Helwan University & UNINETTUNO

College of Engineering and Technology

Communication & Information

ATTENDANCE SYSTEM USING RFID TECHNOLOGY

Presented By:

Mohamed Hussein Ayesh

Mohamed Ahmed El-Amir

Ahmed Mohamed Hamed

Tarek Mohamed Naguib

Ibraheem Mahmoud Ibraheem

Supervised By:

Dr.Samir Gaber

DECLARATION

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Bachelor of Science in *(Communication Engineering)* is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

Registration No.: _____

Date: Day, xx Month Year.

“Feeling gratitude and not expressing it is like wrapping a present and not giving it”. William Arthur Ward. So we dedicate this book to our parents. Without their help we won’t be standing where we stand today. Also we dedicate it to those who contribute in forming our brains for five years.

ACKNOWLEDGMENT

This project would not have been possible without the support of many people. The authors would like to express their deepest gratitude to their supervisor, Prof. Dr. Samir Gaber who was abundantly helpful and offered invaluable assistance, support and guidance.

The authors would also like to thank to the Faculty for providing a proper learning methods & techniques that helped us in our entire project.

The authors wish to express their love and gratitude to their families; for their support & encouraging to finally getting that degree.

Special thanks also to our graduate friends, especially group members; Mohamed El-Prince for sharing their invaluable assistance.

ABSTRACT

This project has been made to replace the old system of taking the attendance in some papers with a new system that uses a more modern technology that will provide assistance to the employees or doctors to take attendance easily with no problems.

In this project we will use RFID technology and integrate it with A GUI and Database system that will contain all the required information & tools for taking the attendance and save it.

TABLE OF CONTENTS

2011

Chapter one

INTRODUCTION

Helwan University
International Telematic University UNINETTUNO

Chapter one

1 INTRODUCTION

1.1 Overview

Attendance management system is software designed for taking attendance for both students and employees in college or for business means. It is a desktop application and will be used for a specific college.

-Having a computerized attendance system has proven to improve the efficiency of any business and in this case it is for college. It will eliminate the need for a lot of paper work and will help in tracking both employee's and student's absence dates.

The attendance system consists of two parts:

1**) Attendance system for students**

The lecturer takes the attendance of students in the class using the RFID reader integrated with software program designed for this purpose & then it will be saved in the database of the system.

2) **Attendance system for employees**

The employee in charge will monitor the process of each employee registering in the system to avoid someone register a fellow employee and he/she is absent. Also this will use RFID reader integrated with software designed for this purpose.

In attendance management system we will have either an admin who will have the right to create or modify database of the system or an ordinary user, which can be doctor or student or employee.

1.1.1 Why we need this new system?

In present working system it has many disadvantages:

1- Time consuming and less efficient.
2- Requires the lecturer to deal with attendance computation manually which is very irritating.
3- Increase the chance of having errors because attendance will be dealt manually.
3- Losing the paper of attendance could cause many problems.
4- Not user friendly

In the new attendance system we will fix the previous errors as:

1- The system will be user friendly as we will provide the user with a GUI that will be easy to learn and use.
2- You will have a backup for your papers, as you will be able to have it softcopy and hardcopy.
3- The software will do all the calculations not the lecturer for example.
4- Saves time and more efficient.
5- No errors when it comes to taking attendance.

1.2 RFID Technology

-RFID is considered as one of many Automatic Identification technologies such as bar code, voice recognition, touch memory, smart cards.

-RFID technology gather data about some object without touching or seeing data carrier (Microchip connected to an antenna forming what is called a Tag) as it depends on electromagnetic waves. The chip then transmits information to a reader within a given range and eventually passes the information to the computer attached to it.

-RFID has a wonderful feature of providing unique identification for objects.

1.2.1The Elements of an RFID System

RFID systems have two basic elements which are:

1) RFID Reader
2) RFID Tags

1.2.1.1 RFID Reader

RFID Reader is a device that is used to interrogate an RFID tag. The reader sends radio waves to the tag and then the tag sense and detects it and sends back its response containing all the information about the tag.
The RFID Reader can read the tag depending on its antenna and the tag's antenna also on how the tag will be placed on the object being detected.

-RFID readers are usually on, continually transmitting radio energy and waits for any tags to respond to it.

-RFID readers come in many sizes.

1.2.1.2 RFID Tag

Also called the transponder. It is the most important link any RFID system. It consists of an integrated circuit and an antenna combined together to form a tag. The tag contains a radio receiver & modulator for sending its response back to the reader. RFID tags detect radio energy emitted by the reader.

- While holding the RFID label into some light, the one can see the components of integrated circuit in it.

- Tags have three types:

1) Passive tags: operates when there is an incoming RF signal, it is smaller, cheaper and have longer lifetime than active tags.
2) Active tags: it has a battery
3) Semi-passive tags: it has a battery but it uses reader's signal to operate

- before buying RFID label, we should determine what tag to be used depending on the application it will be used into as tags could be read only or read write or write once and read many because this could cause a lot of problems.

- RFID tags can communicate with any reader

1.2.2 Radio Frequency Interference

Radio Frequency (RF) interference can be caused by many sources such as:

.Foil and metals will cause problems to RFID as for example metals reflect radio waves.
.Placing RF equipments close to each other will cause interference.
.Being exposed to water and liquids

-This interference can affect RFID performance by prevent reading/writing to the tags for example

1.2.3 Advantages of RFID over Other Technologies

RFID technology has some features that distinguish it from other technologies present now days. For example bar code is used to detect the presence of an item by having a direct line of sight with it, so the item has to face the barcode in the right direction with nothing blocking the beam between them, also for credit cards as the magnetic strips on it must be lined up correctly with the card reader, so the card must be inserted in a particular way.

-But RFID can solve these problems as it has the following advantages:

.A scan does not require line of sight.
.Multiple items can be scanned at the same time.
.RFID tags range in size which allow RFID technology to be used in a lot of places.
.Tags can have read/write memory which barcodes doesn't have.
.There is types of tags that can be written and rewritten many times.
.RFID tags have longer range than barcodes for examples

1.2.4 Things to be considered when implementing RFID

Determine the business need: this is very important as the one must now what he actually needs from the system before he starts doing anything.

Evaluate potential changes: as there are some systems like barcode that could be better than RFID and cheaper.

Have a clear and upfront design: as it is important to see the structure of your application when implementing RFID and putting an upfront design that will give a good idea of how the application will work.

Start small: as the one must start with a mini application before going big and that's will help in reducing any wasting time or money as it is better to fall small than to fall big.

Operate new systems with old existing systems: the main idea of RFID is to improve some system, so it is said that it is better to operate the new system in the presence of the old system and that will help in seeing if the new application operates better than the old one or vice versa, also it will be good to use the old system to get a new ideas for the new system and avoid the limitations of the old system.

Be flexible: it is clever that you be flexible to implement any changes during developing your new system because sometimes you are exposed to ideas that are better than your old ones or a development in technology happened during designing the system.

1.2.5 LIMITATIONS OF RFID TECHNOLOGY

1- The tags could be damaged

2- Putting the tag or the reader in the wrong places or wrong temperatures could harm the system.

3- The ability of readers to read several tags at once. This could cause problems to some systems.

4- In some applications it is necessary that no unauthorized people should read data written in tags

1.2.6APPLICATIONS

1) Identify location of objects that also could be missing. An example for that is tags that are put on animals to identify their locations to ensure that no animal gone missing.
2) Data transfer from or to the RFID tag.

Chapter Two

Database Building

2011

Helwan University

International Telematic University UNINETTUNO

Chapter two

2 DATA BASE BUILDING

.Any attendance system needs a place where its data will be saved or edited. This place is called Database of the system. We need to first build this database before going any further in the project as the database will define the shape of the system. It will be highly important when integrating it with the GUI
We will use SQL Server 2008 (Figure2-1) as the software where we will design our database

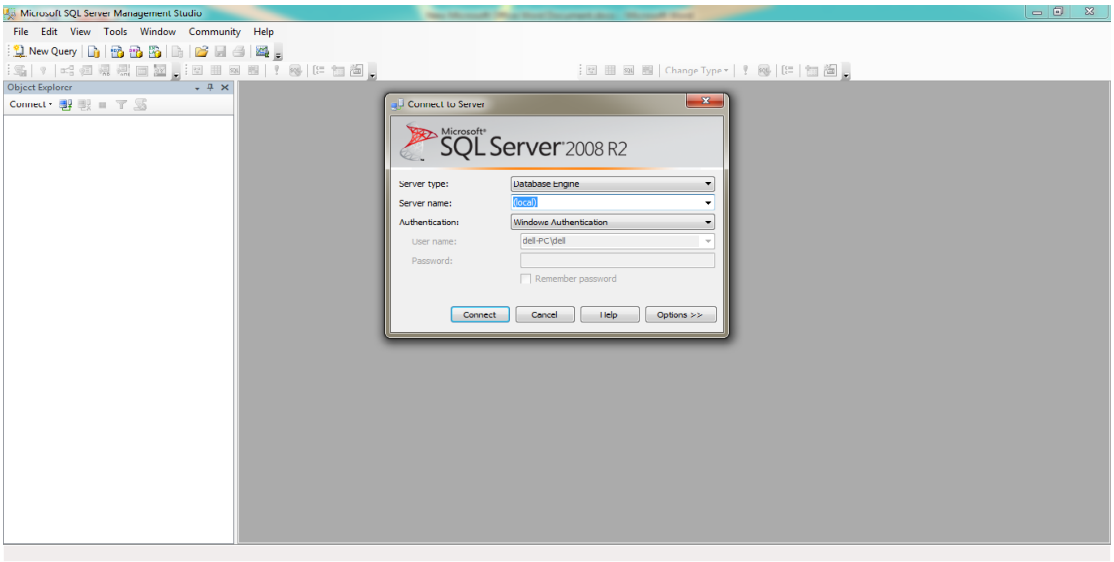


Figure 2-1 Start window of SQL Server 2008

In Figure 2-1 we will first define our server type that we are going to use whether it will be Database engine or analysis services or Reporting services or SQL Server compact or integration services. In our case we will use Database engine.

- Secondly depending on our installation procedure we can choose the authentication to be windows authentication or SQL Server authentication. As we are working on our own lab tops it will be ok to use windows authentication but when this software will be applied in the college so it will be necessary to have SQL Server authentication to keep our work on database secure & not to be modified by anyone who has no right to access the database as the PC could be accessed by anyone in the college.

- Thirdly we will press connect to complete our connection to the database.

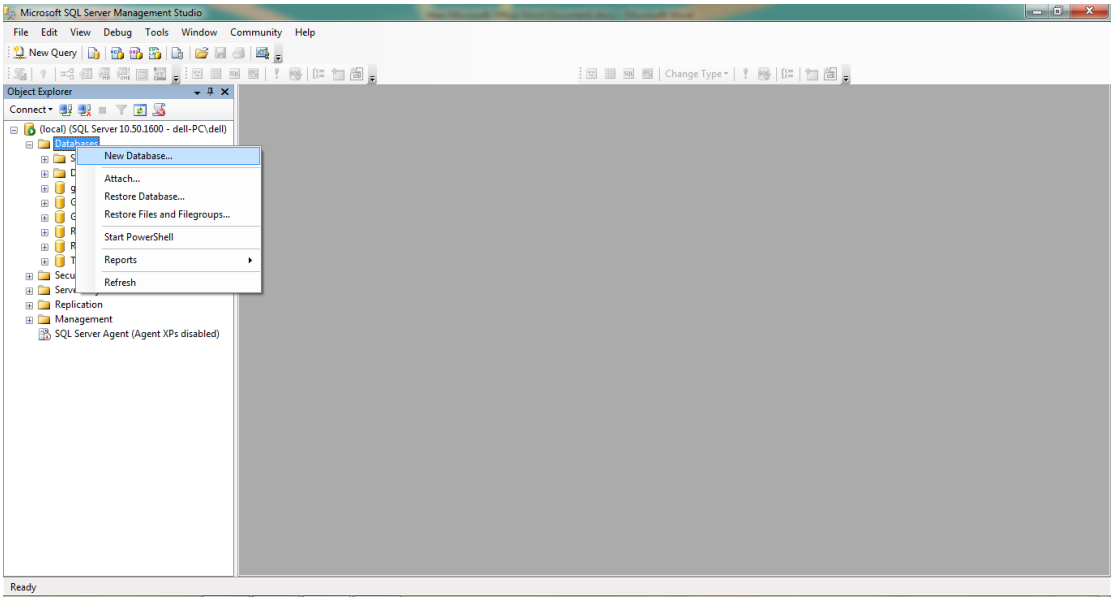


Figure 2-2 start a new database

In Figure 2-2 after we connect to the server we will click on database and start a new database that will create a file for this database.

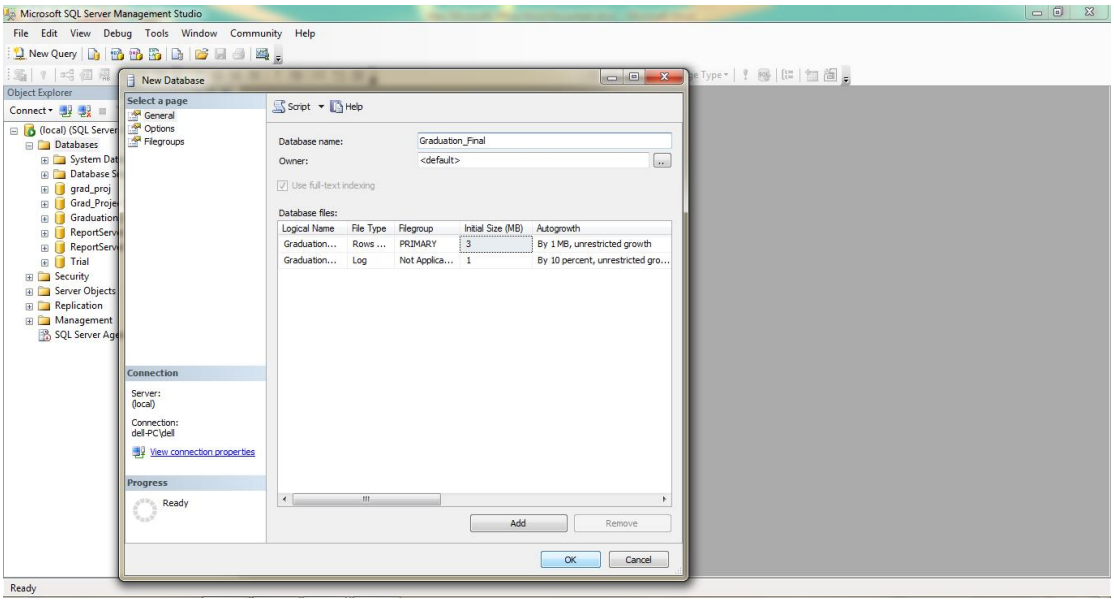


Figure 2-3 naming database

In Figure 2-3 we will set our database name and the path of the file where it will be saved in also we can initialize the size of the file but we will not do anything as the size will be dynamically increase or decrease as we write or delete in our database.

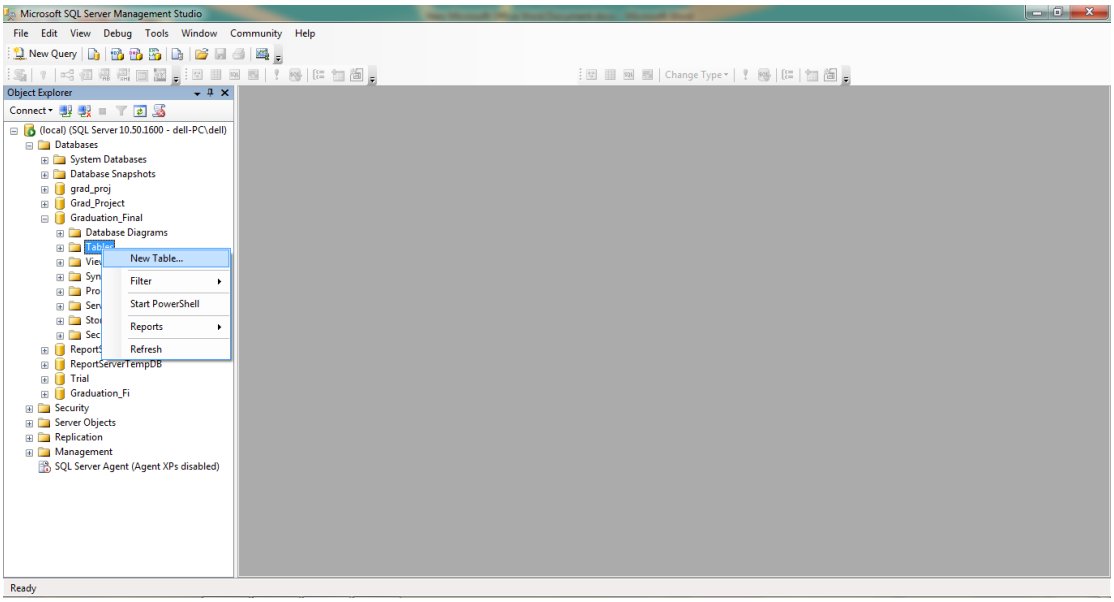


Figure 2-4 beginning with tables

In figure 2-4 we will create our database using tables as by using tables it is faster and easier
So under the name of our database which is Graduation_Final we select a new table to start typing and building our database system.

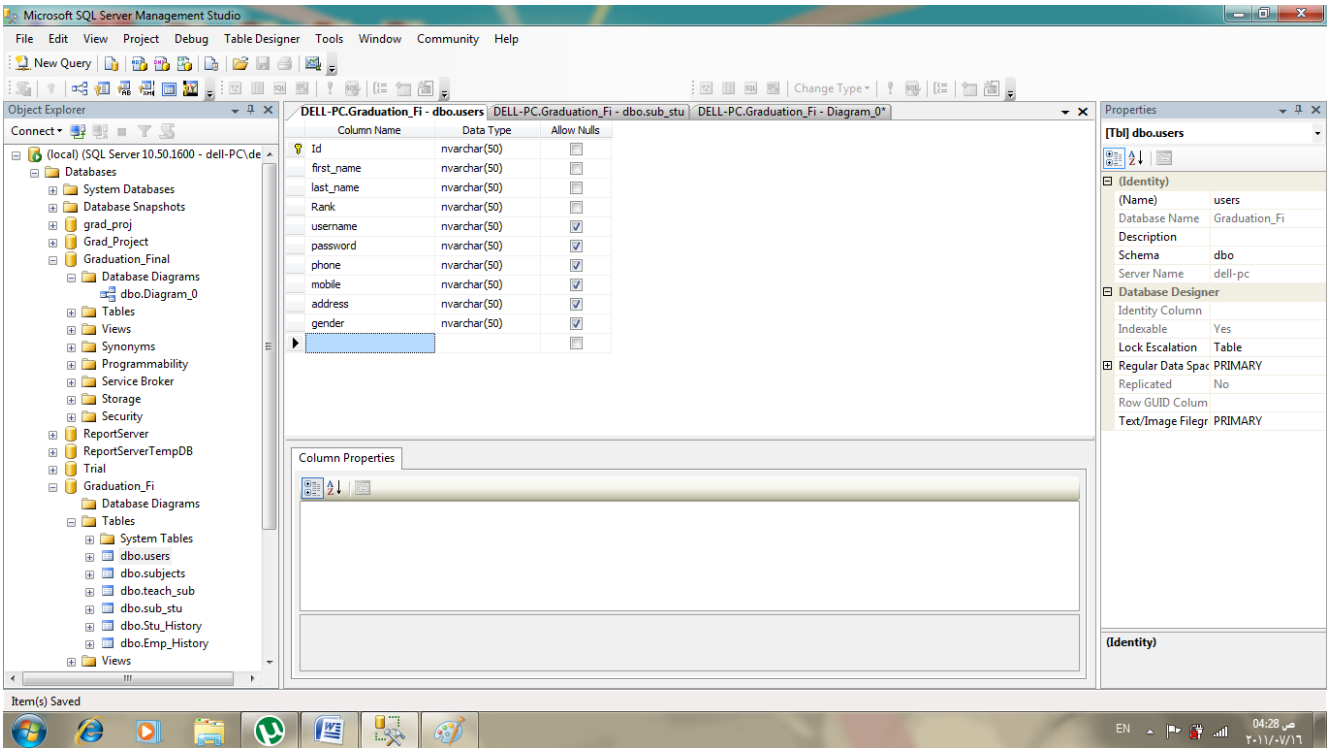


Figure 2-5

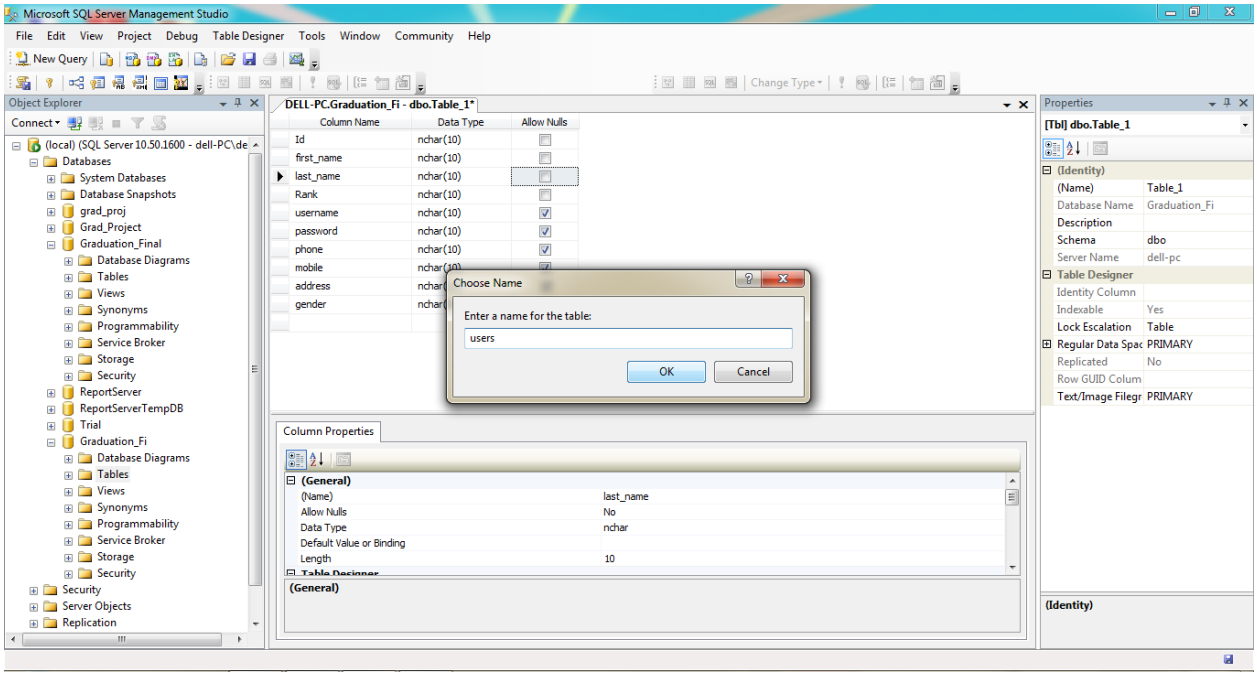


Figure 2-6 building tables (users table)

The users table will determine all users in the database whether they are students or administrators or doctors etc…. & depending on the rank the user will have access to some information that another user will not have or the otherwise.

