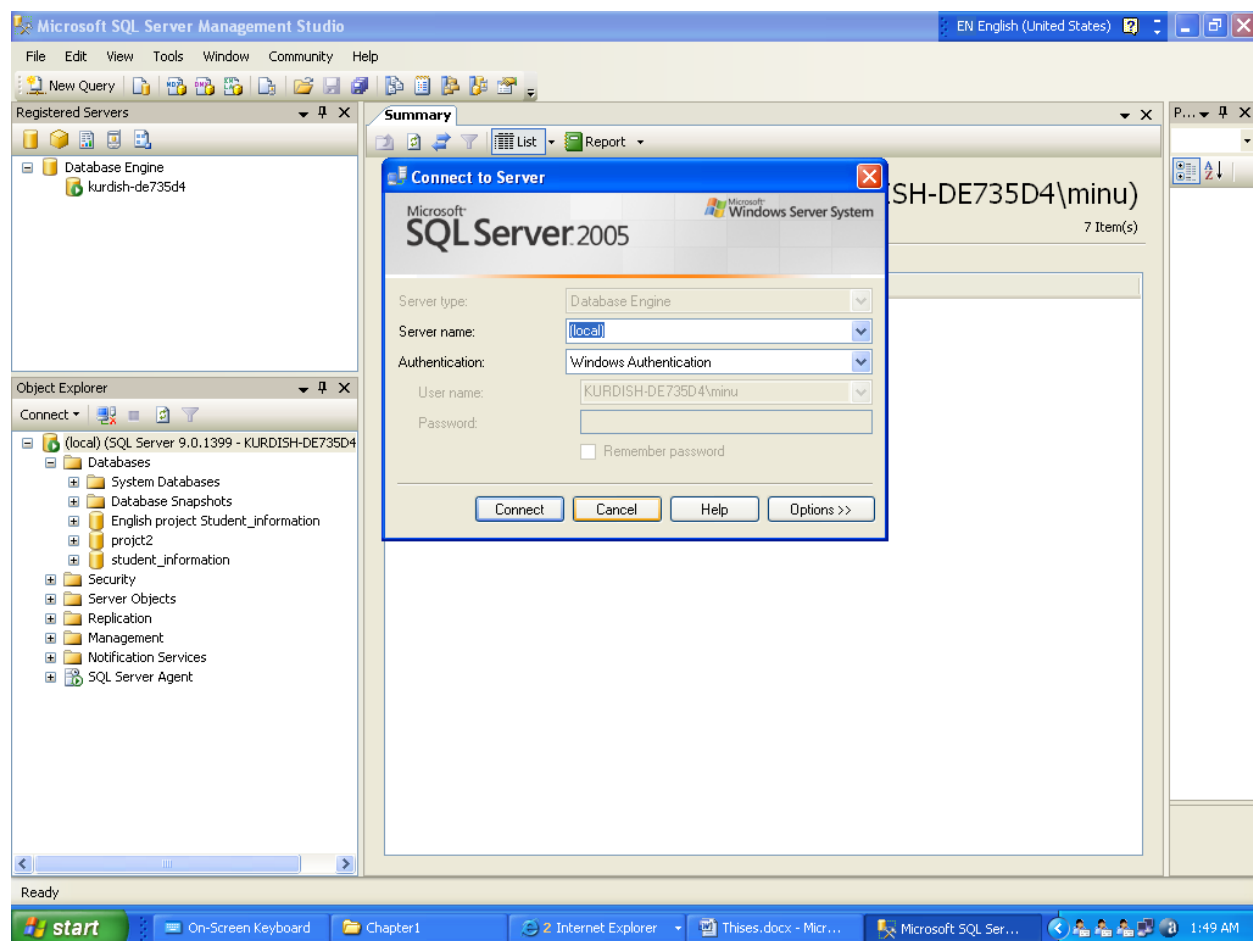

Figure4:Connected to server

5 APPILICATION FOR GRADUATE AFFAIRS IN MAHABAD UNIVERSITY

The application is built by sql server relational database management for storing the database. It also uses the C# language.

Visual Studio supports Visual C# with a full-featured code editor, compiler, project templates, designers, code wizards, a powerful and easy-to-use debugger, and other tools. The .NET Framework class library provides access to many operating system services and other useful, well-designed classes that speed up the development cycle significantly. (Microsoft visual studio [online, referenced 27.04.2010]).

SQL Server Management Studio is a GUI tool included with SQL Server 2005 and later for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. (SQL server [online, referenced 10.02.2010]).

Graduate Affairs application has administration users that give clear information about the student. Users can see and check student information, make searches.

The application enables employees to create, update and delete data that is stored in the database.

5.1 The GUI design application

For design and modify the user interface of my Windows Forms applications, I will need to add, align, and position controls. Controls are objects that are contained within form objects. Each type of control has its own set of properties, methods, and events that make it suitable for a particular purpose.

According to Figure 4 Graduate Affairs web application is separated by two parts. The headers and footer. There are Menu strip and tool strip controls in header and status strip controls in footer.

The **Menu Strip** control represents the container for the menu structure of a form. To add ToolStripMenuItem objects to the Menu Strip that represents the individual menu commands in the menu structure. Each ToolStripMenuItem can be a command for application or a parent menu for other submenu items.(Menu Strip[online, referenced 2010]).

The Menu strip has three menus, namely file, View, Information.