There is also a username and a password that will be assigned only to administrators or doctors with administrative work.

-In figure 2-5 we have three columns: the first is to decide the column name, the second is to define data type that will be written like integer or char or nvarchar and so on, the third determines whether you will allow nulls in this column or not (allow nulls means that when filling database with data you can leave some columns null without making any bad effects on the system also allow nulls columns is important to help the user not to forget about the necessary information that must be written in the database).

- We then start to type the required data in the table and we will name it as users

- We set the primary key at id column.

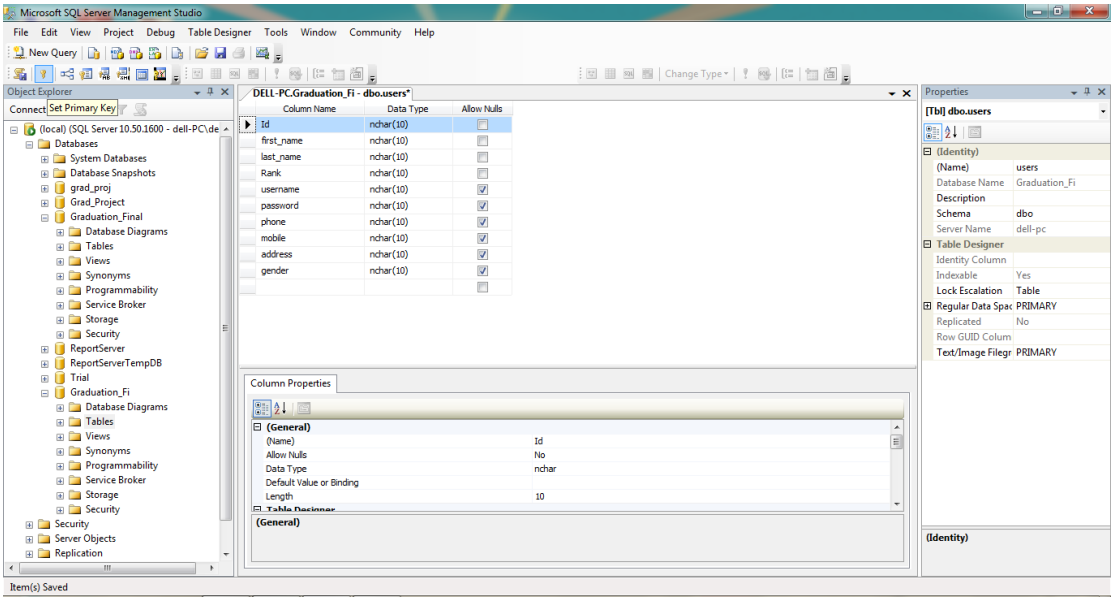


Figure 2-7 final outcome of users table

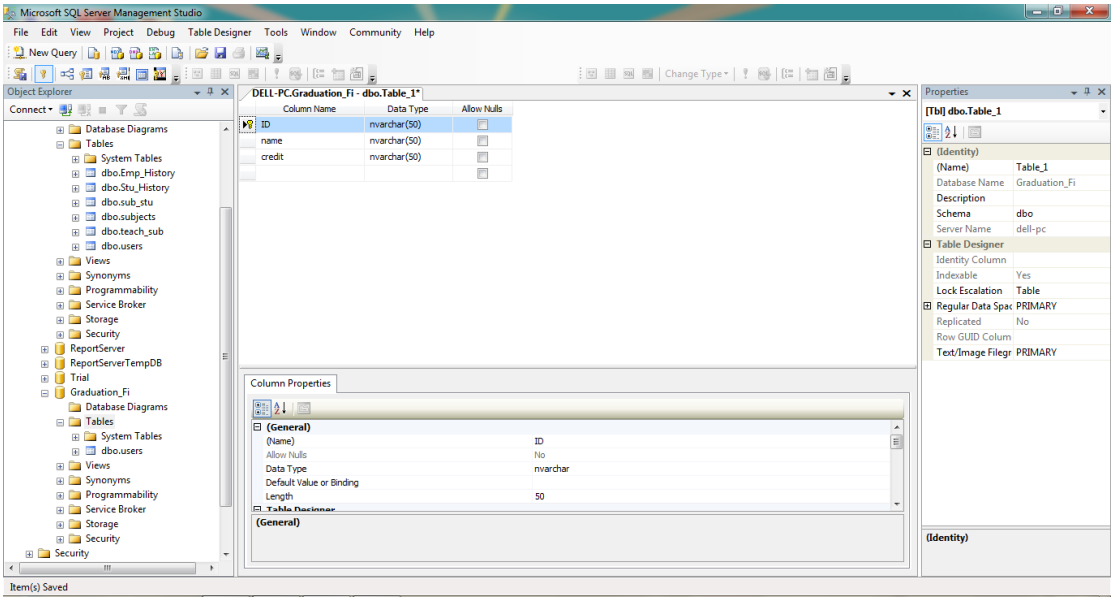


Figure 2-8 Subjects table

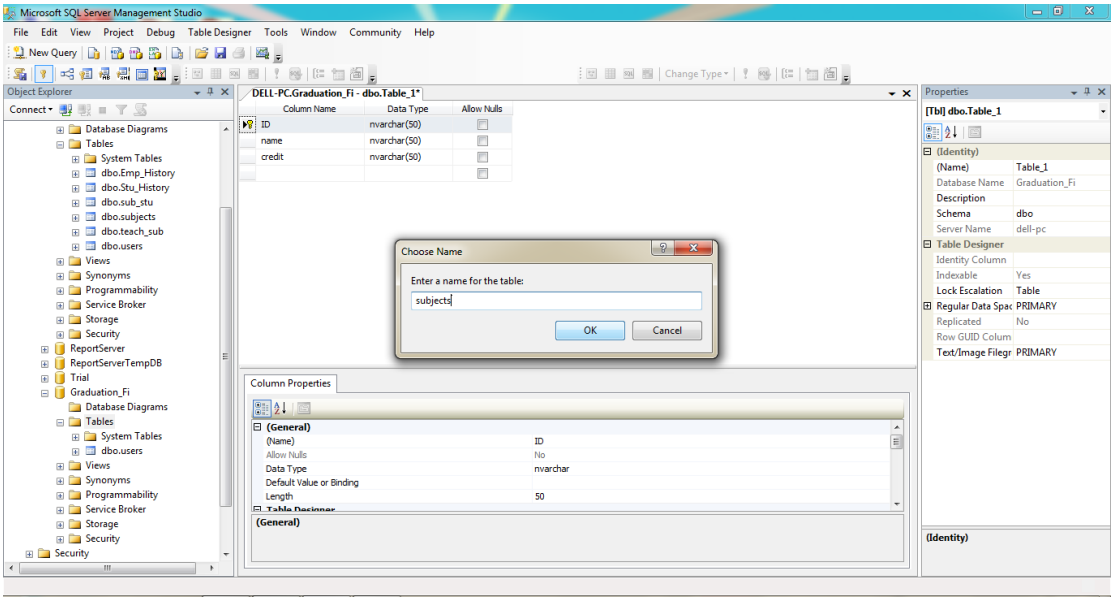


Figure 2-9 Subjects table

. This table will tell us the every subject name and id we have in our database and its credit hours.

As we did in the users table we define our columns here for Subjects and we define its id and name and credit hours of it.

Note: we use nvarchar (50) so that data written in column will adapt to the size of the input data written by user.

- We then start to type the required data in the table and we will name it as subjects.

- We set the primary key at id column.

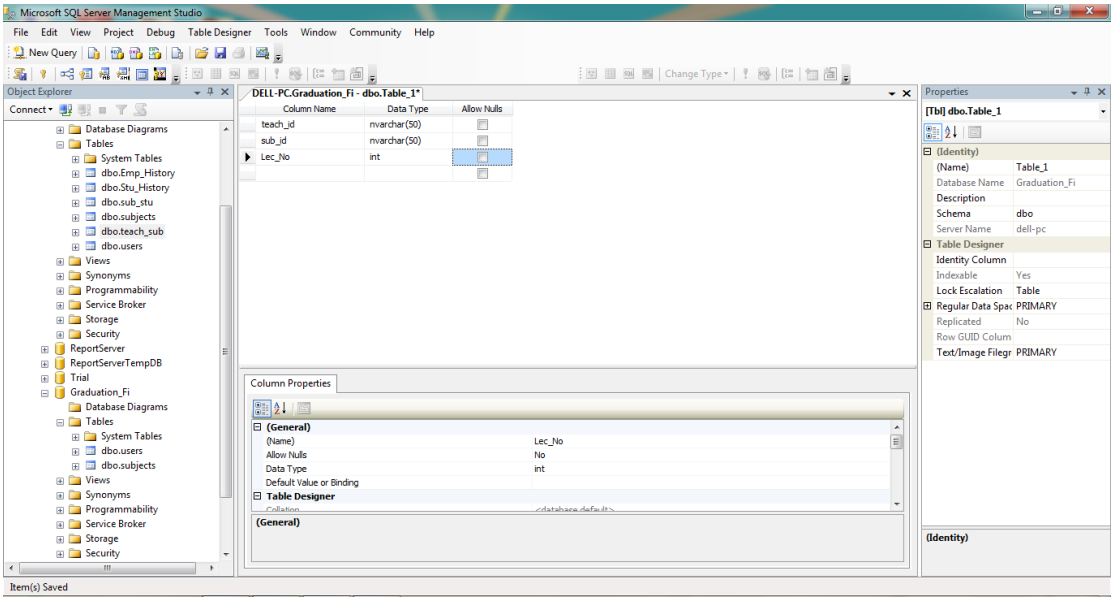


Figure 2-10 Teach_Sub table

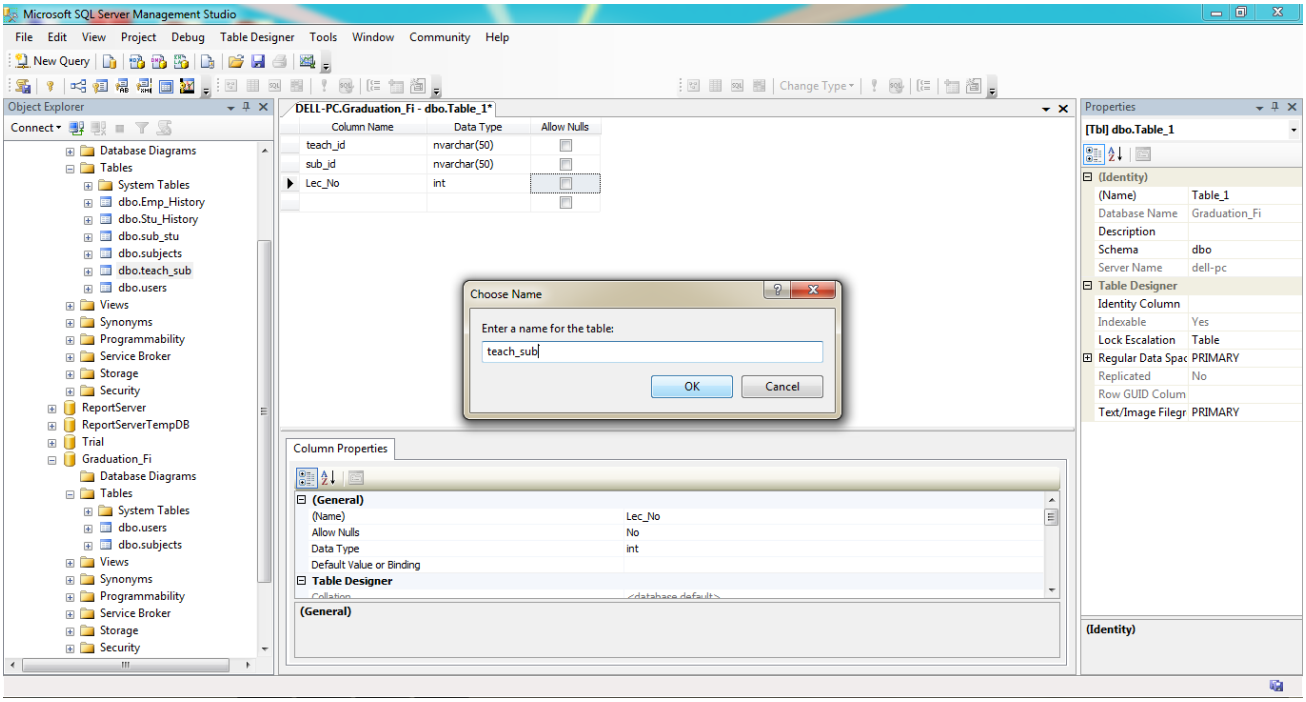


Figure 2-11 Teach_Sub Table

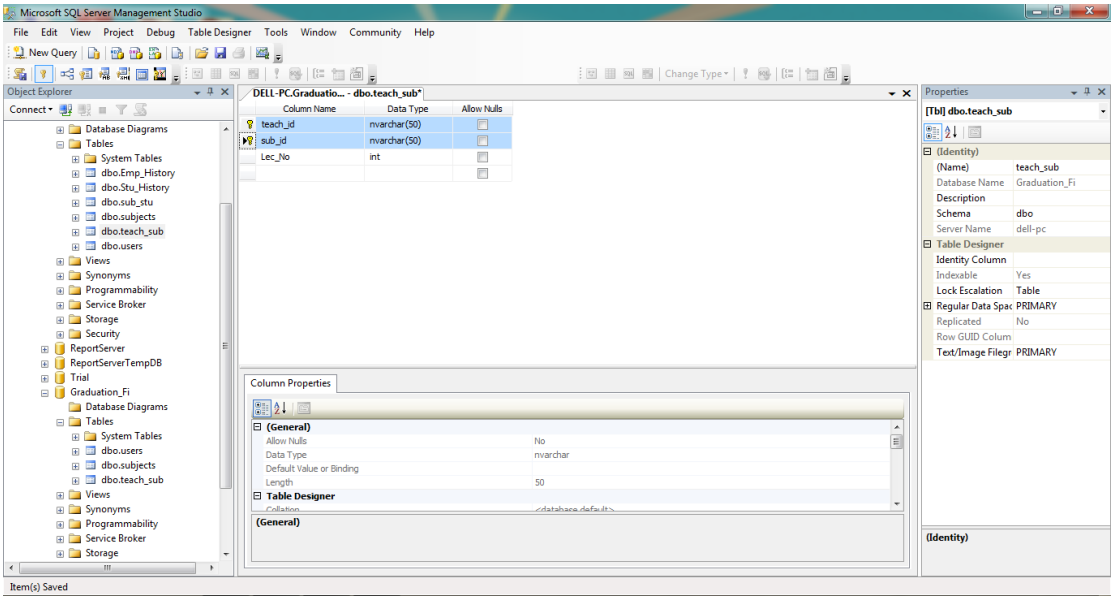


Figure 2-12a Teach_Sub table

. This table will tell us the whole subjects with doctors that will teach them and it also provides the number of lectures that the doctor will give for each subject.

As we did in the users table & subjects table we define our columns here for Teach_Sub and we define teacher_id, subject_id & Lecture number

- We then start to type the required data in the table and we will name it as teach_sub.

- We set the primary key at teach_id column and sub_id column.

Note: here we have two primary keys because we need to uniquely define subject and teacher ids.

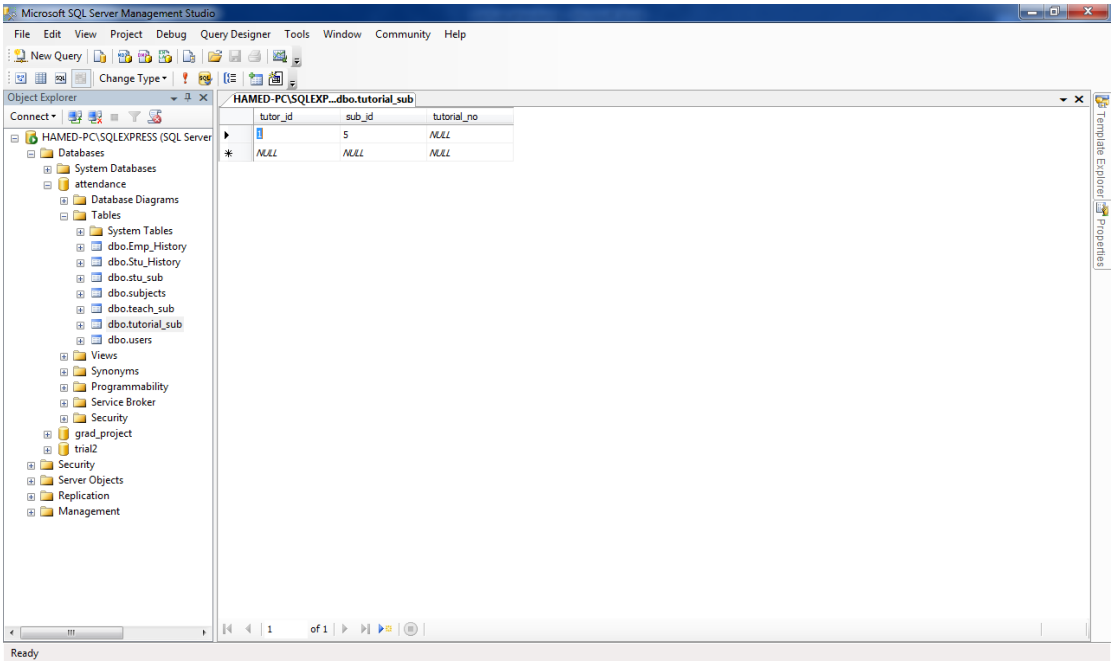


Figure 2-12b Tutor_sub

This table has the same structure as teach_sub table but there is only one difference as instead of the doctor, the table will talk about tutor.

- This table will tell us the whole subjects with tutors that will teach them and it also provides the number of sections that the tutor will give for each subject.

As we did in the users table & subjects table we define our columns here for Tutor_Sub and we define tutor_id, subject_id & section number.

- We then start to type the required data in the table and we will name it as tutor_sub.

- We set the primary key at tutor_id column and sub_id column.

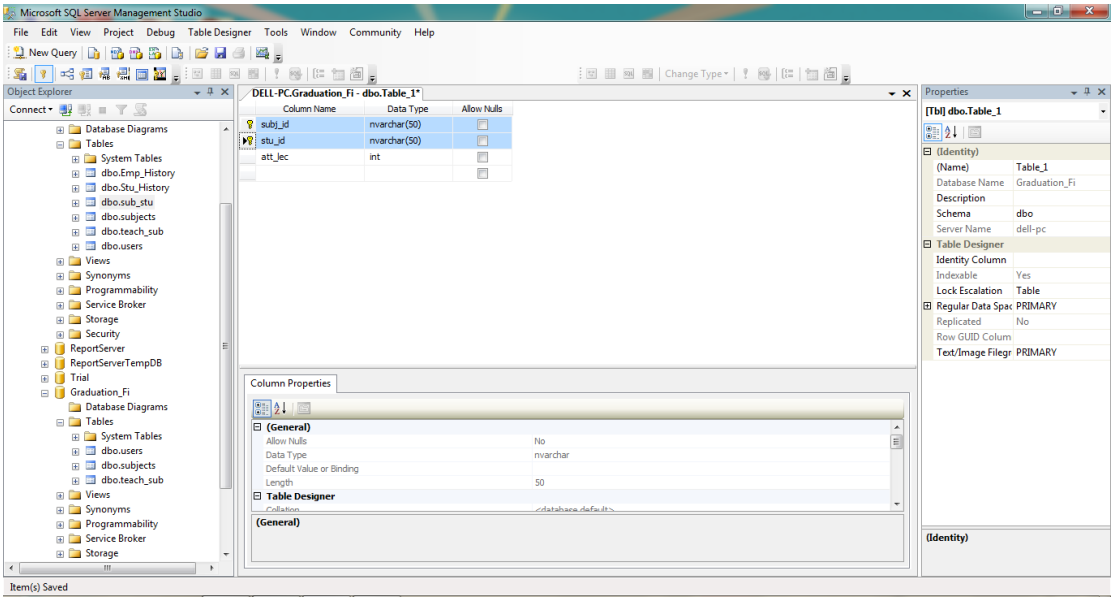


Figure 2-13 sub_stu Table

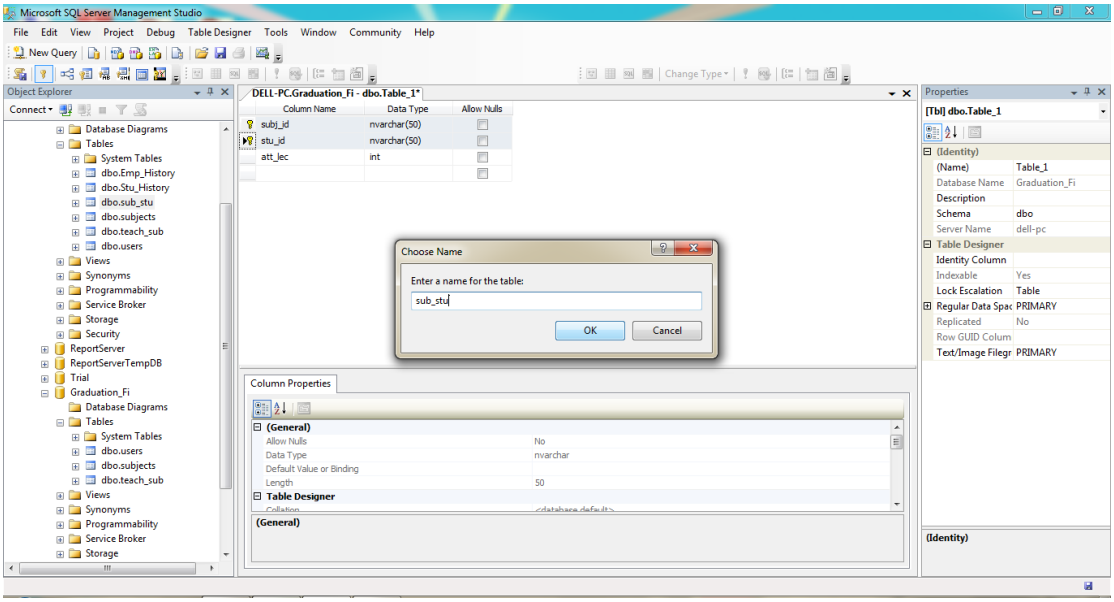


Figure 2-14a Sub_Stu Table

This table tells us all the students and the subjects they have taken & number of lectures in each subject that they have taken.

- We then start to type the required data in the table and we will name it as sub_stu.

- We set the primary key at subj_id column and stu_id column.

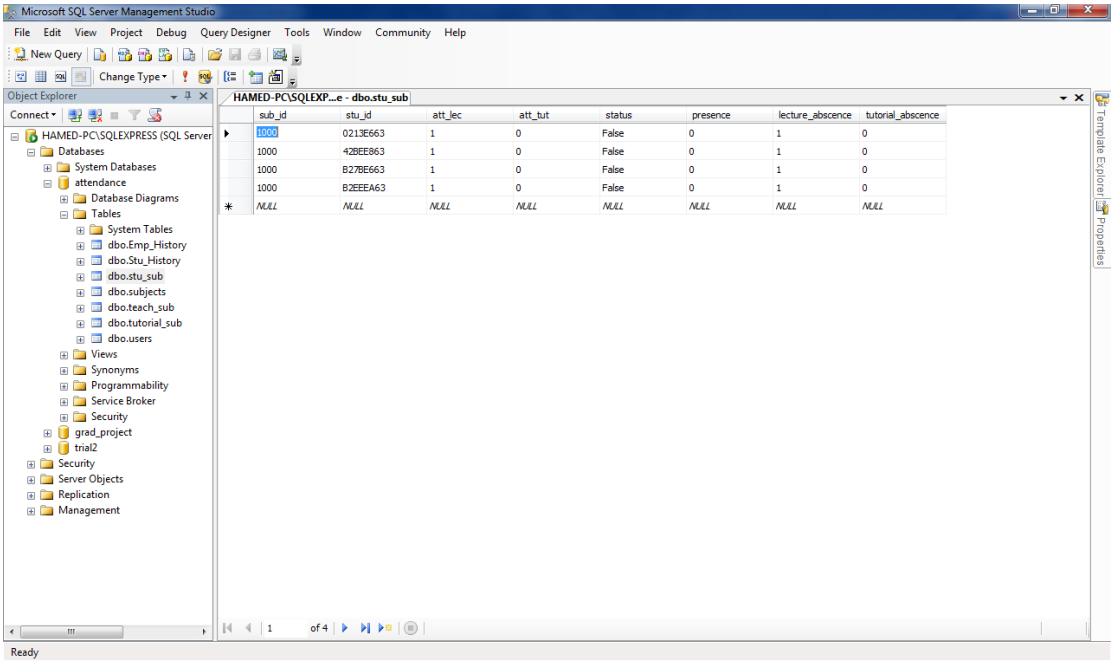


Figure 2-14b Sub_Stu table

- Note that we added other columns to the table and that to adjust table to our new needs so the last figure is the one that we begin with and this figure is the figure that we ended with. The added columns are (att_tut, status, presence, lecture attendance, tutorial attendance).

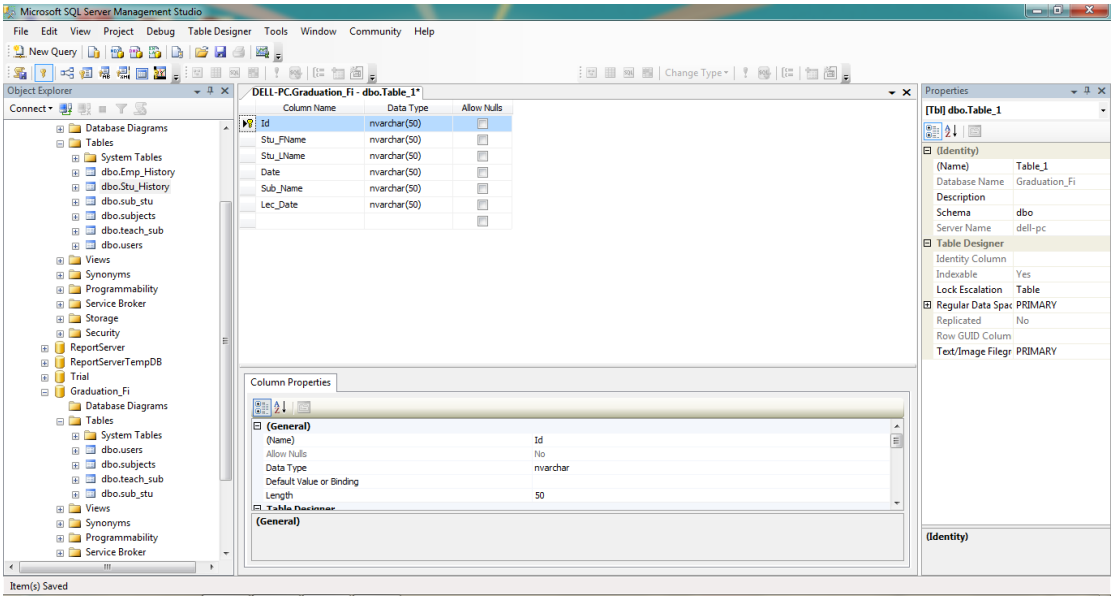


Figure 2-15 Stu_History Table

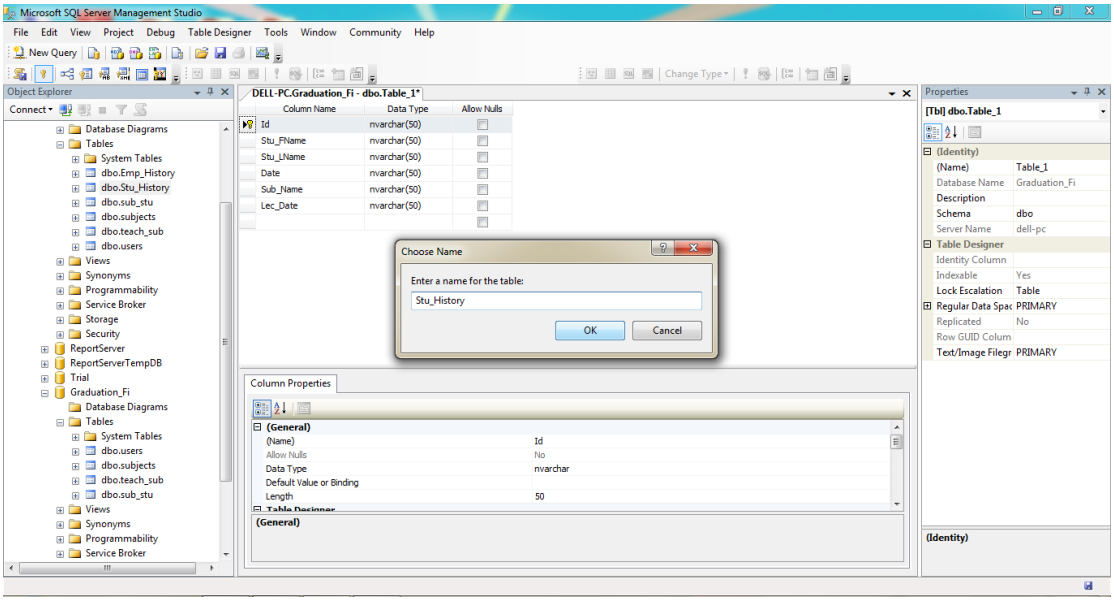


Figure 2-16 Stu_History Table

. This table will provide us with the history of the attendance dates of any student in the database.

-Here we define our columns for Student_History and we define Student_id, Student first name, last name, subject name &lecture date

- We then start to type the required data in the table and we will name it as Stu_History.

- We set the primary key at id column.

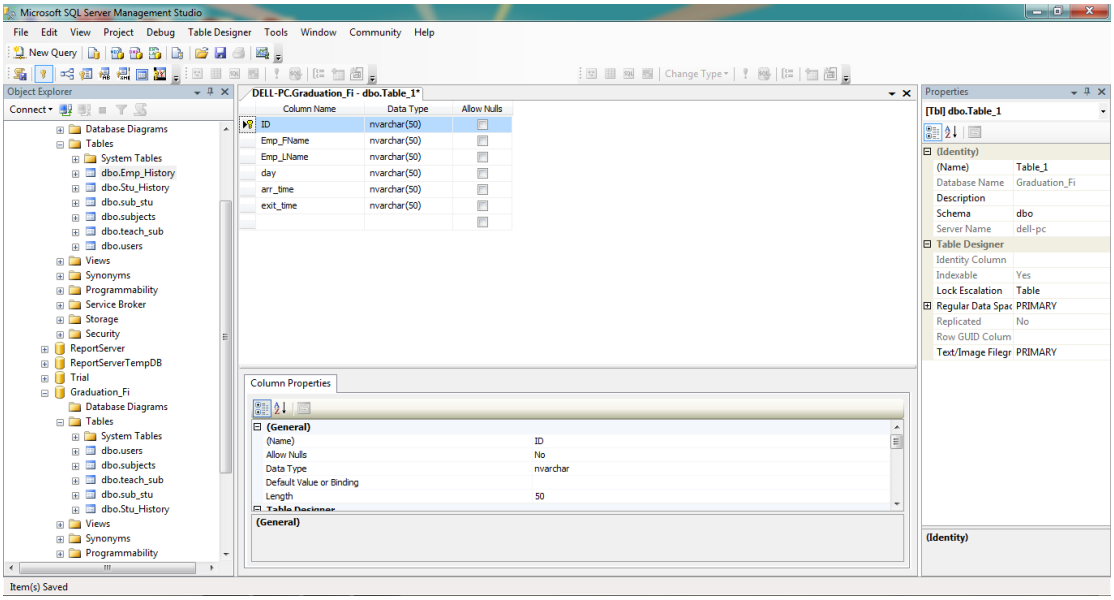


Figure 2-17 Emp_History Table

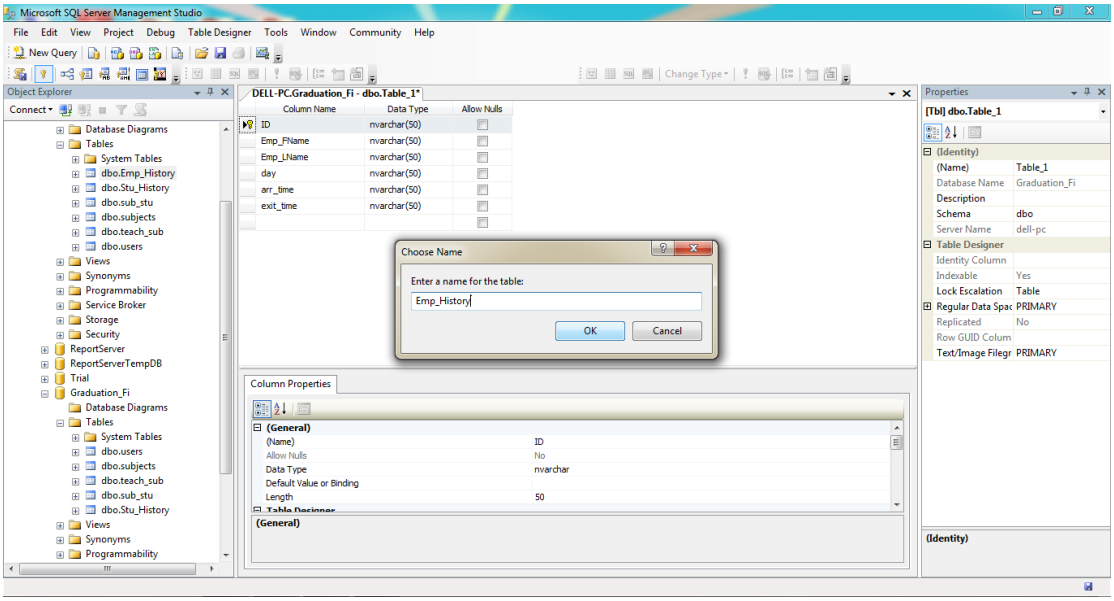


Figure 2-18 Emp_History Table

. This table will provide us with the history of the attendance dates of any employee in the database.

-Here we define our columns for Employee_History and we define Employee_id, Employee first name, last name, arrival time & exit time.

- We then start to type the required data in the table and we will name it as users

- We set the primary key at id column.

Building a database diagram is very essential to define relations between tables and connecting them together in a very easy way yet complex.

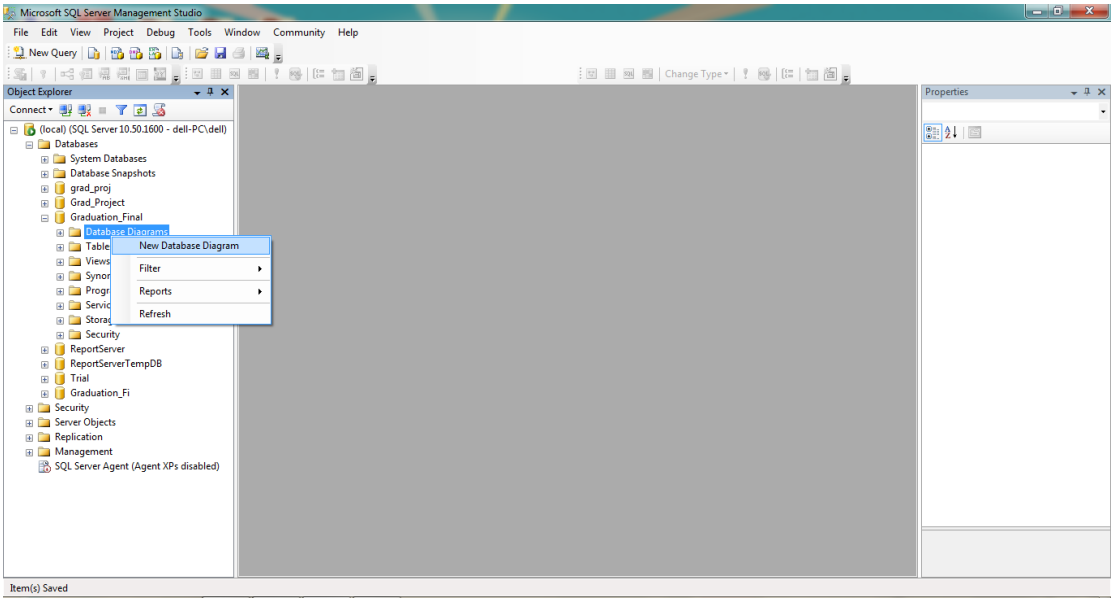


Figure 2-19a Database Diagram

To start a database diagram we click on the name of our database then we look for database diagram then right click on it and choose new database diagram.

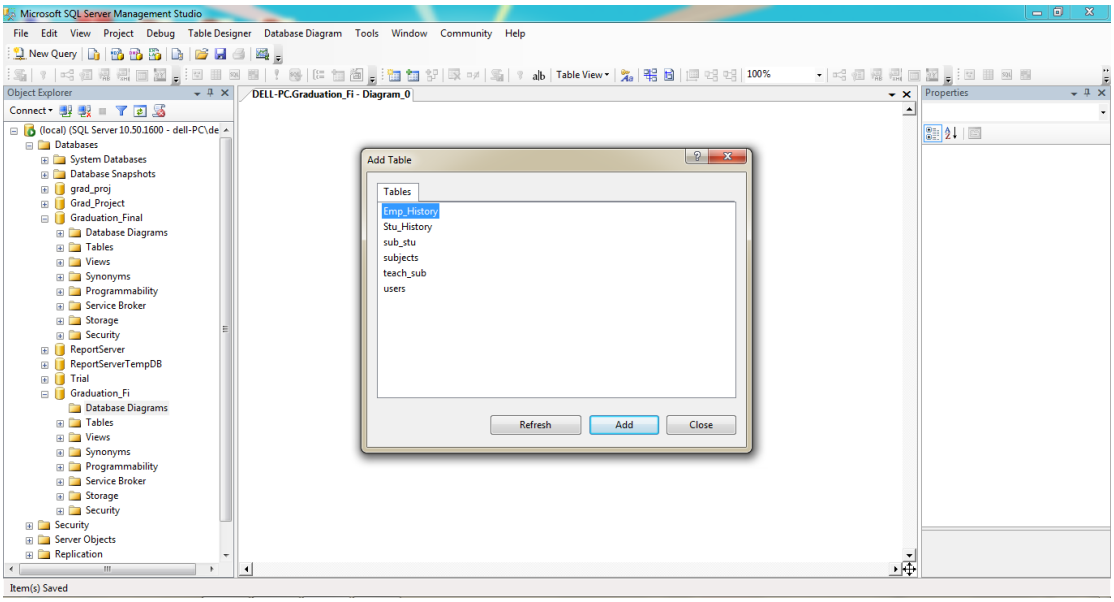


Figure 2-19b Database Diagram

After that there will appear a window that contains all the tables in the database. Then we choose the tables that we want to relate & in our case we will use all of them so we click on each table and press add until the window is empty

Note: we have refresh button that will be used if you have made a new table and it doesn't appear in the window.

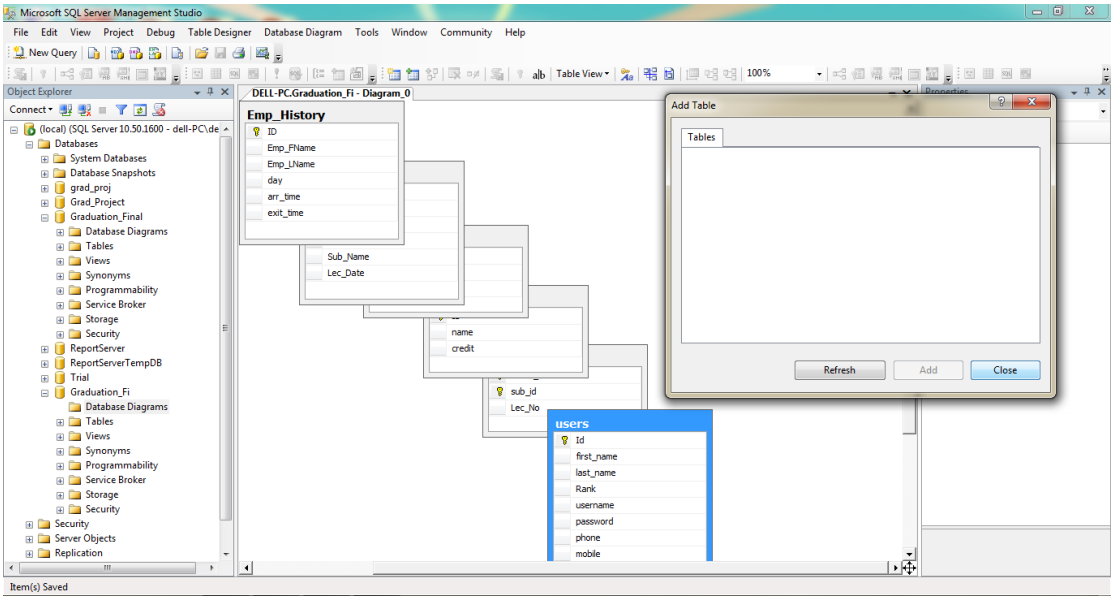


Figure 2-19c Database Diagram

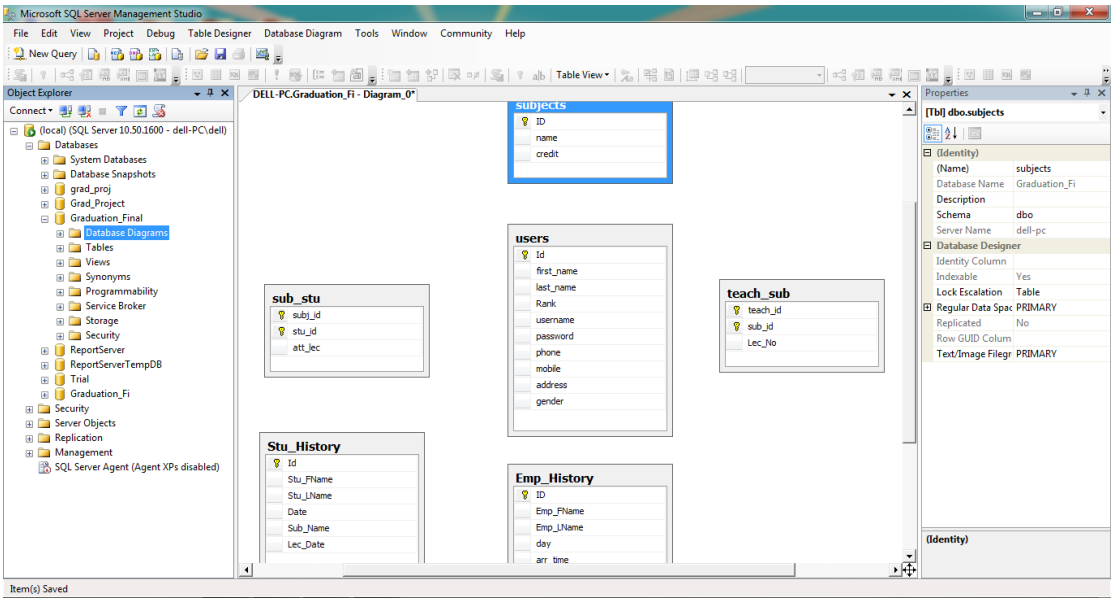


Figure 2-19d Database Diagram

In figure 2-19d this will be the whole tables in the database we try to put them near each other so it will be easy to edit the relations quickly and easily.