File menu has one sub menu, namely Exit. View menu has two sub menus: Tool strip and Status strip and Information menu has four sub menus: Local student, Non local student, Excellent student and Accept student in upper.

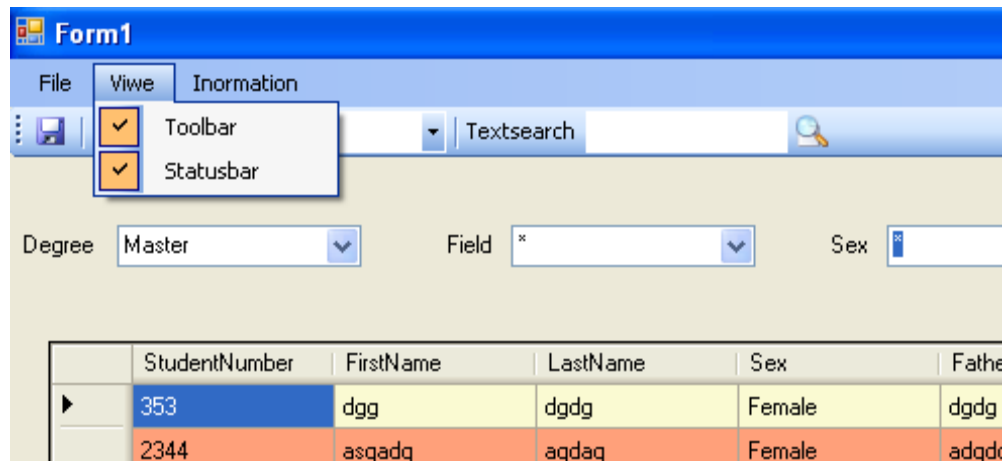

Figure 5: Graduate Affairs Application
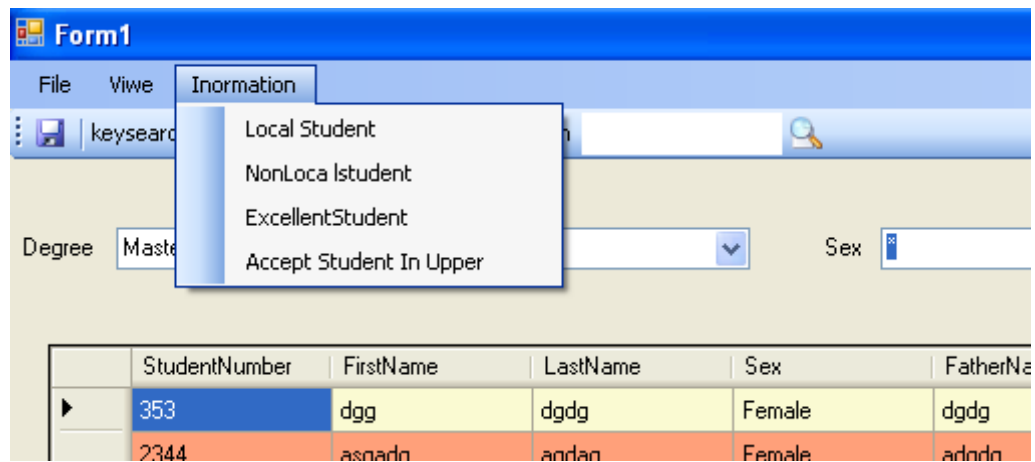
Figure 6: View Menu

Figure 7: Information Menu

- **Tool strip control**: Use Tool Strip and its associated classes in new Windows Forms applications to create toolbars that can have a Windows XP, Office, Internet Explorer, or custom appearance and behavior all with or without themes, and with support for overflow and run-time item reordering. Tool Strip controls also offer a rich design-time experience that includes in-place activation and editing, custom layout, and sharing of horizontal or vertical space within a specified. .(Tool Strip[online, referenced 2010]).

I use the following items of tool strip to design of application:

- Tool strip button that supports both text and images
- Tool strip separator represents a line used to group items of a tool strip or the drop-down items of a menu strip or context menu strip or other tool strip dropdown control
- Tool strip text box it's a text box in a tool strip that allows the user to enter text
- Tool strip combo box a tool strip combo box that is properly rendered in a tool strip. Tool strip label represents a no selectable tool strip Item that renders text and images and can display hyperlinks The tool strip of Graduated Affairs application has a tool strip button that views the **save image** and a **line** that use to separate. The **Key search** label and combo box item that contains the Student information, Student No, First Name, Last Name, Sex, Father Name, ID Number, Date Of Birth, Year Of Admission, Birth Place, Field, Degree, Average, Accept in Upper, Year of Graduation.

Text search label and a **text box** with a **search image** are tool strip items that I use for design.

Figure 8: Key search combo box

Status Strip control displays information about an object being viewed on a Form, the object's components, or contextual information that relates to that object's operation within your application .(Status Strip[online, referenced 2010]).

According to Figure 8 when I choose the **Computer** from **Field check box** Status strip displays **Filter based on field.**

Figure 9: Status Strip

In the application there is four label to descriptive text for the control namely **Degree, Field, Sex, issues Place.** The four **Combo box** to display unbound items. First combo box displays Degree label and shows four items namely None, Bachelor, Master and PHD degrees. Second combo box display Field label and shows all the University fields. Third combo box display Sex label and shows three items namely None, Female, male. Forth combo box display Issues place and shows None, Local and Nonlocal student's items.