Important Note: now we didn't explain why in each table we put a primary key so before going to the next figure we will discuss this.

In this system we will need all the tables to depend on the users table & subjects table which contain all the ids and names of users and all subjects, so for example the teach_sub table it contains teacher id and subject id as primary keys, those two primary keys will have a relation with users and subjects tables.

The teacher id will be taken from users table and subject id will be taken from subjects table in a way that every teacher could be teaching one or more subjects and many subjects could be taught by one teacher.

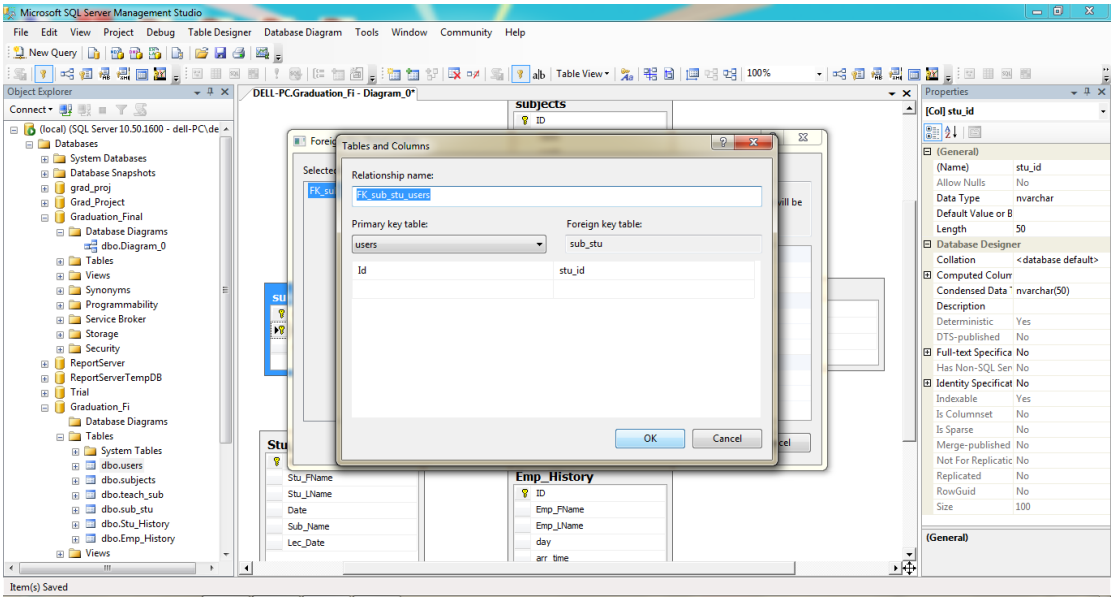


Figure 2-19e Database Diagram

In figure 2-19e we have this window as we were trying to connect and make a relation between sub_stu table and users table using the id of users and stu_id

This means that stu_id is pointing to id in users table and that will also mean that you are creating a foreign key between the primary key of id in users table and student_id that will be known as my foreign key which exists in sub_stu table.

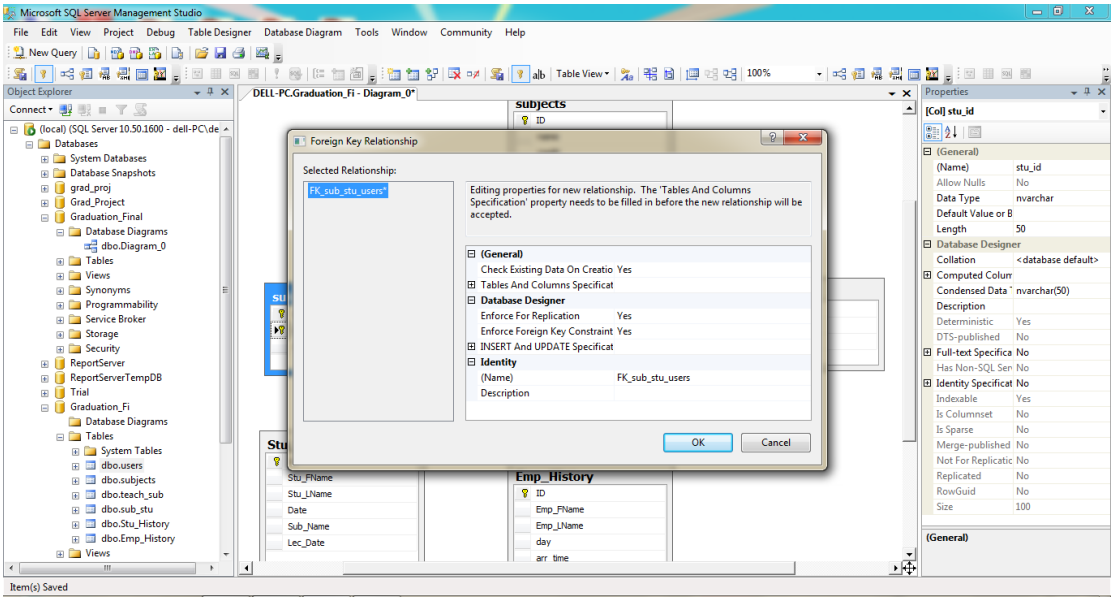


Figure 2-19f Database Diagram

Here we will not do anything but to press ok.

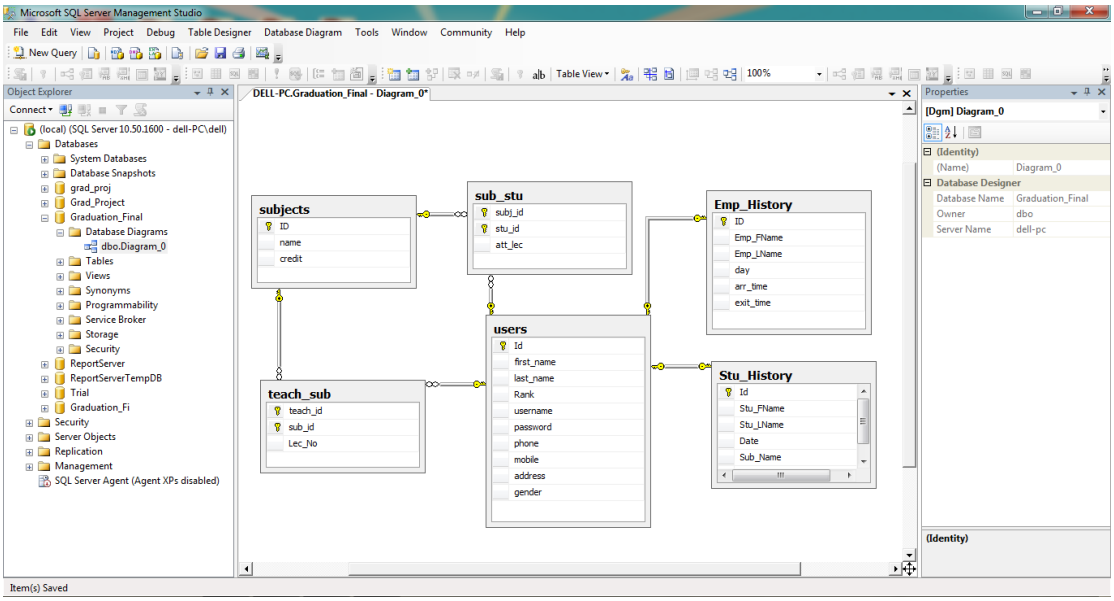


Figure 2-19g Database Diagram

This is the final outcome of diagram. Here we notice that all relationships have been made.

-Subjects table & sub_stu → the relation is between ID and subj_id.
It is one to many relationship.

-Subjects table & teach_sub → the relation is between ID and sub_id.
It is one to many relationship.

-users table & teach_sub → the relation is between id and teach_id.
It is one to many relationship.

-users table & sub_stu → the relation is between id and stu_id.
It is one to many relationship.

-users table & Emp_History → the relation is between id and ID.
It is one to one relationship.

-users table & Stu_History → the relation is between id and id.
It is one to one relationship.

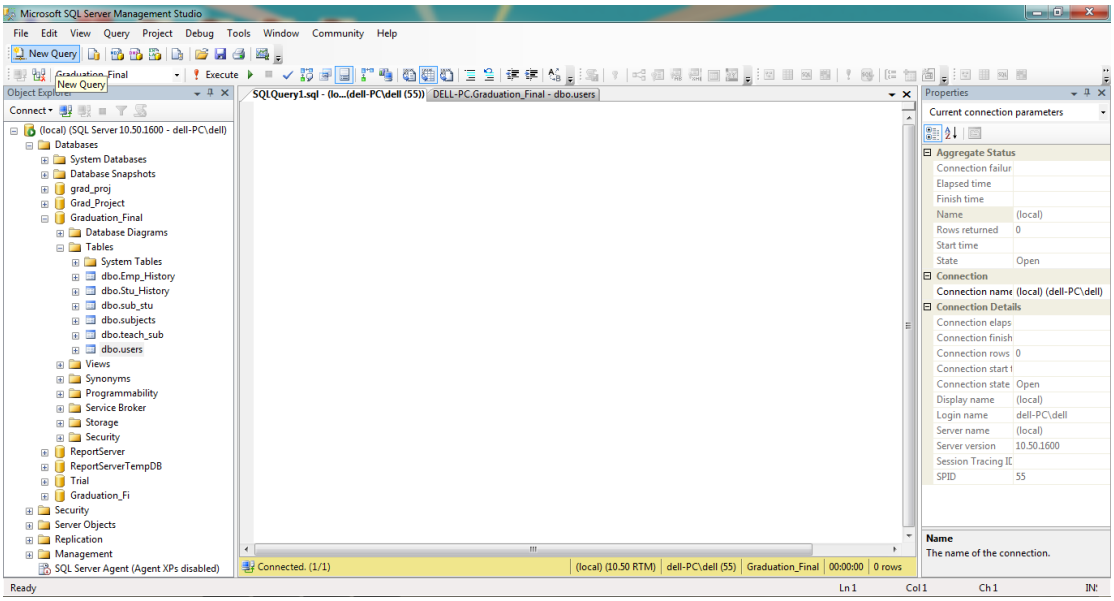


Figure 2-20 Queries

It is very important to use the queries to see if the connection between tables was done successfully and in the way we want.

The following figures are examples for using queries:

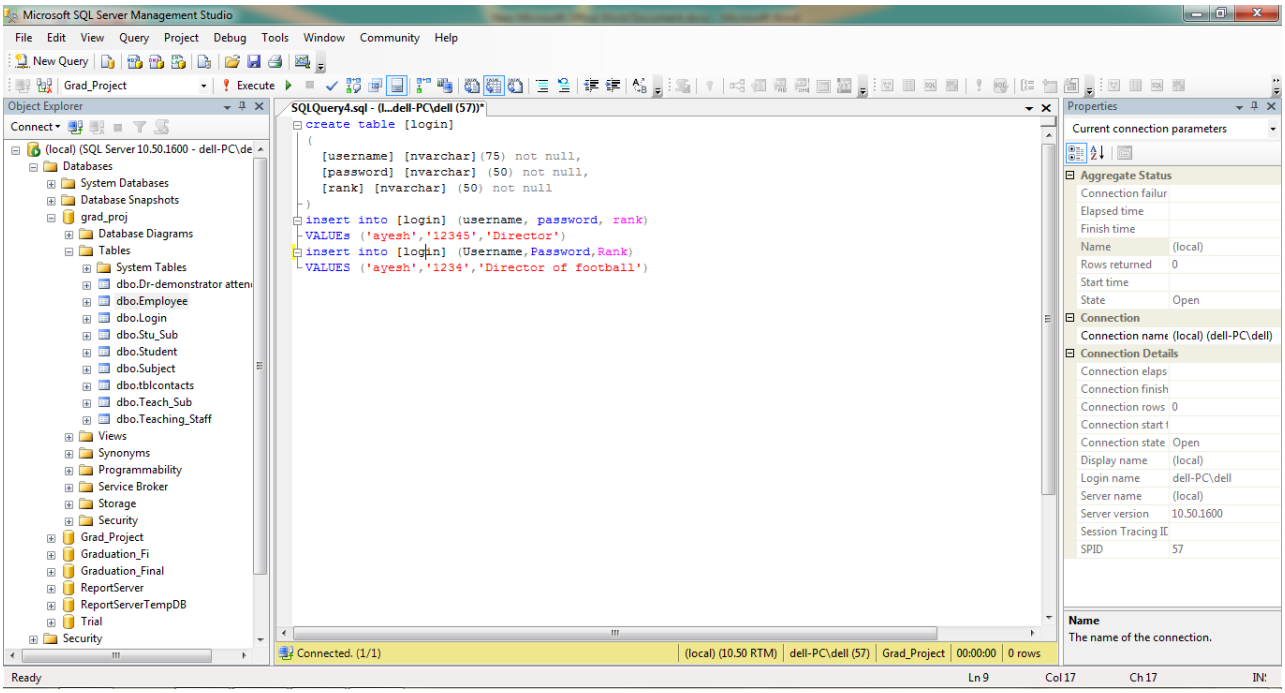


Figure 2-21 example of writing a query

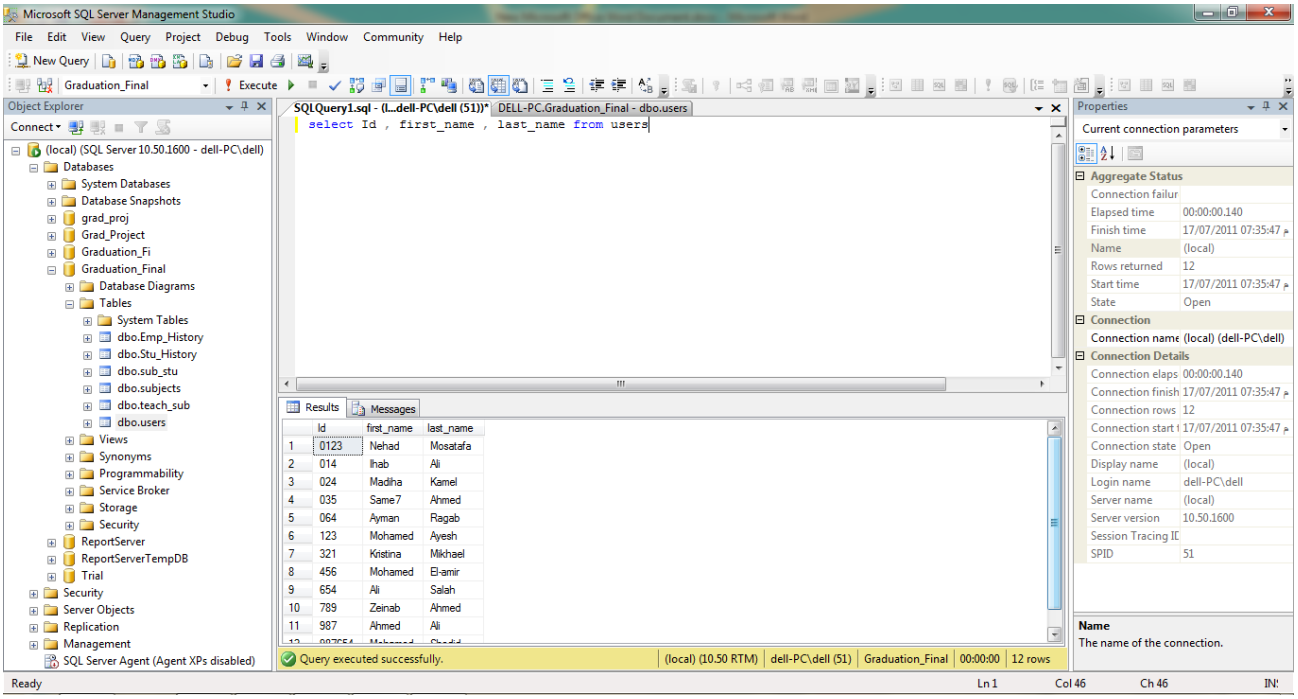


Figure 2-22 another example of writing a query

Chapter Three

GUI Graphical User Interface

2011

Helwan University

International Telematic University UNINETTUNO

Chapter three

3 GUI Graphical User Interface

3.1 Overview

It is the second phase of the system after building database. The GUI is basically used to provide an easy access to the database of the system. Due to its understandable structure any one can use the GUI from the ordinary user to the professional user.

- This phase is very important in attendance system as it will connect reader that will read ids to database of the system. This structure must be easy and understandable by any employee as the normal employee will not understand how to access database or the code of GUI.

- GUI will be built using C# programming language in Microsoft Visual Studio 2008.

3.2 Building GUI

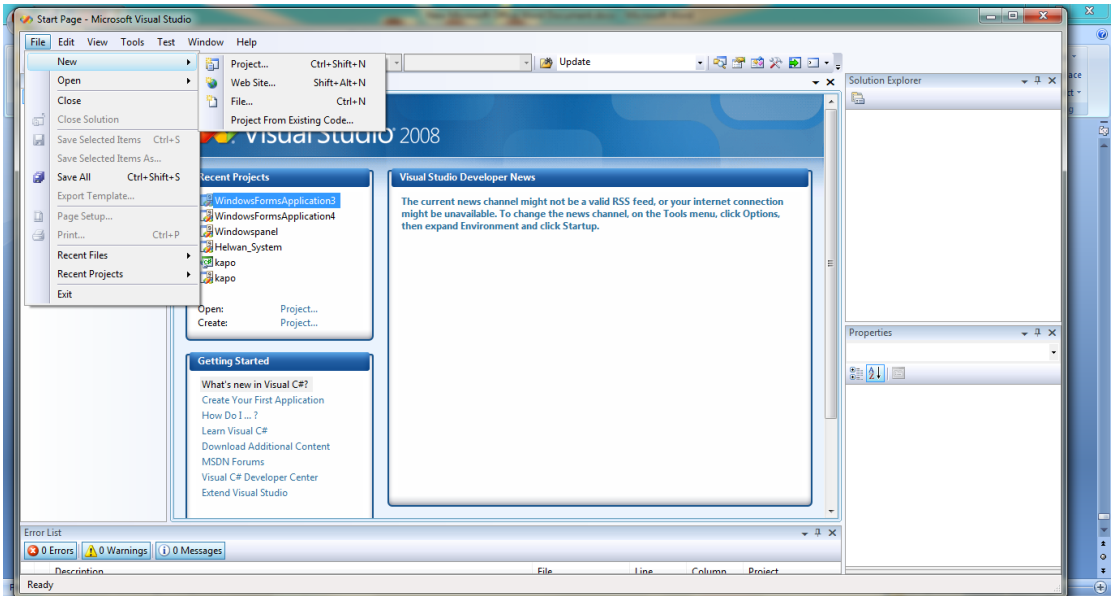


Figure 3-1a Starting page

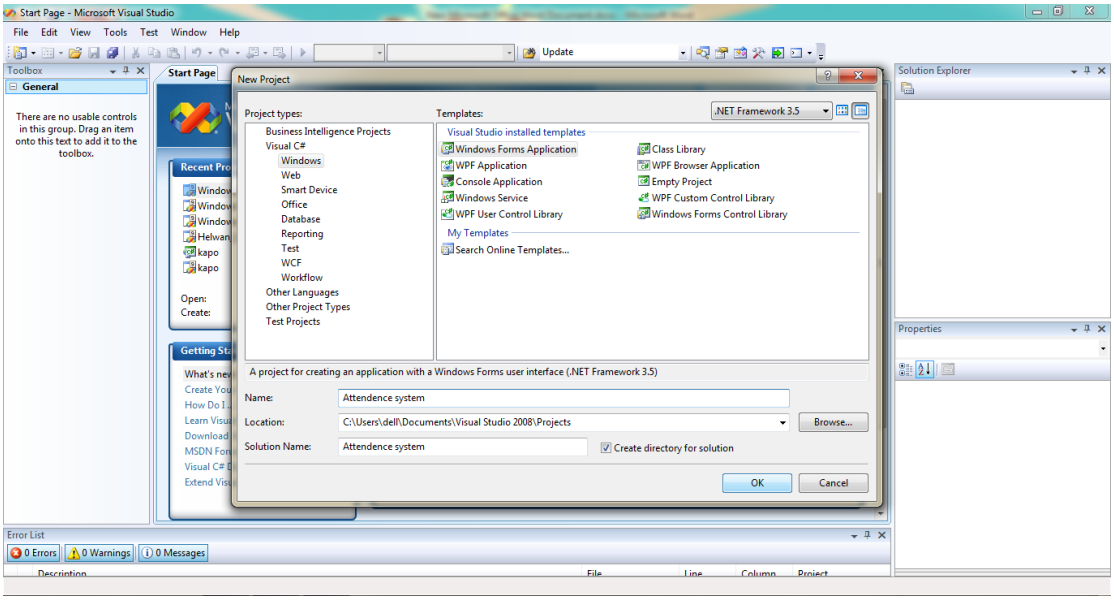


Figure 3-1b creating project

We open Microsoft visual studio 2008 then we choose file→new→project & choose visual C# then as we are working on windows form application we choose Windows→ Windows forms Application after that we name the project Grad_Proj also we specify path where the application will be saved.