Figure 10: Degree Combo Box

§*Figure 11: Field Combo Box*

Figure 12: Sex Combo Box

Figure 13: Issues place Combo Box

5.2 Graduate Affairs' application and Database

Graduate Affairs' application uses the C# and SQL S-server combination. In order to run the application basic procedure are needed. First, a connection to the database server should be established. After establishing a connection, specific database should be selected to work with. Once a connection is established and a database selected, then it is possible to work with tables within them buy using SQL programming language to issue various commands. Data can be retrieved, added, modified, or deleted. Any changes can be made to the selected database server. Furthermore all the above examples are taken from the Graduated Affairs' application in order to explain the relational of the C# and SQL server.

5.3 C# DataGridView

As you can see in the application to display rows and columns of data i use the Data Grid View and add tabular data from data sources. The data display in theDataGridView control will come from a data source. To Binding data to theDataGridView control it is simple setting the Data Source property. Data grid view shows all the student information data namely: Student No, First Name, Last Name, Sex, Father Name, ID Number, Date of Birth, Year Of Admission, Birth Place, Field, Degree, Average, Accept in Upper, Year of Graduation. On the right part of Window you will see the Visual Studio Toolbox. It contains the icons and controls you can add.
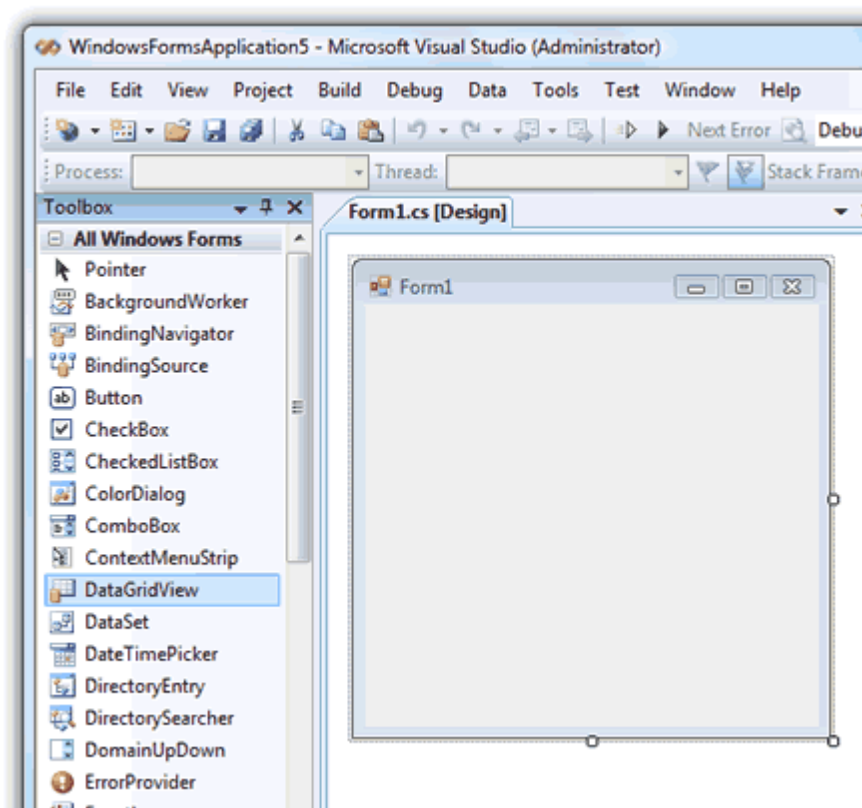
Figure 14: windows forms

5.3.1 Using databases

To generate a simple SDF file for the rest of the program to interact with. We will use Visual Studio for this part. And then Add New Data Source and by Visual Studio wizard add a new database.

5.3.2 Connection to database

Connection to database by attached the file that I created in specific path   in SQL server namely' English project student_information.mdf'.
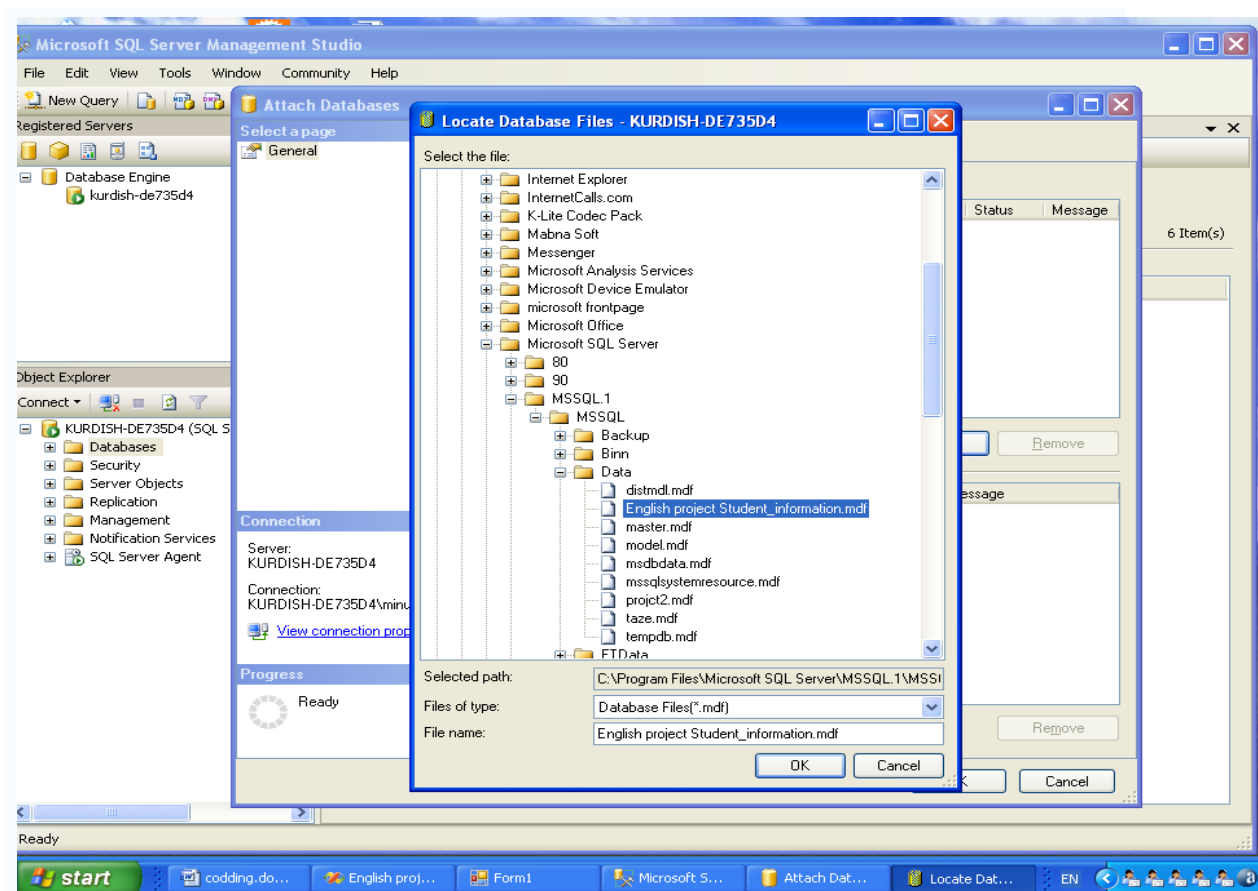
Figure 15: Connection to database

5.3.3 Create table

The database must have table containing row data .Tables are objects that contain all the data in SQL Server databases. Each table represents a type of object that is meaningful to its users. (Tables [online, referenced 2010]).

For example, in the **student Information** database there are tables that contain data about first name, last-name, Sex, Father Name, ID Number, Date Of Birth, Year Of Admission, Birth Place, Field, Degree, Average, Accept in Upper, Year of Graduation.
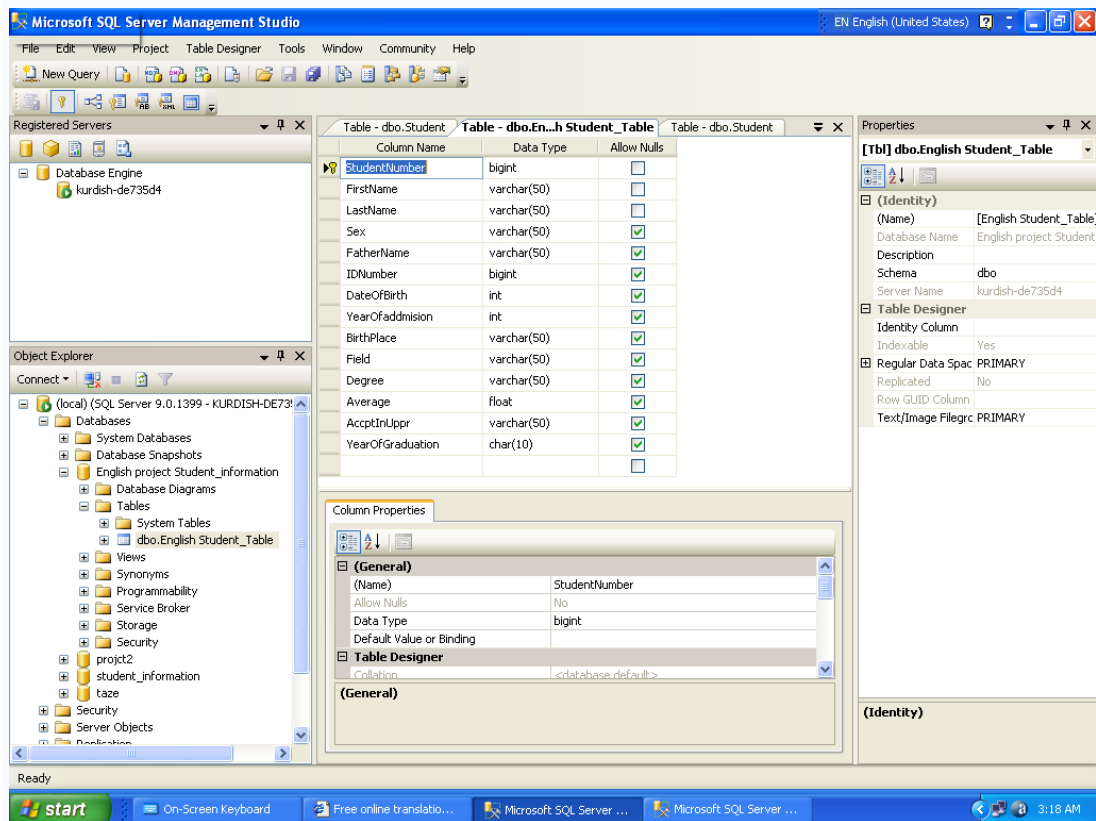
Figure 16: create a table

5.3.4 Add data adapter code

In C# code, it must have a connection to database first, and then create a Data Adapter. The data adapter can configure using a wizard, which makes it easy to create the SQL statements for data access. (Data adapter [online, referenced 2010]).