3.3 LOGIN Page

Figure 3-2a welcome page

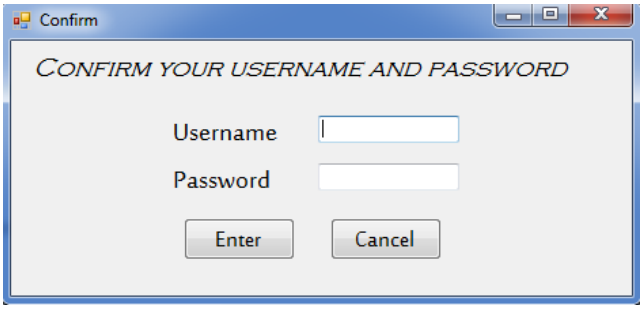


Figure 3-2b login page

As the project is based on taking id from RFID reader, we need to first create a welcome page that will take the id from reader and define which way the user will go in another words as the user pass his id the rank of the user will be determined so if the rank of the user is student it will directly go the student page.

-If the rank of the user is doctor it will go to the page where the doctor will take the attendance of his students.

-If the rank of the user is admin it will show the login page which contains username & password that will take him to the admin page.

-If the rank of the user is tutor it will take the tutor to the page where he will take the attendance of his students.

-Finally if the rank of the user is admin it will allow him to open the page of employee attendance.

3.4 Administration Menu

This menu will be accessed when user rank is administrator and the menu will look like this:

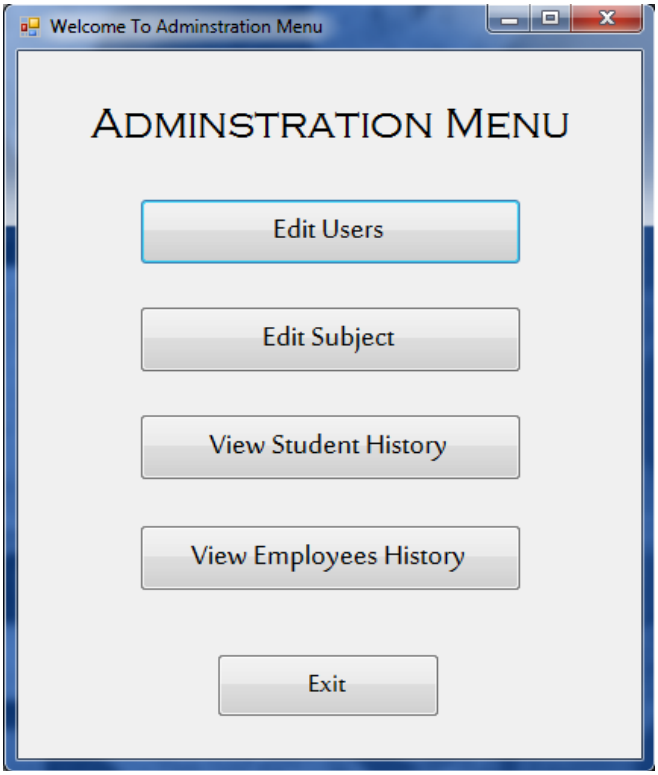


Figure 3-3a Administration Menu

The menu contains buttons: The first button for creating or updating or deleting a user.
The second button is for creating or deleting a subject.
The third button is for viewing student attendance history.
The fourth button is for viewing employee attendance history.

3.4.1 Edit users

3.4.1.1 Students Tab

By clicking on Edit users Button we will have the following form:

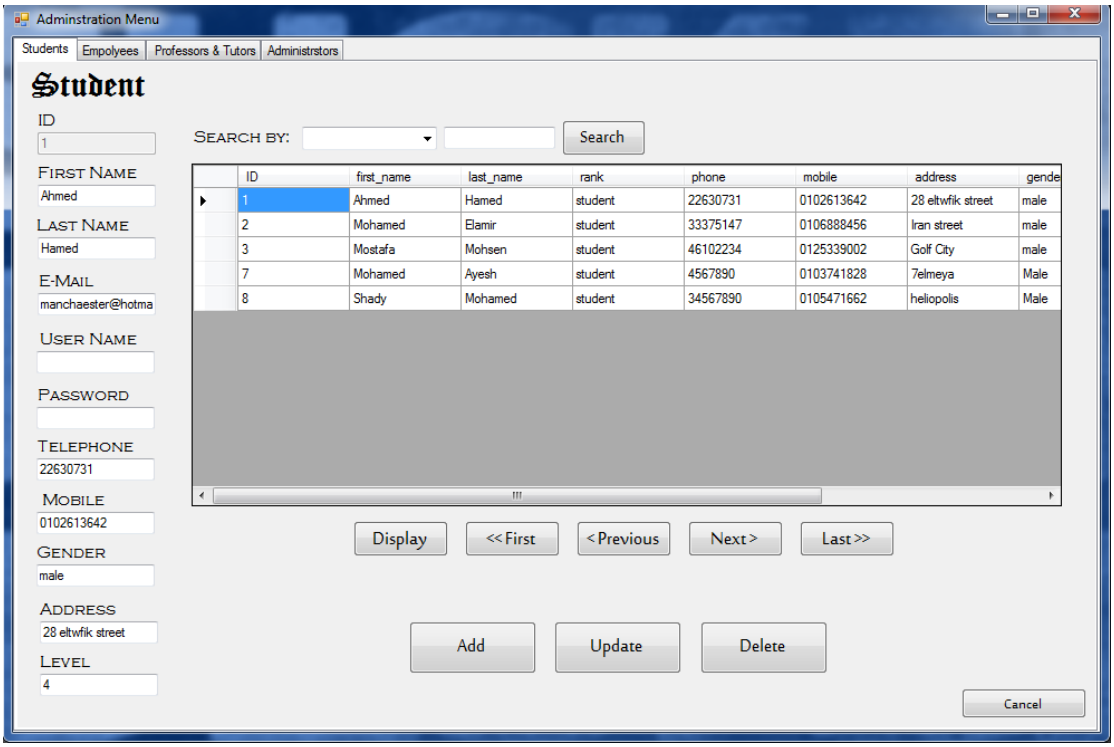
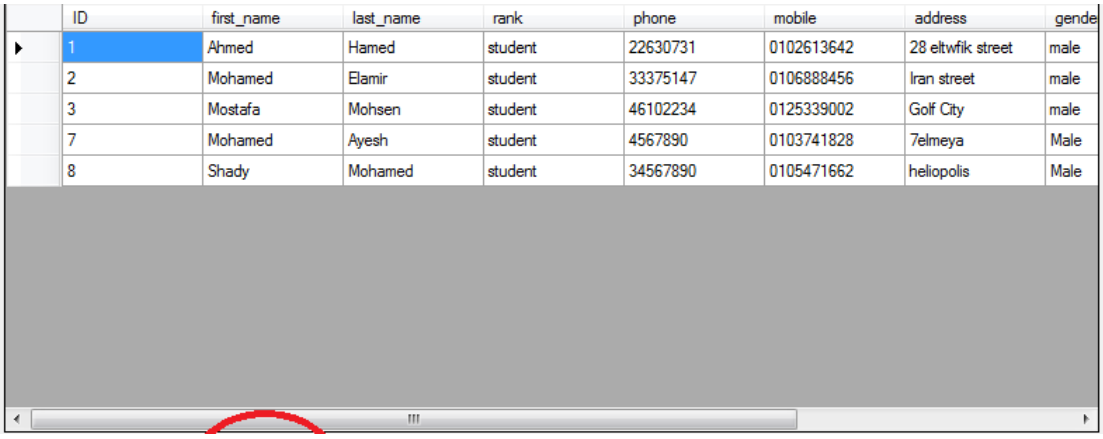


Figure 3-4b

This form is divided into tabs and each tab is divided into specific type of user that you will add or update or delete, &first we will explain the student tabs then we don't forget to connect our database with this form by defining sql library and by using code:

```
using System.Data.SqlClient;

      SqlConnection con = new SqlConnection("Data Source =
.\\SQLEXPRESS; Initial Catalog=attendance; Integrated
Security=true");
```



ID	first_name	last_name	rank	phone	mobile	address	gende
1	Ahmed	Hamed	student	22630731	0102613642	28 eltwfik street	male
2	Mohamed	Elamir	student	33375147	0106888456	Iran street	male
3	Mostafa	Mohsen	student	46102234	0125339002	Golf City	male
7	Mohamed	Ayesh	student	4567890	0103741828	7elmeya	Male
8	Shady	Mohamed	student	34567890	0105471662	heliopolis	Male

Figure 3-4c

This is called datagridview and it is used to view tables in database but only for students. The code will be as follows:

```
stu_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,rank,phone,mobile,address,gender,level,email
from users where rank='student'", con);
```

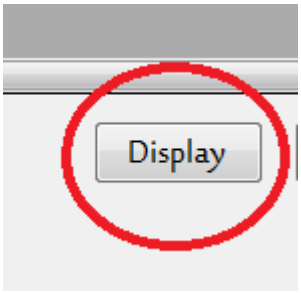


Figure 3-4d

The button Display will display all the students in the database by using the following code:

```
      stu_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender,level,email from
users where rank='student'", con);
```

Figure 3-4e

These buttons are for navigating inside the datagridview. The First button is for moving to the first row, the Previous button is for moving up in datagridview, the Next button is for moving down in the datagridview, and finally the Last button is for moving to the last row.

In case of empty datagridview and we try to navigate using any of the previous buttons there will be a message box that tells the user (There is no data in the datagridview to navigate).

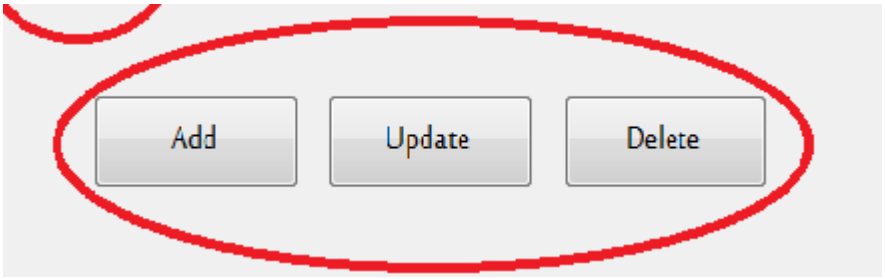


Figure 3-4f

The next three buttons that we will discuss are:

1- Add
2- Update
3- Delete

1- Add:

This form will be used for adding a student to our database. When clicking on this button the next form will appear:

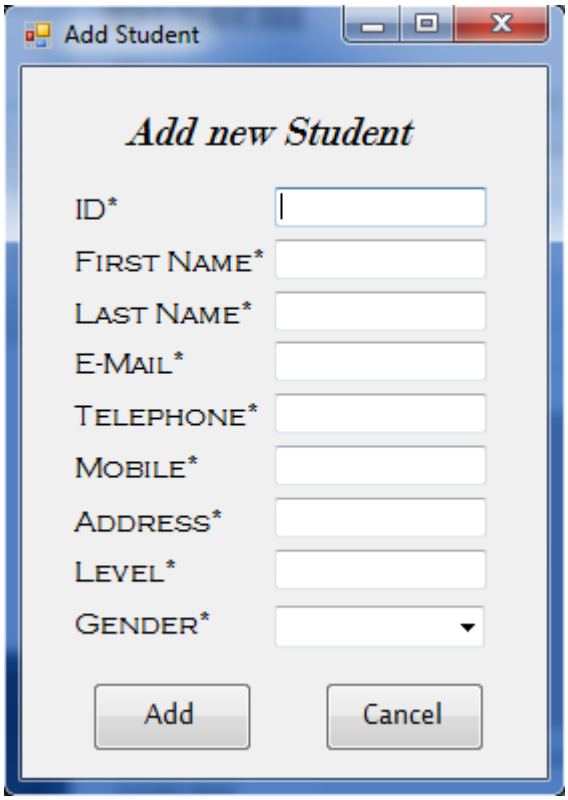


Figure 3-4g

In this form we will create a new student by providing the required information about every student.

The * means that these text boxes must be filled with data so that it can create the student in the database else there will appear a message box tells that there is some data missing. We will use the following code:

```
SqlCommand com = new SqlCommand("insert into users values ('" +
id_txt.Text + "','" + first_txt.Text + "','" + last_txt.Text + "','"
+ x + "','" + null + "','" + null + "','" + tel_txt.Text + "','" +
mob_txt.Text + "','" + add_txt.Text + "','" + gen_txt.Text + "','" +
email_txt.Text + "','" + level_txt.Text + "')", con);
```

Note: we notice in figure 3-4g that id textbox will always be locked as this textbox will read from the attached RFID reader directly and the user will not manipulate it also the lock is useful for avoiding any mistakes done by the administrators while adding the new student id.

2- Update

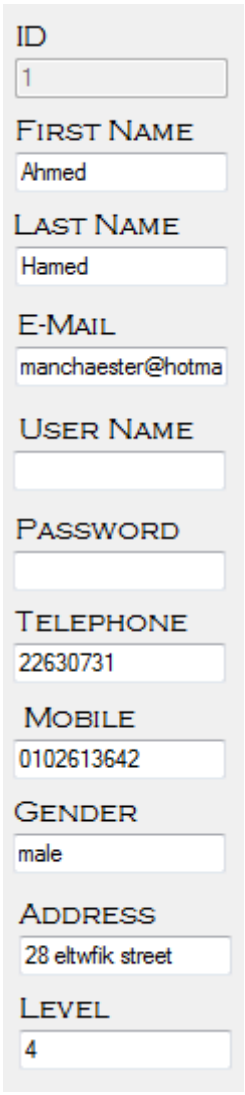


Figure 3-4h

This is appearing in the left side of the tab and it is used in updating some information about the student like updating the mobile or the telephone of the student. An example of updating is as follows:

```
SqlCommand com1 = new SqlCommand("update users set first_name='" +
stu_first_txt.Text + "' where ID='" + stu_id_txt.Text + "'", con);

SqlCommand com2 = new SqlCommand("update users set last_name='" +
stu_last_txt.Text + "'where ID='" + stu_id_txt.Text + "'", con);
```

Note: the username and password will always be empty as the student won't have one and only administrator will have a username and password.

3- Delete

From datagridview we select a specific row and then choose to delete it, so this user will be deleted permanently from all tables in database which means it will be deleted from users table and stu_sub table (will be deleted from all the subjects he/she assigned to).

```
SqlCommand com = new SqlCommand("delete from users  where ID='" +
stu_id_txt.Text + "'", con);

SqlCommand com1 = new SqlCommand("delete from stu_sub  where
stu_id='" + stu_id_txt.Text + "'", con);
```

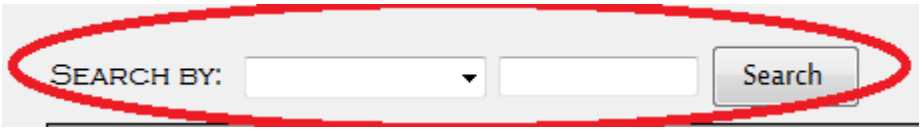


Figure 3-4i

It is existed above datagridview. It is used to search for the student in the database. It consists of one combo box, one text box and one button. The combobox will be used to specify the search category which will be Id or firstname or level and the search button is used to begin the search process.

```
stu_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender,level from users
where First_Name ='" + stu_search_txt.Text + "'and rank='student'",
con);

stu_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender,level from users
where Level ='" + stu_search_txt.Text + "'and rank='student'", con);

stu_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender,level from users
where Gender ='" + stu_search_txt.Text + "'and rank='student'", con);
```

```
stu_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender,level from users
where ID ='" + stu_search_txt.Text + "'and rank='student'", con);
```

-In case of no data meet the search specification a messagebox will appear telling the user the following:

```
MessageBox.Show("Check the subject you entered or category you
search");
```

3.4.1.2 Employees Tab

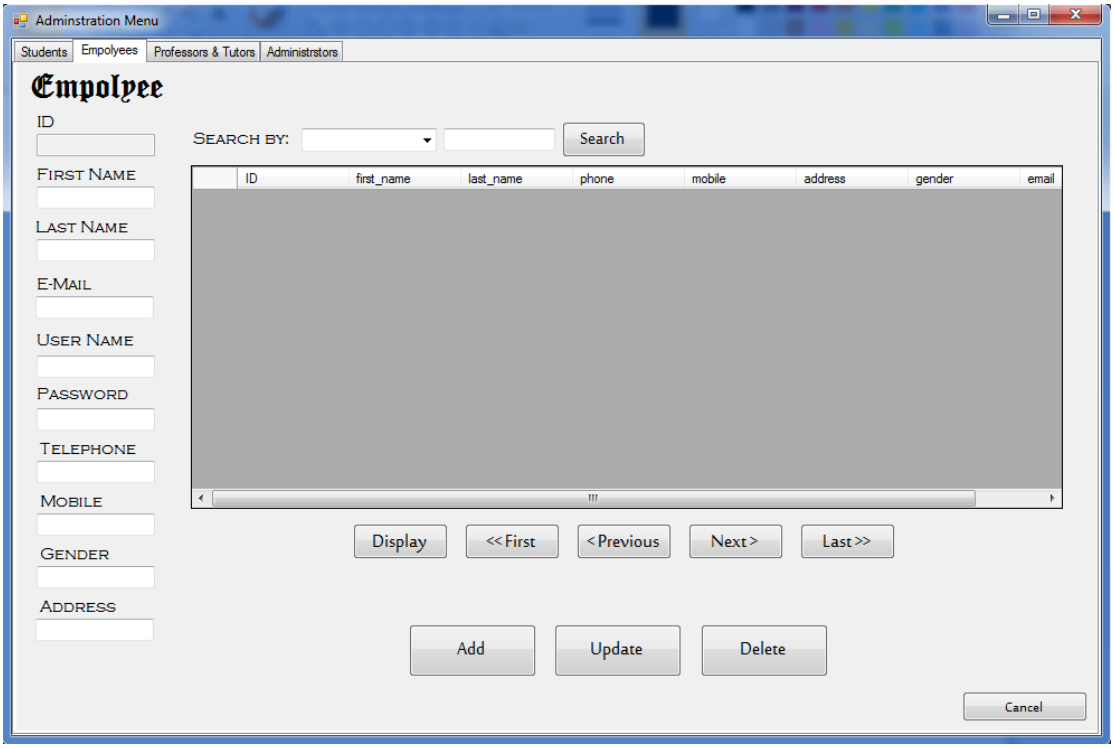
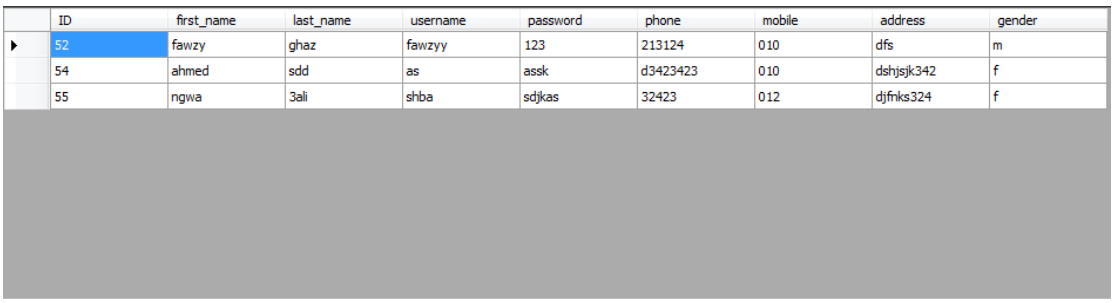


Figure 3-5a

This is the tab that is used in creating or updating or deleting an employee in the database.