Table1: Add data adapter code

```csharp
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace English_projct_Student_info

{

public partial class btnCancle : Form

    {

        public btnCancle()

        {

            InitializeComponent();

        }

private void Form1_Load(object sender, EventArgs e)

        {

      // TODO: This line of code loads data into the
'english_project_Student_informationDataSet1.English_Student_Table'
table. You can move, or remove it, as needed.


this.english_Student_TableTableAdapter1.Fill(this.english_project_St
udent_informationDataSet1.English_Student_Table);

          // TODO: This line of code loads data into the
'english_project_Student_informationDataSet.English_Student_Table'
table. You can move, or remove it, as needed.}
```

5.4 Query type in the database

Any valid and complete SQL statement can be passed as parameters to this function. Graduated Affairs' application uses four types of queries to make up SQL data manipulation. These data manipulate DataGridViewtion languages are SELECTE, INSERT, UPDATE, and DELETE should make it in store procedure stored in student information database

5.4.1 Store procedure

Stored procedure is a subroutine available to applications accessing a relational database system. Stored procedures are actually stored in the database data dictionary.(Store Procedure[online, referenced 2010]

To create, update, delete and save data in web application we should make it in store procedure stored in student information database.
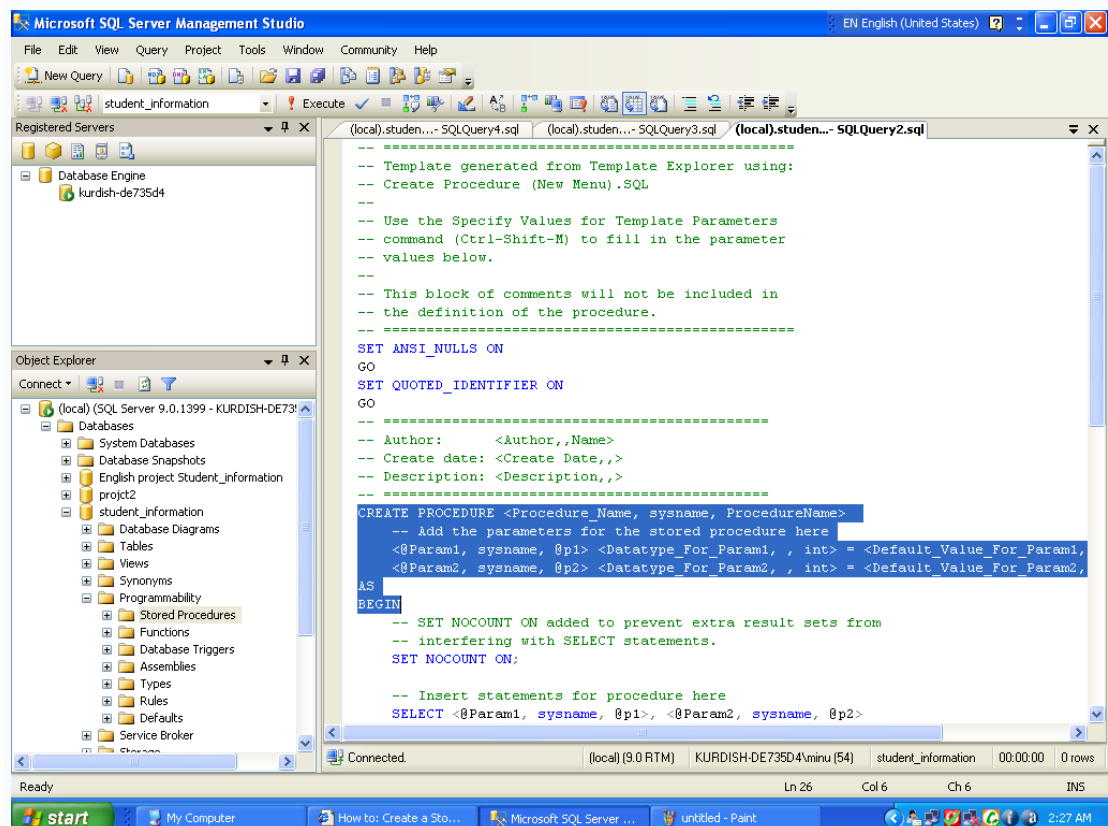
Figure17 : Stored procedure
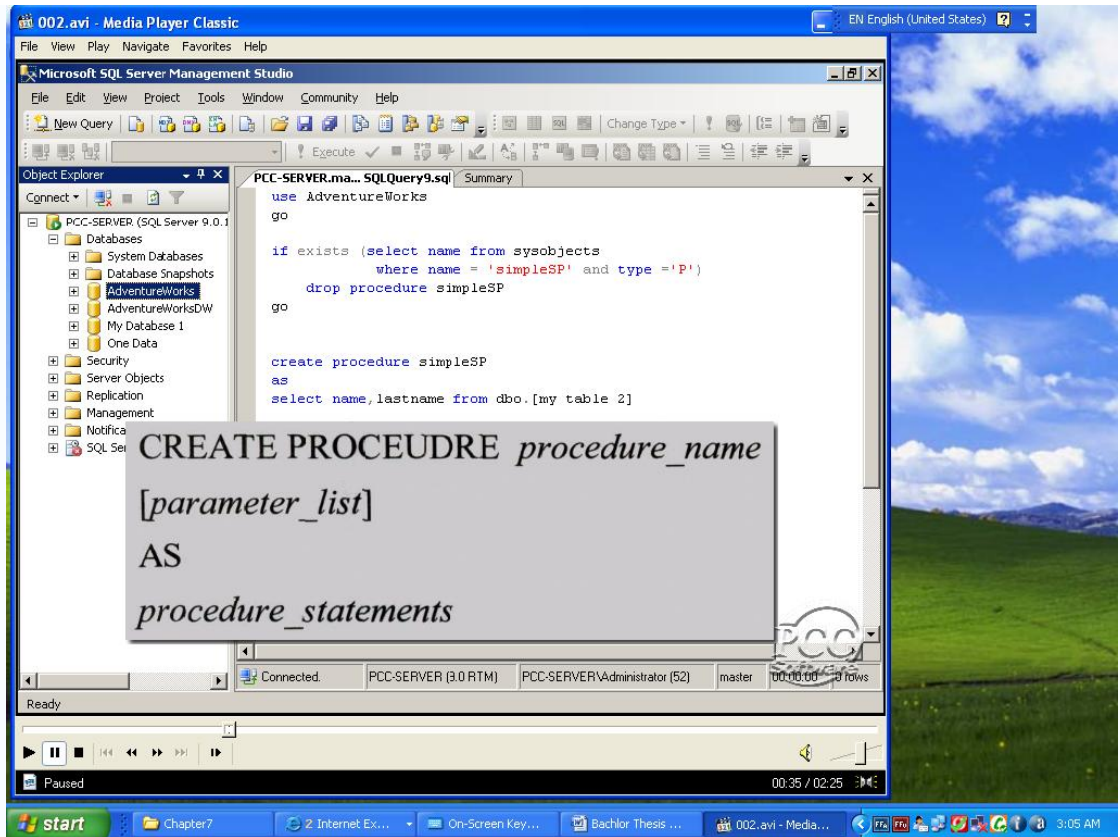
Figure18: Create stored procedure

The following examples describe how these query used in the application.

A. Data selection

Table 2 shows query example retrieves data from Student table within the variable database name English_projct_Student_info.

Table 2: Example of data selecting

```
BEGIN

      SELECT StudentNumber,FirstName,LastName,sex,FatherName,ShSh,

BirthYear,BirthPlace,YearOfAccept,Field,Section,Average,AcceptInUpper,YearOfGr
aduate
    from Student
END
```

Table 3: Example of deletion data

```
ALTER PROCEDURE [dbo].[DeleteStudentData]

@StudentNumber bigint
AS
BEGIN

      SET NOCOUNT ON;
Delete From Student Where StudentNumber=@StudentNumber
END
```

Table 3 shows insert query examples statement describe to perform adding new data into Student table.

Table 4: Data insertion

```
ALTER PROCEDURE [dbo].[InsertStudentData]
      @StudentNumber bigint,
@FirstName nvarchar(50),
@LastName nvarchar(50),
@sex nvarchar(50),
@FatherName nvarchar(50),
@ShSh bigint,
@BirthYear int,
@BirthPlace nvarchar(50),
@YearOfAccept int,
@Field nvarchar(50),
@Section nvarchar(50),
@Average Float,
@AcceptInUpper nvarchar(50),
@YearOfGraduate int
AS
BEGIN

    SET NOCOUNT ON;
    insert into Student values(@StudentNumber,@FirstName,@LastName,@sex,@FatherName,
            @BirthYear,@BirthPlace,@YearOfAccept,@Field,@Section,@Average,@Acc
END
```

Table 5: Updating data in the database

```
ALTER PROCEDURE [dbo].[UpdateStudentData]
@StudentNumber bigint,
@FirstName nvarchar(50),
@LastName nvarchar(50),
@sex nvarchar(50),
@FatherName nvarchar(50),
@ShSh bigint,
@BirthYear int,
@BirthPlace nvarchar(50),
@YearOfAccept int,
@Field nvarchar(50),
@Section nvarchar(50),
@Average Float,
@AcceptInUpper nvarchar(50),
@YearOfGraduate int
AS
BEGIN

.

      SET NOCOUNT ON;

   Update Student set
StudentNumber=@StudentNumber,FirstName=@FirstName,LastName=@LastName,sex=@sex,

FatherName=@FatherName,ShSh=@ShSh,BirthYear=@BirthYear,BirthPlace=@BirthPlace,
              YearOfAccept=@YearOfAccept,Field=@Field,Section=@Section,Average=@Avera
              AcceptInUpper=@AcceptInUpper,YearOfGraduate=@YearOfGraduate
```

6 CONCLUSION

The usage of data and application has advantage over traditional system but has disadvantage as well. In this section I briefly explain advantage and disadvantage of our database lifecycle model:

- **Requirements collection**

  To set convenient time for interview was hard with the university staff and sometimes they were not understood by analysts. But with the friendly environment and with structuring interview questions, I able to provide insight into the problem.

- **DBMS**

  The Structure query Language is a main query language for database management system in our case. For query processing it provides storage structure and provides backup and recovery. It reduced cost of data entry, storage, and retrieval .But as we know database system is difficult, and time-consuming to design. In the Graduate Affairs there is no many technical expertise and often the repair process took a great deal of time.