ID	first_name	last_name	username	password	phone	mobile	address	gender
52	fawzy	ghaz	fawzyy	123	213124	010	dfs	m
54	ahmed	sdd	as	assk	d3423423	010	dshjsjk342	f
55	ngwa	3ali	shba	sdjkas	32423	012	djfnks324	f

Figure 3-5b

This is called datagridview and it is used to view tables in database but only for employees.

```
emp_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender,email from users
where rank='empolyee'", con);
```

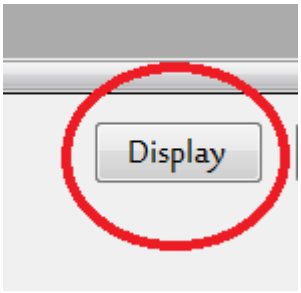


Figure 3-5c

This button will be used to display all the users who are employees. We will be using the following code:

```
emp_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender,email from users
where rank='empolyee'", con);
```

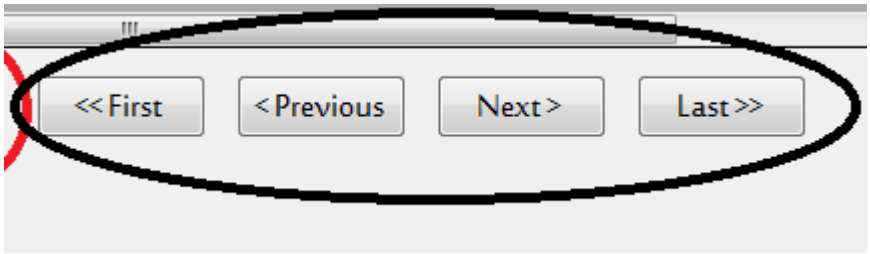


Figure 3-5d

These buttons are for navigating inside the datagridview. The First button is for moving to the first row, the Previous button is for moving up in datagridview, the Next button is for moving down in the datagridview, and finally the Last button is for moving to the last row. The code is the following in order:

```
emp_bs.MoveFirst();
            emp_dgupdate();
            emp_rows();
emp_bs.MovePrevious();
            emp_dgupdate();
            emp_rows();

emp_bs.MoveNext();
            emp_dgupdate();
            emp_rows();

emp_bs.MoveLast();
            emp_dgupdate();
```

```
emp_rows();
```

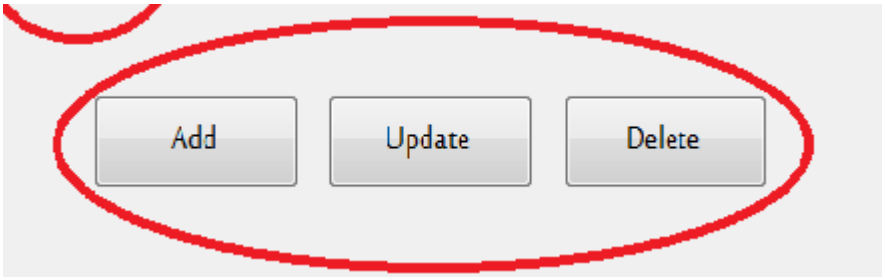


Figure 3-5e

The next three buttons that we will discuss are:

1- Add
2- Update
3- Delete

1- Add:

This form will be used for adding an employee to our database. When clicking on this button the Add Employee form will appear:

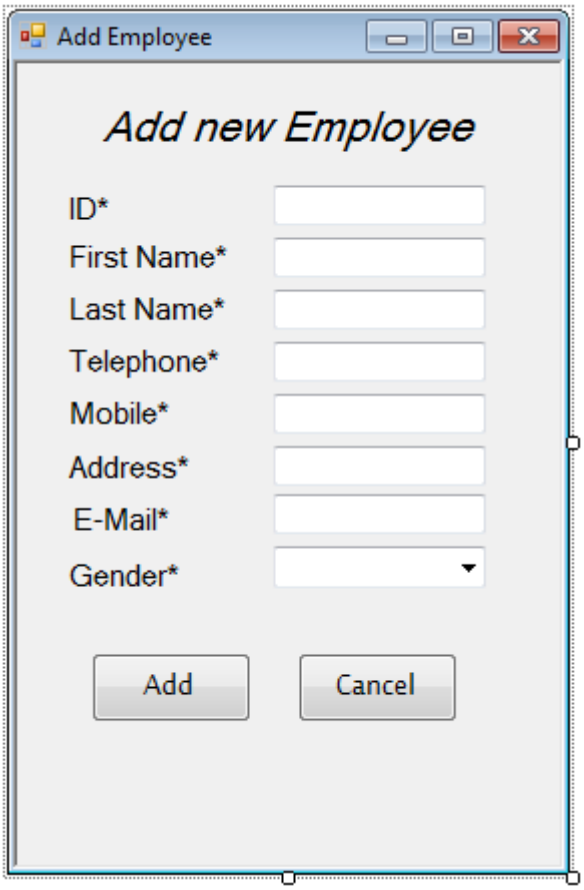


Figure 3-5f

In this form we will create a new employee by providing the required information about every employee. The * means that these text boxes must be filled with data so that it can create the employee in the database else there will appear a message box tells that there is some data missing. We will use the following code:

```
com = new SqlCommand("insert into users values ('" + id_txt.Text +
"','" + first_txt.Text + "','" + last_txt.Text + "','" + x + "','" +
null + "','" + null + "','" + tel_txt.Text + "','" + mob_txt.Text +
"','" + add_txt.Text + "','" + gen_txt.Text + "','" + email_txt.Text
+ "','" + null + "')", con);
```

Note: we notice in figure 3-5f that id textbox will always be locked as this textbox will read from the attached RFID reader directly and the user will not manipulate it also the lock is useful for avoiding any mistakes done by the administrators while adding the new employee id.

2-Update:

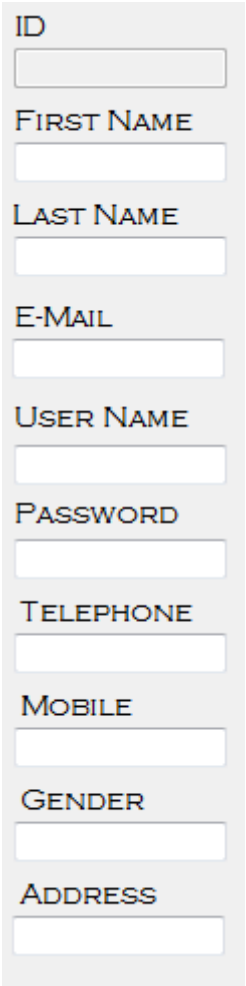


Figure 3-5g

This is appearing in the left side of the form and it is used in updating some information about the employee like updating the mobile or the telephone of the employee. For example the code will be as follows:

```
com1 = new SqlCommand("update users set first_name='" +
emp_first_txt.Text + "' where ID='" + emp_id_txt.Text + "'", con);

com2 = new SqlCommand("update users set last_name='" +
emp_last_txt.Text + "'where ID='" + emp_id_txt.Text + "'", con);

com6 = new SqlCommand("update users set phone='" + emp_tel_txt.Text +
"' where ID='" + emp_id_txt.Text + "'", con);
```

3-Delete:

From datagridview we select a specific row and then choose to delete it, so this user will be deleted entirely from database. We will use the following code:

```
com = new SqlCommand("delete from users  where ID='" +
emp_id_txt.Text + "'", con);
```

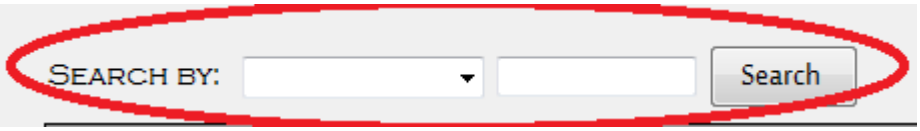


Figure 3-5h

It is existed above datagridview. It is used to search for the employee in the database. It consists of one combo box, one text box and one button. The combobox will be used to specify the search category which will be Id or firstname or gender and the search button is used to begin the search process.

```
emp_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender from users where
First_Name ='" + emp_seach_txt.Text + "'and rank='empolyee'", con);

emp_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender from users where
Gender ='" + emp_seach_txt.Text + "'and rank='empolyee'", con);

emp_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender from users  where
ID ='" + emp_seach_txt.Text + "'and rank='empolyee'", con);
```

-In case of no data meet the search specification a messagebox will appear telling the user the following:

```
MessageBox.Show("Check the subject you entered or category you
search");
```

3.4.1.3Professors & Tutors Tab

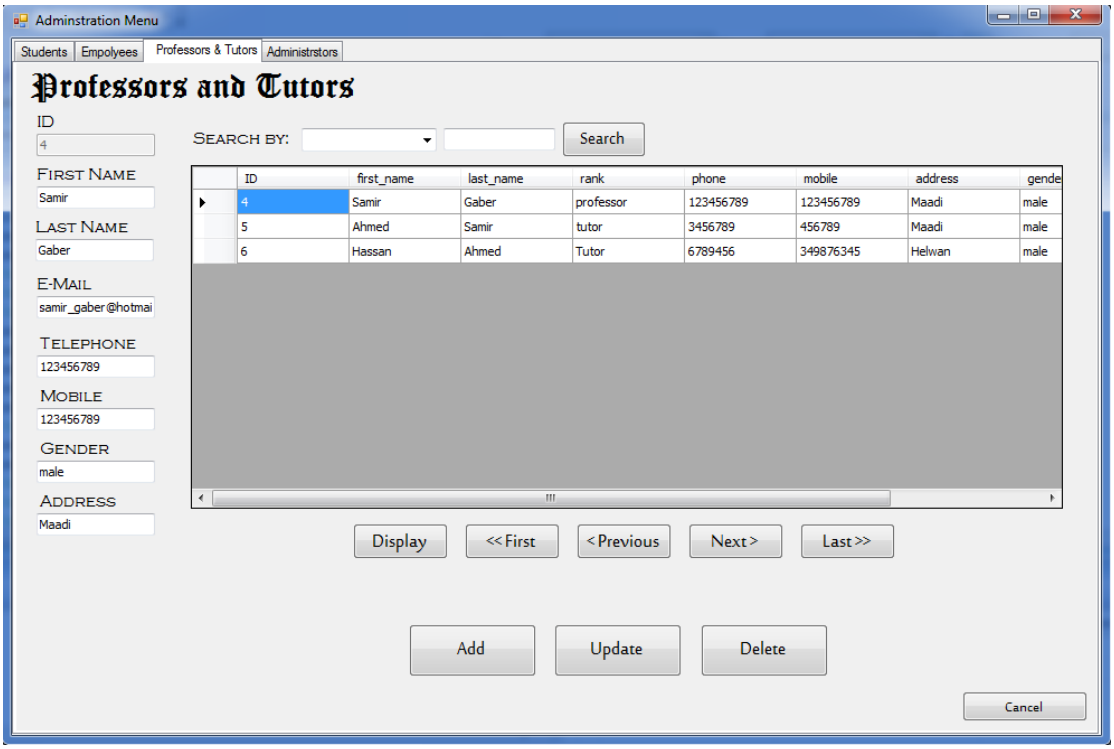


Figure 3-6a

This is the tab that is used in creating, updating and deleting the professors and tutors in the database.

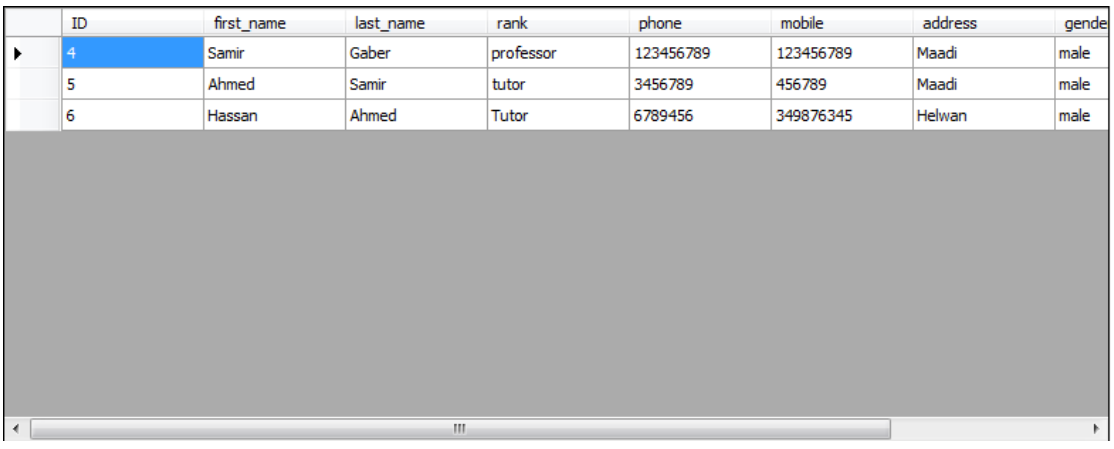


Figure 3-6b

This is called datagridview and it is used to view tables in database but only for professors and tutors. This is the following code:

```
prof_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,rank,phone,mobile,address,gender,email from
users where rank='professor' OR rank='tutor'", con);
```

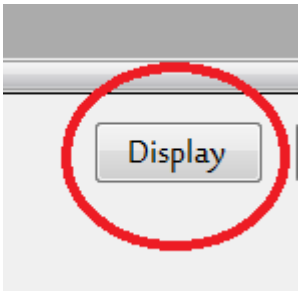


Figure 3-6c

This button will be used to display all the users who are employees. We will be using the following code:

```
prof_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,rank,phone,mobile,address,gender,email from
users where rank='professor' OR rank='tutor'", con);
```

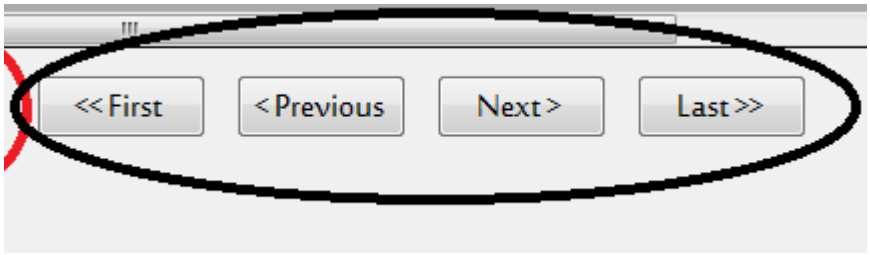


Figure 3-6d

These buttons are for navigating inside the datagridview. The First button is for moving to the first row, the Previous button is for moving up in datagridview, the Next button is for moving down in the datagridview, and finally the Last button is for moving to the last row. The following codes are as follows:

```
prof_bs.MoveFirst();
            prof_dgupdate();
            prof_rows();

prof_bs.MovePrevious();
            prof_dgupdate();
            prof_rows();

prof_bs.MoveNext();
            prof_dgupdate();
            prof_rows();

prof_bs.MoveLast();
            prof_dgupdate();
            prof_rows();
```

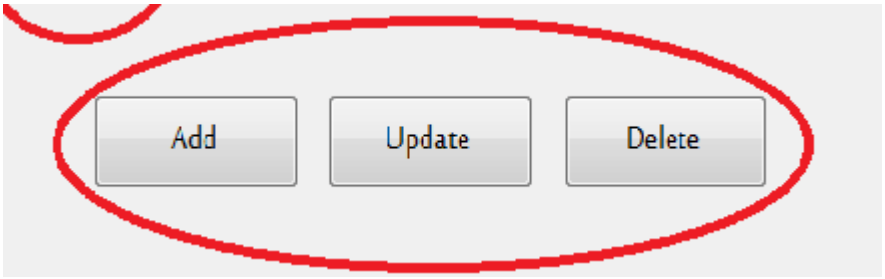


Figure 3-6e

The next three buttons that we will discuss are:

1- Add
2- Update
3- Delete

1- Add:

This form will be used for adding a professor to our database. When clicking on this button the Add professor form will appear:

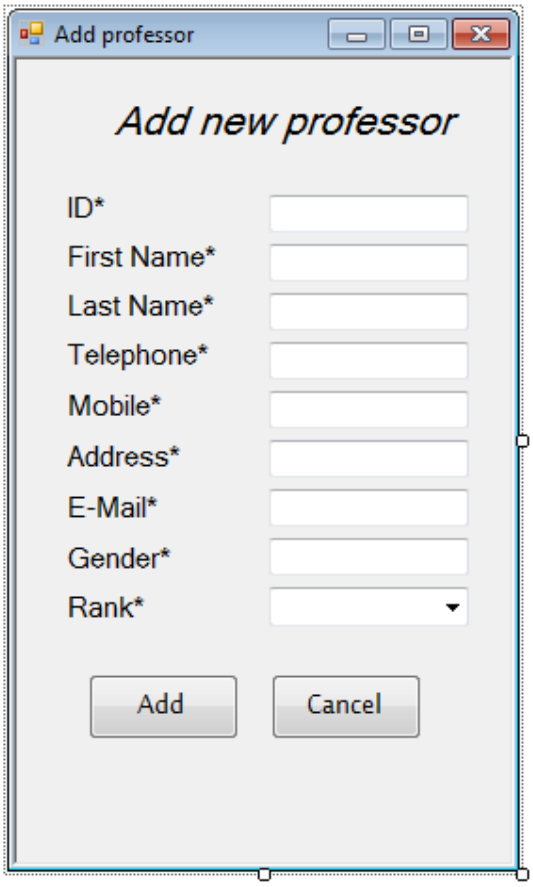


Figure 3-6f

In this form we will create a new professor by providing the required information about every professor. The * means that these text boxes must be filled with data so that it can create the professor in the database else there will appear a message box tells that there is some data missing. We will use the following code:

```
com = new SqlCommand("insert into users values ('" + id_txt.Text +
"','" + first_txt.Text + "','" + last_txt.Text + "','" +
rank_txt.Text + "','" + null + "','" + null + "','" + tel_txt.Text +
"','" + mob_txt.Text + "','" + add_txt.Text + "','" + gen_txt.Text +
"','" + email_txt.Text + "','" + null + "')", con);
```

2- Update:

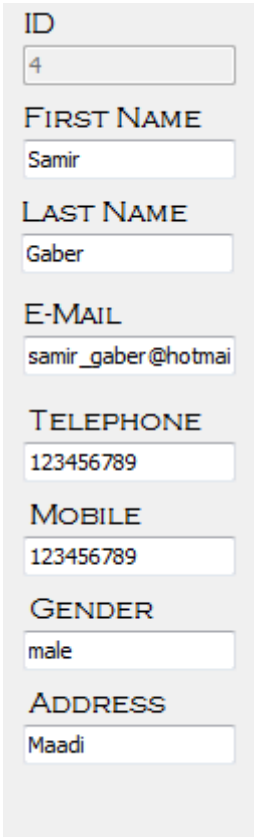


Figure 3-6g

This is appearing in the left side of the form and it is used in updating some information about the professors and tutors like updating the mobile or the telephone of the professors/tutors. For example the code will be as follows:

```
com1 = new SqlCommand("update users set first_name='" +
prof_first_txt.Text + "' where ID='" + prof_id_txt.Text + "'", con);

com2 = new SqlCommand("update users set last_name='" +
prof_last_txt.Text + "'where ID='" + prof_id_txt.Text + "'", con);

com6 = new SqlCommand("update users set phone='" + prof_tel_txt.Text
+ "' where ID='" + prof_id_txt.Text + "'", con);
```

3- Delete:

From datagridview we select a specific row and then choose to delete it, so this user will be deleted entirely from database. The code is as following:

```
com = new SqlCommand("delete from users  where ID='" +
prof_id_txt.Text + "'", con);

com1 = new SqlCommand("delete from teach_sub  where teach_id='" +
prof_id_txt.Text + "'", con);
```

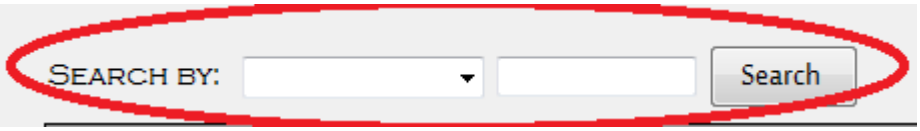


Figure 3-6h

It is existed above datagridview. It is used to search for the professors and tutors in the database. It consists of one combo box, one text box and one button. The combobox will be used to specify the search category which will be Id or firstname and the search button is used to begin the search process. The code is as following:

```
prof_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender from users where
first_name='" + prof_search_txt.Text + " and 'rank='professor'",
con);

prof_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,phone,mobile,address,gender from users where
id='" + prof_id_txt.Text + "'rank='professor'", con);
```

-In case of no data meet the search specification a messagebox will appear telling the user the following:

```
MessageBox.Show("Check the subject you entered or category you
search");
```

3.4.1.4 Administrators

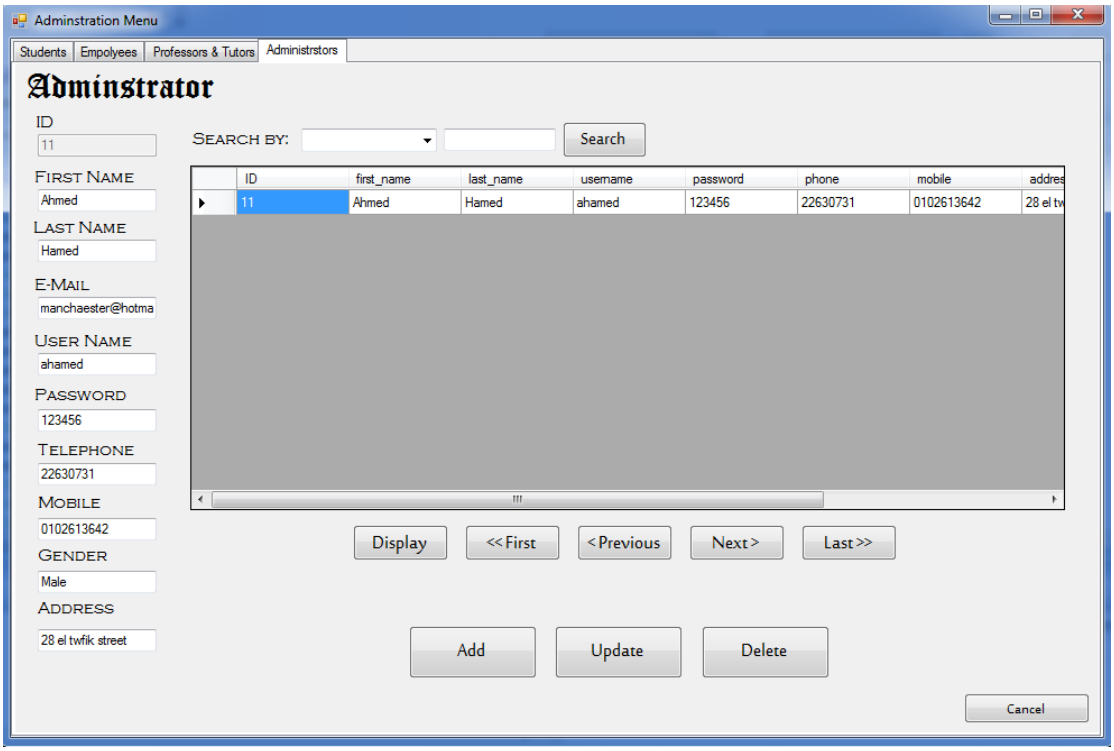


Figure 3-7a

This is the tab that is used in creating, deleting and updating the administrators in the database.