- **Application design**

  Web application has four user interfaces for the employees. That is include field of study, degree, gender and local or non local students and functional and friendly application that employee can create, update and delete data that is stored in the database. However there is disadvantage of the database approach, such as: data security. They should Improve Data Security and Storage.

- **Prototyping**

  Prototyping is creating the subset of requirements/functionality, and refining for system. An advantage is that users use the system and give you real feedback.

- **Implementation**

In our system the application programs are implemented using the c# programming language which is simple, object oriented and powerful which include support with Microsoft SQL Server. After the application was ready, we tried to install it on the computer but there were some problems, one of which was that in the graduate affair section any separate computer, on which the SQL Server Program and Visual Studio to be install was not found. I had to ask for a computer, I did but there wasn't any answer to my request. Secondly, I wanted to test the programs on computers of the graduate affair section but none of the staff there, let me. They were afraid that their computer might get out of work and the repair process took a great deal of time, their work might be delayed because there is only one person to repair for all the computers. Therefore, we tested the system on my computer then train two staff to create, edit, delete, update the system and saved the program on CD to use it and they happily accepted and thanked me.

Through working on the Gradate Affairs application I learnt a lot of how to developed application with the Sql server and C# and making a combination of sql and C# needs careful works.

List of references List of the references

Literature:

Thomas Connolly& Carolyn Begg.2004.Database Solutions.

Electronic sources:

Mahabad university web Site, [online, referenced 27.04.2010]

<URL: http://www.iau-mahabad.ac.ir>

Menu Strip, [online, referenced 2010].

<URL:http://msdn.microsoft.com/enus/library/system.windows.forms.menustrip.aspx

Microsoft SQL Server, [online, referenced 10.02.2010]

URL:http://en.wikipedia.org/wiki/SQL_Server_2005#SQL_Server_2005

Status Strip,[online, referenced 2010].

<URL:http://msdn.microsoft.com/enus/library/system.windows.forms.statusstrip.aspx

Store Procedure [online, referenced 2010].

<URL: http://en.wikipedia.org/wiki/Stored_procedure>

SQL Server Management Studio,[online, referenced 10.02.2010]

< URL:http://en.wikipedia.org/wiki/SQL_Server_Management_Studio>

Tables [online, referenced 2010].

<URL: http://msdn.microsoft.com/en-us/library/aa214359(SQL.80).aspx>

Tool Strip [online, referenced 2010].

<URL: http://msdn.microsoft.com/en-us/library/system.windows.forms.toolstrip>

Source code of data Adapter   Appendix1

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace English_projct_Student_info
{
    public partial class btnCancle : Form
    {
        public btnCancle()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'english_project_Student_informationDataSet1.English_Student_Table' table. You can
move, or remove it, as needed.

this.english_Student_TableTableAdapter1.Fill(this.english_project_Student_informationDataSet1.English_Student_Table);
            // TODO: This line of code loads data into the
'english_project_Student_informationDataSet.English_Student_Table' table. You can
move, or remove it, as needed.


        }
```

Source code of Degree, Field,sex, issuce Place and Text Search.   Appendix2

```csharp
    {

    }

    private void toolStripTextBox2_Click(object sender, EventArgs e)
    {

    }

    private void informationToolStripMenuItem_Click(object sender, EventArgs e)
    {

    }

    private void toolStripComboBox1_Click(object sender, EventArgs e)
    {

    }

    private void statusStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
    {

    }

    private void menuStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
    {

    }

    private void label1_Click(object sender, EventArgs e)
    {

    }


    private void pictureBox1_Click(object sender, EventArgs e)
    {

    }
```

```csharp
    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

    private void cmbIssusePlace_SelectedIndexChanged(object sender, EventArgs e)
    {
       if (cmbIssusePlace.Text!= "*")
       {
          if (cmbIssusePlace.Text == "Local")
             bindingSource1.Filter = "BirthPlace like '" + "Mahabad" + "'";
          else if (cmbIssusePlace.Text == "NonLocal")
             bindingSource1.Filter = "BirthPlace not like '" + "Mahabad" + "'";
       }
       else
       {
          bindingSource1.RemoveFilter();
       }

    }

    private void bindingSource1_CurrentChanged(object sender, EventArgs e)
    {

    }

    private void btnSave_Click(object sender, EventArgs e)
    {
       this.Validate();
       this.bindingSource1 .EndEdit();

this.english_Student_TableTableAdapter1.Update(english_project_Student_informati
onDataSet1.English_Student_Table);
    }

    private void toolStripMenuItem2_Click(object sender, EventArgs e)
    {
       bindingSource1.Filter = "BirthPlace like '" + "Mahabad" + "'";
       toolStripStatusLabel1.Text = "Local Student ";

    }

    private void cmbSex_SelectedIndexChanged(object sender, EventArgs e)
    { bindingSource1.Filter = "BirthPlace not like '" + "Mahabad" + "'";
```