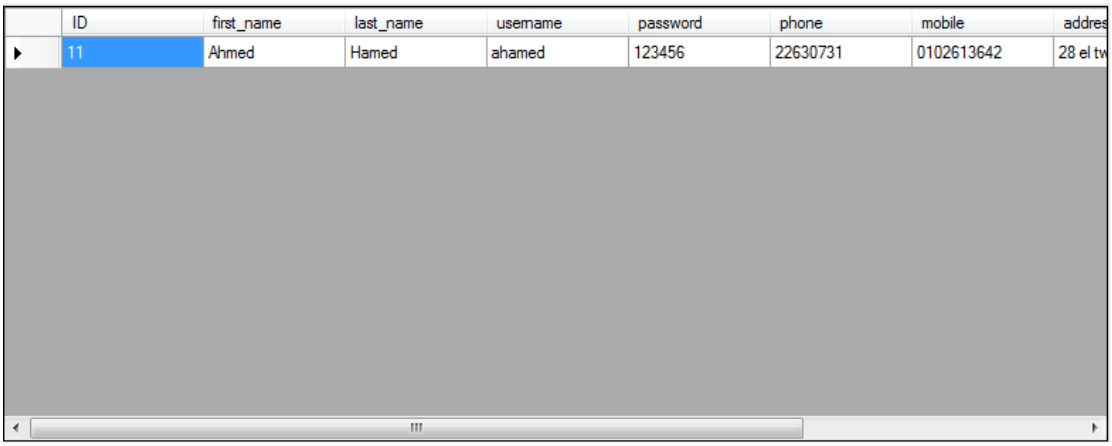


Figure 3-7b

This is called datagridview and it is used to view tables in database but only for administrators. The code will be the following:

```
ad_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,username,password,phone,mobile,address,gender
,email from users where rank='admin'", con);
```

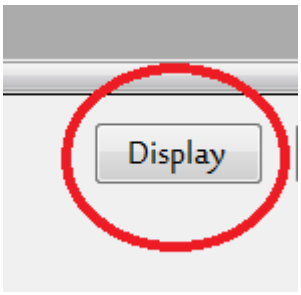


Figure 3-7c

This button will be used to display all the users who are employees. We will be using the following code:

```
ad_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,username,password,phone,mobile,address,gender
,email from users where rank='admin'", con);
```

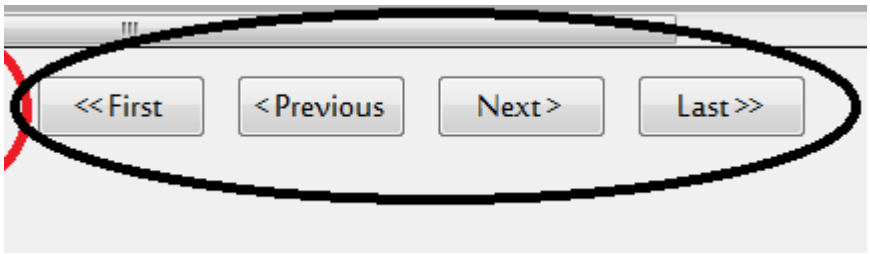


Figure 3-7d

These buttons are for navigating inside the datagridview. The First button is for moving to the first row, the Previous button is for moving up in datagridview, the Next button is for moving down in the datagridview, and finally the Last button is for moving to the last row.

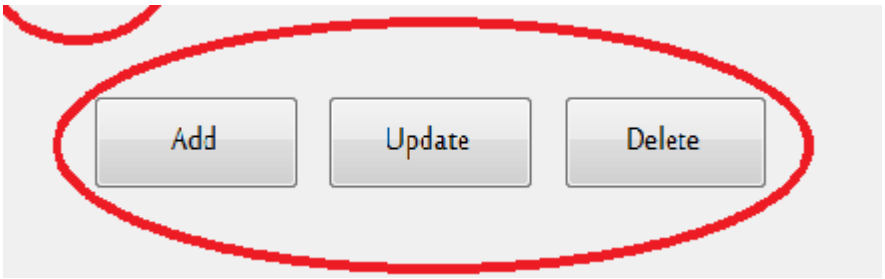


Figure 3-7e

The next three buttons that we will discuss are:

1- Add
2- Update
3- Delete

1- Add:

This form will be used for adding an administrator to our database. When clicking on this button the Add Admin form will appear:

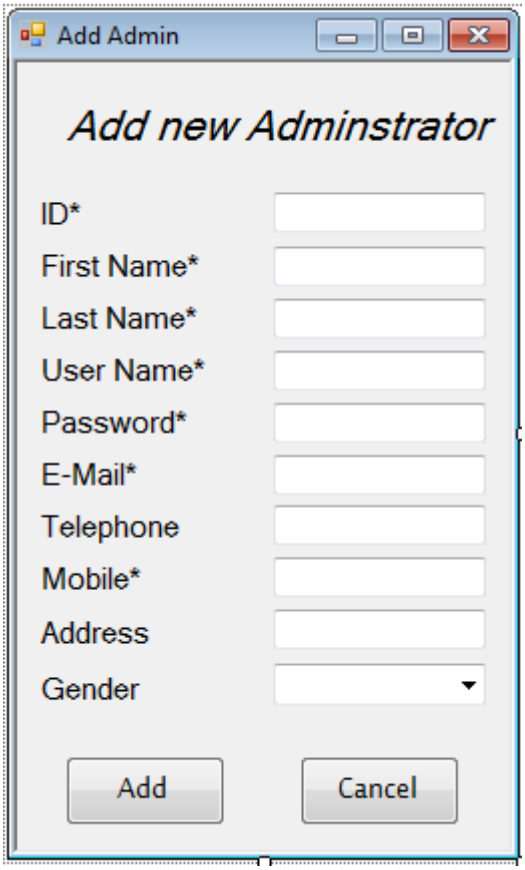


Figure 3-7f

In this form we will create a new student by providing the required information about every student. The * means that these text boxes must be filled with data so that it can create the student in the database else there will appear a message box tells that there is some data missing. We will use the following code:

```
com = new SqlCommand("insert into users values ('" + id_txt.Text +
"','" + first_txt.Text + "','" + last_txt.Text + "','" + x + "','" +
user_txt.Text + "','" + pass_txt.Text + "','" + tel_txt.Text + "','"
+ mob_txt.Text + "','" + add_txt.Text + "','" + gen_txt.Text + "','"
+ email_txt.Text + "','" + null + "')", con);
```

Note: we notice that here we have to provide a username and password for the administrator and that is to improve security and not to allow any unauthorized users to access the database with no permission.

2- Update:

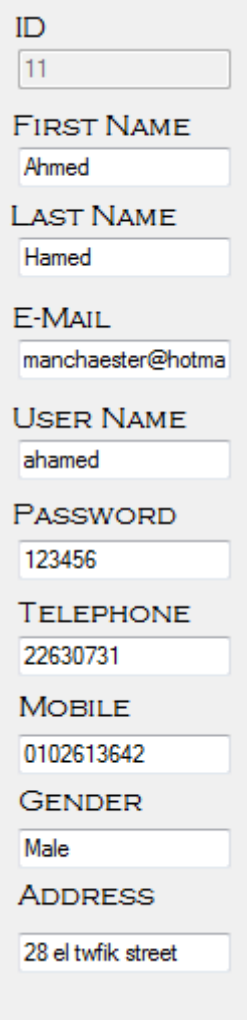


Figure 3-7g

This is appearing in the left side of the form and it is used in updating some information about the administrator like updating the mobile or the telephone of the administrator. The code is as follows:

```
com1 = new SqlCommand("update users set first_name='" +
ad_first_txt.Text + "' where ID='" + ad_id_txt.Text + "'", con);

com2 = new SqlCommand("update users set last_name='" +
ad_last_txt.Text + "'where ID='" + ad_id_txt.Text + "'", con);

com3 = new SqlCommand("update users set email='" + ad_email_txt.Text
+ "' where ID='" + ad_id_txt.Text + "'", con);

com4 = new SqlCommand("update users set username='" +
ad_user_txt.Text + "' where ID='" + ad_id_txt.Text + "'", con);
```

3- Delete:

From datagridview we select a specific row and then choose to delete it, so this user will be deleted entirely from database. The code will be as follows:

```
com = new SqlCommand("delete from users  where ID='" + ad_id_txt.Text
+ "'", con);
```

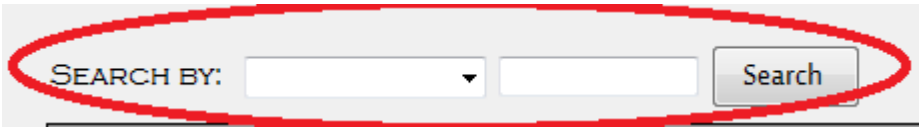


Figure 3-7h

It is existed above datagridview. It is used to search for the administrators in the database. It consists of one combo box, one text box and one button. The combobox will be used to specify the search category which will be Id or firstname and the search button is used to begin the search process. The code is as following:

```
ad_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,username,password,phone,mobile,address,gender
from users where first_name='" + ad_search_txt.Text + "' and
rank='admin'", con);

ad_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,username,password,phone,mobile,address,gender
from users where gender='" + ad_search_txt.Text + "'and
rank='admin'", con);

ad_da.SelectCommand = new SqlCommand("select
ID,first_name,last_name,username,password,phone,mobile,address,gender
from users where id='" + ad_search_txt.Text + "'and rank='admin'",
con);
```

-In case of no data meet the search specification a messagebox will appear telling the user the following:

```
MessageBox.Show("Check the subject you entered or category you
search");
```

3.4.2 Edit Subject

After we defined the shape of the Edit Users button we move on to the next button which is Edit Subject.
-When we press on Edit Subject button the forms in figure 3-8a and 3-8b will appear as follows:

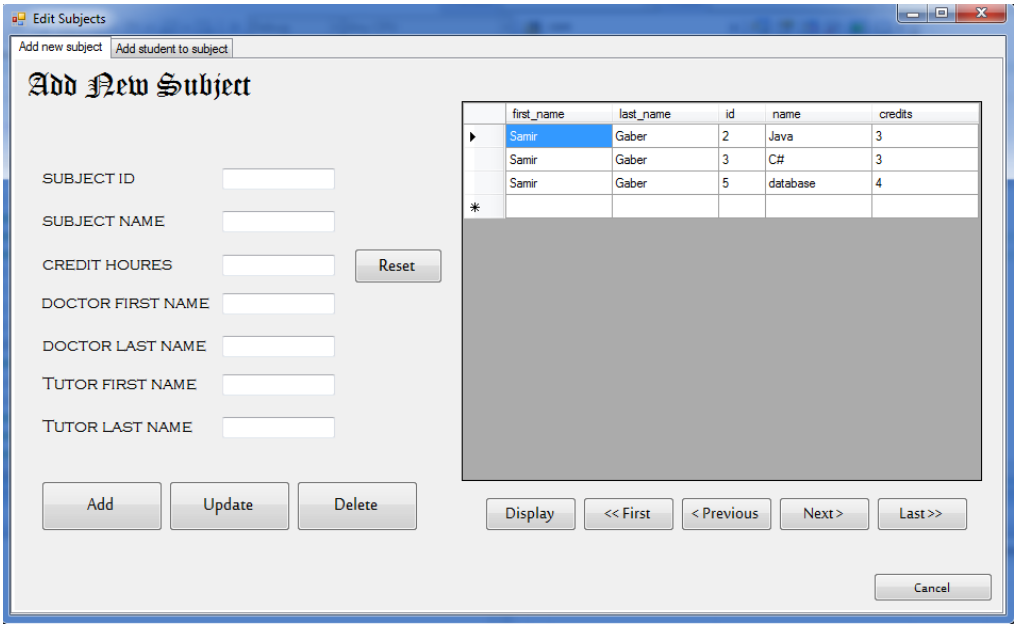


Figure 3-8a

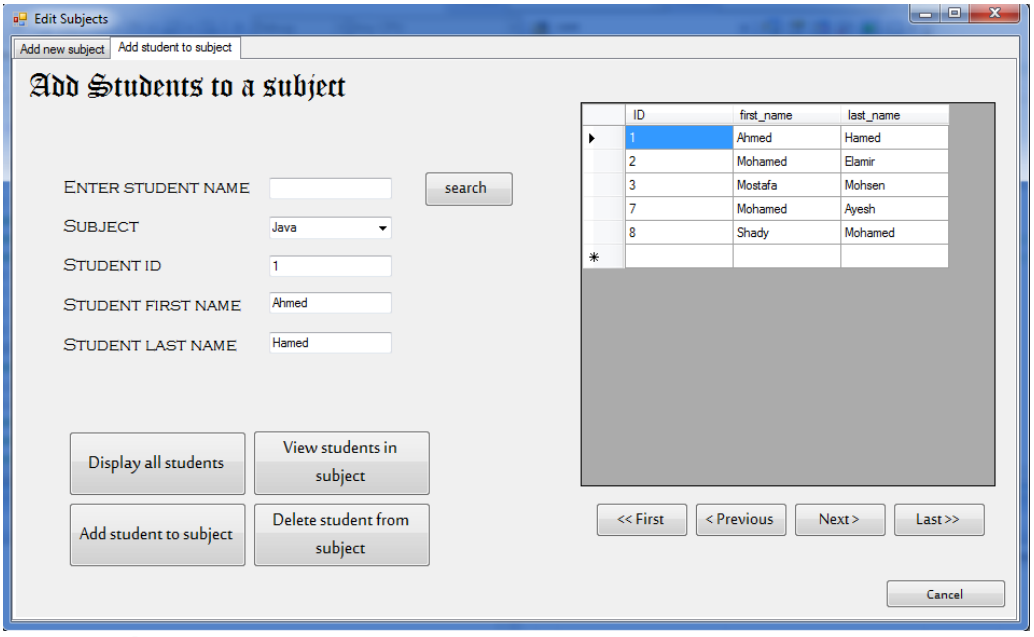


Figure 3-8b
These figures consist of two tabs the first will be to add new subject and the other tab will be for adding a student to a specific subject.

-We will now discuss the two tables in details:

3.4.2.1 Add New Subject

Here we will define the elements of the form as follows:

3.4.2.1.1- Datagridview

3.4.2. 1.2- Display button

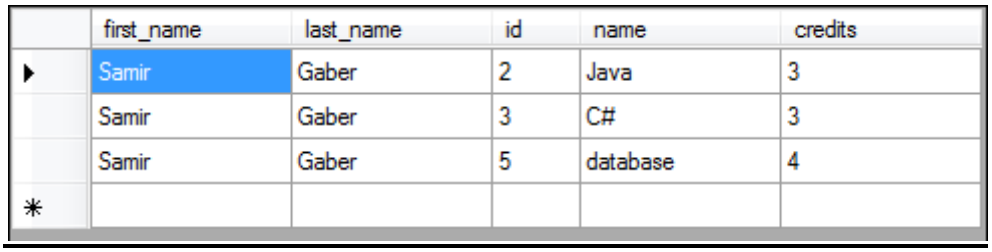
3.4.2. 1.3- First, previous, next, last

3.4.2. 1.4- Labels & textboxes

3.4.2. 1.5- Add, Delete, and Update buttons

3.4.2. 1.6- Reset button

3.4.2.1.1- Datagridview



	first_name	last_name	id	name	credits
▶	Samir	Gaber	2	Java	3
	Samir	Gaber	3	C#	3
	Samir	Gaber	5	database	4
*					

Figure 3-9a

-This datagridview is used to view the tables from the database. In the previous datagridviews represented in this chapter we didn't have a problem just selecting some columns from specific table but right here we have a challenge of selecting specific columns from three tables and connect them together to appear in the datagridview.
- The code will be as follows:

```
      da.SelectCommand = new SqlCommand("select
users.first_name,users.last_name,
subjects.id,subjects.name,subjects.credits from
subjects,users,teach_sub where teach_sub.sub_id=subjects.id and
teach_sub.teach_id=users.id", cs);
```

The code says that we will select the firstname, lastname from users table and id, name, credit hours from subjects table and that will happen in the condition that id in the subjects table matches the id of the subjects in teach_sub table and id in the users matches the id of the teach_id in the teach-sub table.

3.4.2. 1.2- Display button

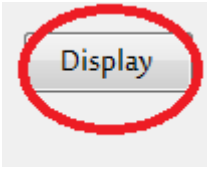


Figure 3-9b

-This button is used to display and view all data from tables in database at any time.

- The code will be as follows:

```
      da.SelectCommand = new SqlCommand("select
users.first_name,users.last_name,
subjects.id,subjects.name,subjects.credits from
subjects,users,teach_sub where teach_sub.sub_id=subjects.id and
teach_sub.teach_id=users.id", cs);
```

where cs is the connection to the database.

3.4.2. 1.3- First, previous, next, last

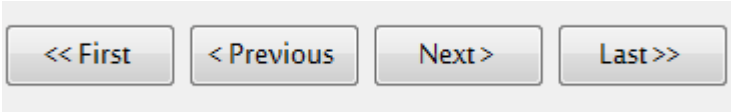


Figure 3-9c

These four buttons is important in navigating through the datagridview. The first button will be used for moving to the first row, the previous button is for moving up in the datagridview, the Next button is for moving down in the datagridview, and finally the Last button is for moving to the last row.

3.4.2. 1.4- Labels & textboxes

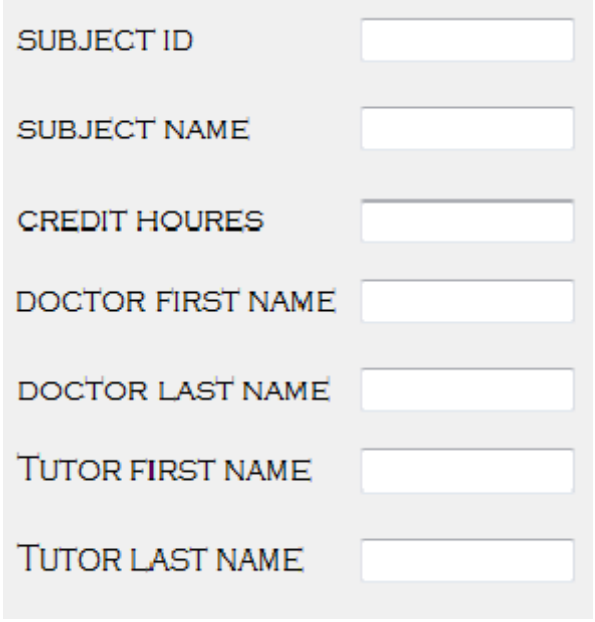


Figure 3-9d

The labels and text boxes here will be bounded to the datagridview so when adding some information in these textboxes, it will have an effect in the datagridview and subsequently in the database of the system.

3.4.2. 1.5- Add, Delete, and Update buttons

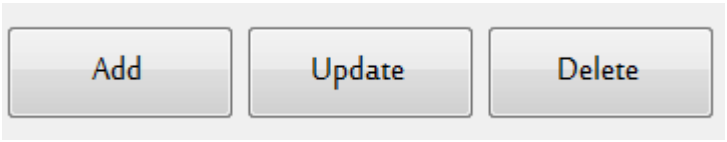


Figure 3-9e

These three buttons represent the artery of the form as the entire from can be summarized to what these buttons will do and also the need for their codes to be correct and not having any problems using them.

1- Add:

The add button will be used for adding a new subject, so to add subjects we insert the subject information in the textboxes

-We will select id from users by inserting in the textboxes the first and lastname of the doctor and the same for tutors as it will select id from users by inserting in the textboxes the first and lastname of the tutor
-We then insert into teach_sub table id of the doctor and the id of the subject
-We then insert into tutorial_sub table id of the doctor and the id of the subject
-And finally in the table of subjects we insert subject id from textbox, the name of the subject and its credit hours

```
SqlCommand comm = new SqlCommand("select id from users where
first_name ='" + doctor1_txt.Text + "'and last_name='" +
doctor2_txt.Text + "'", cs);

SqlCommand comm2 = new SqlCommand("select id from users where
first_name ='" + tutor1_txt.Text + "'and last_name='" +
tutor2_txt.Text + "'", cs1);

SqlCommand comand = new SqlCommand("insert into Teach_Sub values('" +
x + "','" + sub_id_txt.Text + "',null)", cs);

Where x is the id of the doctor saved in the reader.

SqlCommand command = new SqlCommand("insert into tutorial_sub
values('" + y + "','" + sub_id_txt.Text + "',null)", cs);

Where y is the id of the tutor saved in the reader.

com = new SqlCommand("insert into Subjects values('" +
sub_id_txt.Text + "','" + sub_txt.Text + "','" + credit_txt.Text +
"')", cs);
```

2- Update

-This button will be used for updating some information about any subject

-To update the subjects table we let subject name in the database take subject name from textbox, we also let credits column to take information from credit textbox and that's when id column equal to the subject id from textbox

-And to update the subject information in teach_sub table we identify the doctor's firstname and lastname and then it will get the id of the doctor from the users table. And then we get teach_id from users table according to the first code written and that's when sub_id equal to subject id in textbox

```
SqlCommand comm = new SqlCommand("select id from users where
first_name ='" + doctor1_txt.Text + "'and last_name='" +
doctor2_txt.Text + "'", cs);

com = new SqlCommand("update subjects set name='" + sub_txt.Text +
"',credits='" + credit_txt.Text + "' where id='" + sub_id_txt.Text +
"'", cs);

SqlCommand updatedoc = new SqlCommand("update teach_sub set
teach_id='" + x + "' where sub_id='" + sub_id_txt.Text + "'", cs);
```

3- Delete

-The delete button will be used to delete a subject from the database.