```csharp
                if (cmbSex.Text != "*")
                {
                    bindingSource1.Filter = "Sex like '" + cmbSex.Text + "'";
                    toolStripStatusLabel1.Text = "Filter based on sex ";
                }
                else
                {
                    bindingSource1.RemoveFilter();
                    toolStripStatusLabel1.Text = "";
                }
            }

            private void cmbDegree_SelectedIndexChanged(object sender, EventArgs e)
            {
                if (cmbDegree.Text != "*")
                {
                    bindingSource1.Filter = "Degree like '" + cmbDegree.Text + "'";
                    toolStripStatusLabel1.Text = "Filter based on Degree";
                }
                else
                {
                    bindingSource1.RemoveFilter();
                    toolStripStatusLabel1.Text = "";
                }
            }

            private void cmbField_SelectedIndexChanged(object sender, EventArgs e)
            {
                if (cmbField.Text != "*")
                {
                    bindingSource1.Filter = "Field like '" + cmbField.Text + "'";
                    toolStripStatusLabel1.Text = "Filter based on field ";
                }
                else
                {
                    bindingSource1.RemoveFilter();
                    toolStripStatusLabel1.Text = "";
                }
            }




            private void toolStripComboBox1_Click_1(object sender, EventArgs e)
            {
                try
                {
                    toolStripComboBox1.Items.Clear();
                    foreach (DataGridViewColumn col in dataGridView1.Columns)
```

```csharp
                {
                    toolStripComboBox1.Items.Add(col.HeaderText);
                }
            }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void toolStripButton2_Click(object sender, EventArgs e)
    {
        try
        {
            toolStripStatusLabel1.Text = "Search ";
            if (toolStripComboBox1.Text == "")
            {
                MessageBox.Show("Enter a KeySearch", "Student",
MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
                toolStripComboBox1.Focus();
            }
        }
    }
```

Source Code for Data Grid View   Appendix3

```
            {

        else
                    int colIndex = 0;
            foreach (DataGridViewColumn col in dataGridView1.Columns)
            {
                if (col.HeaderText == toolStripComboBox1.Text)
                {
                    colIndex = col.Index;
                    break;
                }
            }
            for (int rowCount = dataGridView1.CurrentRow.Index + 1 ==
dataGridView1.Rows.Count - 1 ?
                0 : dataGridView1.CurrentRow.Index + 1; rowCount <=
dataGridView1.Rows.Count - 1; rowCount++)
            {
                if (dataGridView1.Rows[rowCount].Cells[colIndex].Value != null)
                {
                    if
(dataGridView1.Rows[rowCount].Cells[colIndex].Value.ToString().Contains(toolStri
pTextBox1.Text))
                    {
                        dataGridView1.Rows[rowCount].Cells[colIndex].Selected = true;
                        dataGridView1.CurrentCell =
dataGridView1.Rows[rowCount].Cells[colIndex];
                        break;
                    }
                }

            }
          }
        }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

    private void exitToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Close();
```

```csharp
        }

    private void dataGridView1_RowStateChanged(object sender,
DataGridViewRowStateChangedEventArgs e)
    {
        counter.Text = (dataGridView1.RowCount - 1).ToString();
        for (int row = 0; row < dataGridView1.RowCount; row++)
        {
            if (row % 2 == 0)
                dataGridView1.Rows[row].DefaultCellStyle.BackColor =
Color.LightGoldenrodYellow;
            else

 dataGridView1.Rows[row].DefaultCellStyle.BackColor =

Color.LightSalmon;
        }
    }

    private void Toolbar_Click(object sender, EventArgs e)
    {
        if (Toolbar.Checked)
        {
            Toolbar.Checked = false;
            toolStrip1.Visible = false;
        }
        else
        {
            Toolbar.Checked = true;
            toolStrip1.Visible = true;
        }
    }

    private void Statusbar_Click(object sender, EventArgs e)
    {

    }

    private void btnExit_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            if
```

```csharp
(this.english_project_Student_informationDataSet1.English_Student_Table.GetChang
es()!= null)
            {
                if (MessageBox.Show("Do you want to cancle?", "Student",
MessageBoxButtons.OKCancel,
                MessageBoxIcon.Question, MessageBoxDefaultButton.Button2) ==
DialogResult.OK)
                {

this.english_project_Student_informationDataSet1.English_Student_Table.RejectCha
nges();
                    ClearErrors();
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void ClearErrors()
    {
        try
        {
            foreach (DataGridViewRow row in dataGridView1.Rows)
            {
                foreach (DataGridViewCell cel in row.Cells)
                {
                    cel.ErrorText = "";
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void NonLocalStudentstm_Click(object sender, EventArgs e)
    {
        bindingSource1.Filter = "BirthPlace not like '" + "Mahabad" + "'";
        toolStripStatusLabel1.Text = "Non Local Student";
    }

    private void ExcellentStudentstm_Click(object sender, EventArgs e)
    {
        bindingSource1.Filter = "Average>17";
```

```csharp
        toolStripStatusLabel1.Text = "Excellent Student";
    }

    private void AcceptStudentInUpperstm_Click(object sender, EventArgs e)
    {
        bindingSource1.Filter = "AccptInUppr like '" + "Yes" + "'";
        toolStripStatusLabel1.Text = "Accept in Upper";
    }

    private void toolStripButton1_Click(object sender, EventArgs e)
    {
        this.Validate();
        this.bindingSource1.EndEdit();

this.english_Student_TableTableAdapter1.Update(english_project_Student_informati
onDataSet1.English_Student_Table);
    }

    private void toolStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
    {

    }

    private void dataGridView1_CellContentClick_1(object sender,
DataGridViewCellEventArgs e)
    {

    }


    }
}
```