- Here we will face a problem of deleting subject from the whole database and as the subject is used in more than one table in the database we face a problem of deleting it from every place in the database system.

- In this database the subject will be deleted from 3 places or tables which are subjects table, teach_sub table, and stu_sub table

-Note: the stu_sub was included here as it includes all of the subject's ids that the students are registered in and so deleting the any subject id we will have to delete all the student's ids.

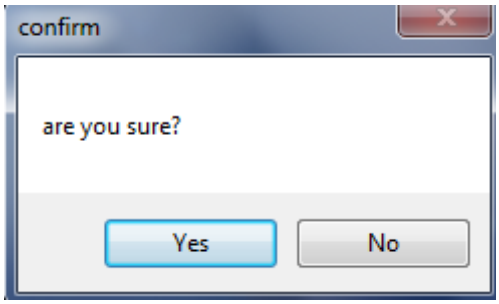
-The codes that will be used to perform deletion operation are:

```
SqlCommand delete1 = new SqlCommand("delete from stu_sub where
Sub_id='" + sub_id_txt.Text + "'", cs);

SqlCommand delete2 = new SqlCommand("delete from Teach_Sub where
sub_id='" + sub_id_txt.Text + "'", cs);

SqlCommand delete3 = new SqlCommand("delete from Subjects where id='"
+ sub_id_txt.Text + "'", cs);
```

- As the deletion process is a very critical operation we have to show the user a message box to confirm the deletion operation.

```
MessageBox.Show("Are you sure?", "confirm", MessageBoxButtons.YesNo);
```

-And the message box will look like this in this figure:



3.4.2. 1.6- Reset button

Figure 3-9f

This button will be used to clear textboxes from the information it has so when adding another subject we have a clear boxes. It will be very useful for the user who is typing and updating this information so he/she won't be confused. For example a subject could be assigned to the wrong tutor or doctor.

```
sub_id_txt.Clear();
            sub_txt.Clear();
            doctor1_txt.Clear();
            doctor2_txt.Clear();
            credit_txt.Clear();
```

3.4.2.2 Add Student to A new Subject

Here we will define the elements of the form as follows:

3.4.2.2.1- Datagridview

3.4.2.2.3- Display all students button

3.4.2.2.4- View students in subject

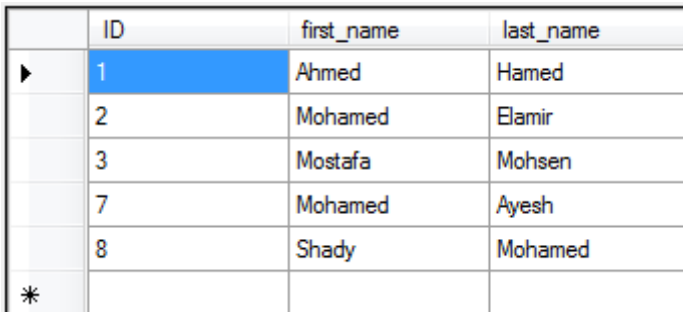
3.4.2.2.5- Add student to subject

3.4.2.2.6- Delete student from subject

3.4.2.2.7- Labels, textboxes, comboboxes

3.4.2.2.8- Search button

3.4.2.2.1- Datagridview



	ID	first_name	last_name
▶	1	Ahmed	Hamed
	2	Mohamed	Elamir
	3	Mostafa	Mohsen
	7	Mohamed	Ayesh
	8	Shady	Mohamed
*			

Figure 3-10a

-This datagridview is used to view the tables from the database.

- Here we will display students according to two different codes depending on some specifications that will be discussed in further details later.

```
da2.SelectCommand = new SqlCommand("SELECT ID,first_name,last_name
from users where rank='student'", cs);

SqlCommand select = new SqlCommand("select id from subjects where
name='" + comboBox1.Text + "'", cs);
```

3.4.2.2.2- First, previous, next, last

Figure 3-10b

These four buttons is important in navigating through the datagridview. The first button will be used for moving to the first row, the previous button is for moving up in the datagridview, the Next button is for moving down in the datagridview, and finally the Last button is for moving to the last row. For example we have the following code:

```
bs.MoveFirst();
            dgupdate2();
```

And the same will be applied to the other three buttons with respect to their function.

3.4.2.2.3- Display all students button

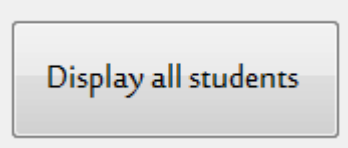


Figure 3-10c

- This button will be used to display all the students from users table that exist in the database system. The code will be as follows:

```
da2.SelectCommand = new SqlCommand("SELECT ID,first_name,last_name
from users where rank='student'", cs);
```

3.4.2.2.4- View students in subject

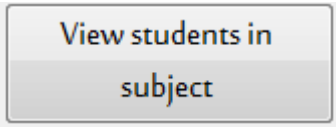
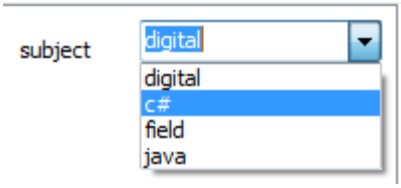


Figure 3-10d

- This button is slightly different from the previous button as it will be working with two tables in the database, the first one will be users table and the second one will be stu_sub table.

- We will select the user id, firstname, lastname from users table and we will compare it with the student id in the stu_sub table and filter the results by specifying the subject id that have specific number of students in it and view it.

- Summarizing what we have just said is that we will view the ids of the students in a specific subject and that will be done by selecting it from the combo box that contains all the subjects and by selecting one subject from combo box, the students registered in this subject will appear in the datagridview.




```
da2.SelectCommand = new SqlCommand("select
users.id,users.first_name,users.last_name from users,stu_sub where
stu_sub.stu_id=users.id and stu_sub.Sub_id='" + k + "' ", cs);

SqlCommand select = new SqlCommand("select id from subjects where
name='" + comboBox1.Text + "'", cs);
```

3.4.2.2.5- Add student to subject

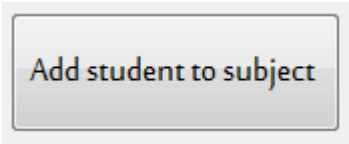


Figure 3-10e

-This button will be used for adding a specific student to a specific subject.

- This will be done by first selecting a subject name from combo box and that will then select id from subjects table in the database and after that we will insert the subject id and student id from textbox into the stu_sub table in the database.

-we should consider getting the user attention when he is missing some data when adding the student to a subject and that will be done by showing the user a message box that says there is a missing data and he/she should insert it to complete the registration process.

- When the user has successfully entered the correct information and added the student successfully. He/she will have a message box telling that the registration process has been completed successfully.

```
SqlCommand select = new SqlCommand("select id from subjects where
name='" + comboBox1.Text + "'", cs);


SqlCommand ins_stu = new SqlCommand("insert into stu_sub values('" +
v + "','" + id_txt.Text + "',null,null,null)", cs);
```

-Where (v) is the subject id.

3.4.2.2.6- Delete student from subject

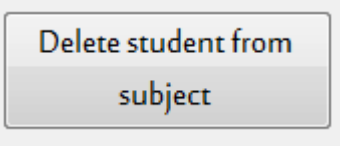


Figure 3-10f

- This button will be used to delete a specific student from a specific subject

- Now to delete a student from the subject we first select a subject name from combo box and that will then select id of subject from subjects table in the database.

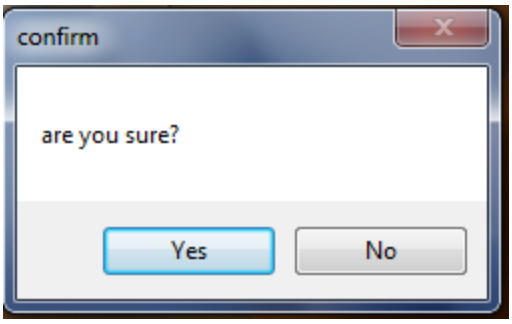
- Then we delete the student from the stu_sub table where student id is equal the student id in the textbox and the subject name/id is equal that in the combo box.

```
SqlCommand select = new SqlCommand("select id from subjects where
name='" + comboBox1.Text + "'", cs);

delete = new SqlCommand("delete from stu_sub where stu_id='" +
id_txt.Text + "' and sub_id='" + v + "' ", cs);
```

- As the deletion process is a very critical operation we have to show the user a message box to confirm the deletion operation.

```
MessageBox.Show("Are you sure?", "confirm", MessageBoxButtons.YesNo);
```

-And the message box will look like this in this figure:



3.4.2.2.7- Labels, textboxes, comboboxes

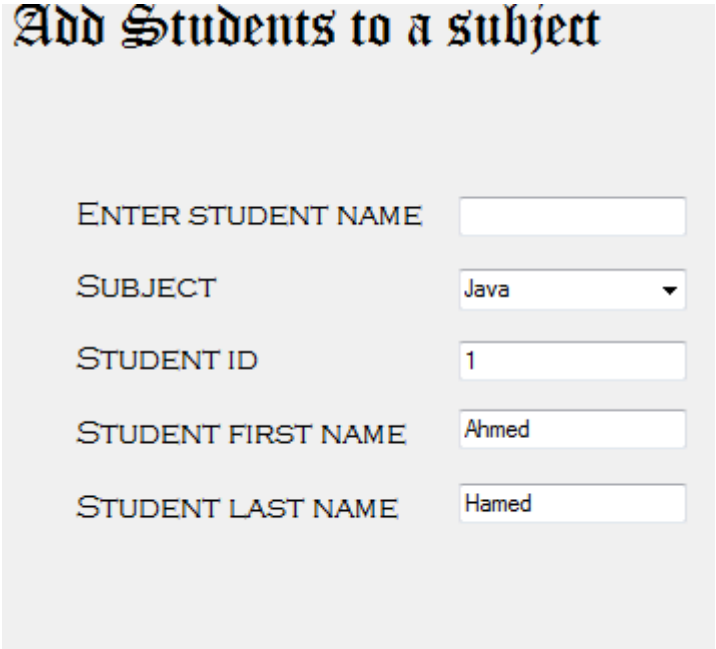
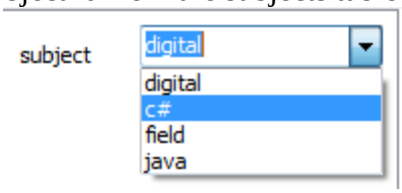


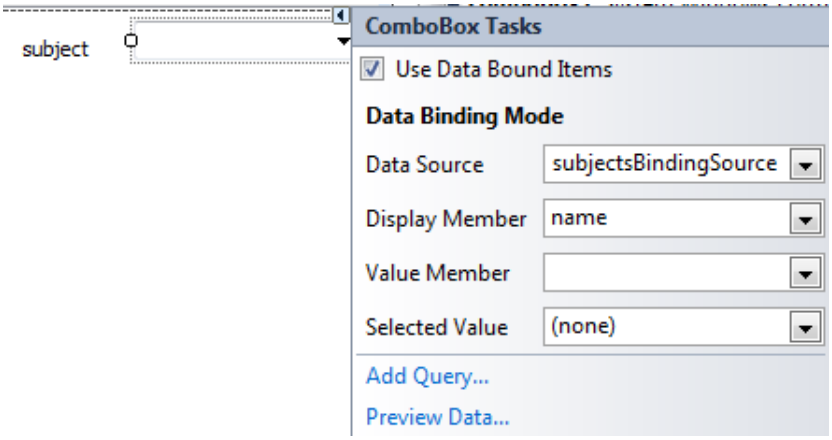
Figure 3-10g

- These labels and textboxes are bounded to the datagridview.

For combo box:

-The subject id we can get it by selecting the subject id from the subjects table where the subject name selected from the combo box.





The combo box is bounded by the column (name) in the table (subject) and this is the easier way to connect combo box to the database.

3.4.2.2.8 Search button:

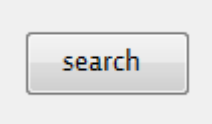


Figure 3-10h

-The search button is used to search for a certain student from the users table

-The condition used is that the first name is match with the text box and the rank is student.

-The sql command used is as follows:

```
new SqlCommand("select  first_name,last_name,id from users where
first_name='" + search_txt.Text + "' and rank='student' ", cs);
```

3.4.3 View Student History

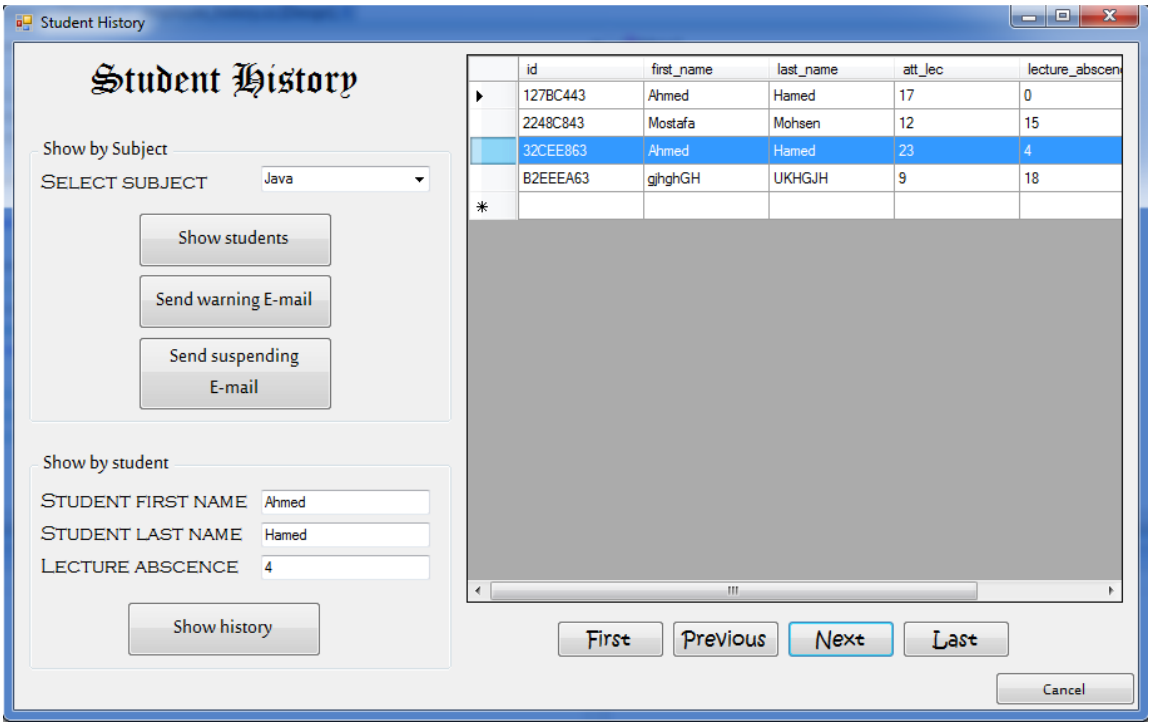


Figure 3-11a

- This form will provide two search engines in which the first will be used to show the whole students attendance history in a specific subject and the other one will be for checking a specific student's attendance. The form will also provide the feature of sending the student a warning mail to inform the student that he will pass his absence percentage also it will provide the feature of sending the student a suspending mail telling him that he is suspended from a specific subject.

	id	first_name	last_name	att_lec	lecture_abscen
▶	127BC443	Ahmed	Hamed	17	0
	2248C843	Mostafa	Mohsen	12	15
	32CEE863	Ahmed	Hamed	23	4
	B2EEEA63	gjhghGH	UKHGJH	9	18
*					

Figure 3-11b

-This datagridview is used to view the tables from the database.

- The datagridview will show id, firstname, lastname, att_lec, lecture_absence, att_tut of some student from users table, subjects table, and stu_sub table and that's when the user id will be equal to the one in student id in stu_sub table and subjects id will be equal to the id of the subjects in stu_sub table and we will get the name of the subject from a combo box that exists in our form and the code will be written as follows:

```
da.SelectCommand = new SqlCommand("select
users.id,first_name,last_name,att_lec,lecture_abscence,att_tut from
users,subjects, stu_sub where users.id=stu_sub.stu_id AND
subjects.id=stu_sub.sub_id AND subjects.name='" + comboBox1.Text +
"'", con);
```

`lecture_abscence`: is lecture number – lecture attended as we put the lecture absence, lecture number and lecture attended as integers.

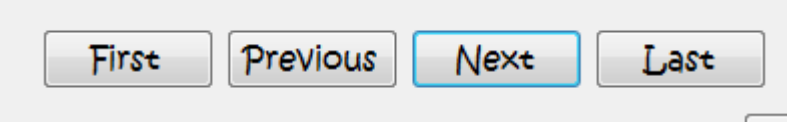


Figure 3-11c

These four buttons is important in navigating through the datagridview. The first button will be used for moving to the first row, the previous button is for moving up in the datagridview, the Next button is for moving down in the datagridview, and finally the Last button is for moving to the last row.

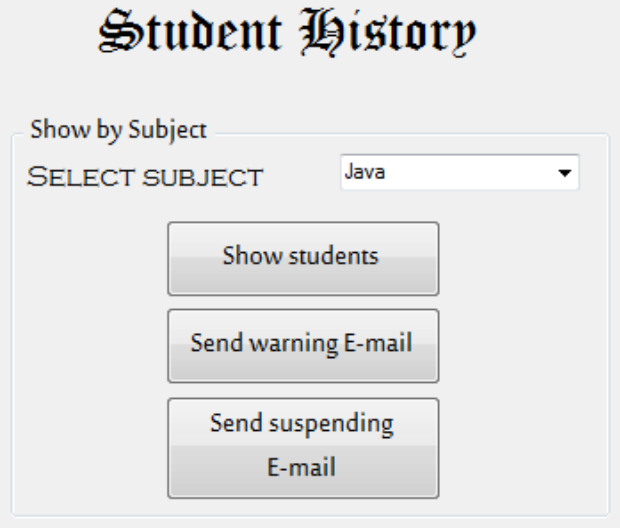


Figure 3-11d

- This is the first group box in the form and it will contain the combo box that is connected to the database and three buttons in which the first one will be to display all the students in a specific subject, the second button will be used to send a warning e-mail to a student telling him that he will pass his absence percentage and the third button will be used for telling the student that he has passed his absence percentage and will be suspended from this subject.

- The code is as follows:

→ for first button.

```
da.SelectCommand = new SqlCommand("select
users.id,first_name,last_name,att_lec,lecture_abscence,att_tut from
users,subjects, stu_sub where users.id=stu_sub.stu_id AND
subjects.id=stu_sub.sub_id AND subjects.name='" + comboBox1.Text +
"'", con);
```

→For second button

```
SmtpClient client = new SmtpClient("smtp.gmail.com", 587);
client.Credentials = new
NetworkCredential("manchaester@gmail.com","ahmedhamed123456" );
            MailMessage msg = new MailMessage();
            msg.To.Add(new MailAddress(email));
            msg.From = new MailAddress("manchaester@gmail.com");
            msg.Subject = "warning";
msg.Body = "Dear student'" + textBox1.Text + "' '" + textBox2.Text +
"' \n you have been abscent in subject:'"+comboBox1.Text+"' for
'"+textBox3+"' lecture \n ";
```

- Here we define how the mail will be sent.
We first define the server of the mail which will be Gmail.com. then we define sender mail that the student will receive his warning from then we create an object from mail message class and then put the parameters of the mail in which we define the subject, the sender and the receiver of the mail and then create a formal structure that will be

sent to any student warning him about passing his absence percentage in a certain subject

→For third button

```
SmtpClient client = new SmtpClient("smtp.gmail.com", 587);
client.Credentials = new NetworkCredential("manchaester@gmail.com",
"ahmedhamed123456");
            MailMessage msg = new MailMessage();
            msg.To.Add(new MailAddress(email));
            msg.From = new MailAddress("manchaester@gmail.com");
            msg.Subject = "Helwan University - Faculty of Engineering
- Aminstration";
            msg.Body = "Dear student'" + textBox1.Text + "' '" +
textBox2.Text + "' \n you have been suspend from subject:'" +
comboBox1.Text + "' because you have been abscence for '" +
textBox3.Text + "' lecture \n ";
```

- Here it will be the same as sending warning mail but the body of the mail will be for suspending the student not warning him.

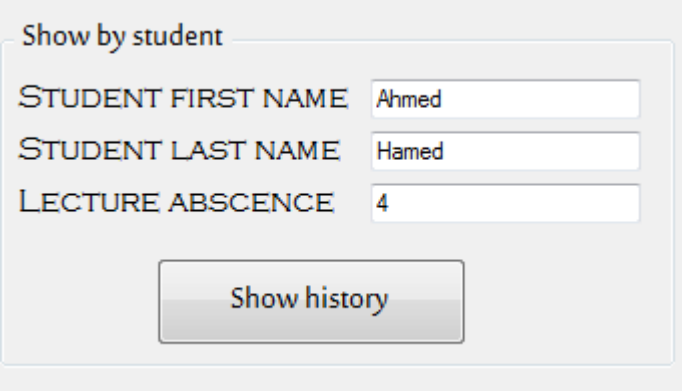


Figure 3-11e

- Here we will show the absence of a specific student in specific subject and that's in a condition that text boxes of student first and lastname is not equal as well as the subject's combo box

```
if (textBox1.Text != "" && textBox2.Text != "" && comboBox1.Text !=
"")
            {
                data.SelectCommand = new SqlCommand("select * from
Stu_History where Stu_FName ='" + textBox1.Text + "' AND Stu_LName='"
+ textBox2.Text + "' AND Sub_Name='" + comboBox1.Text + "'", con);
```

3.4.4 View Employee History

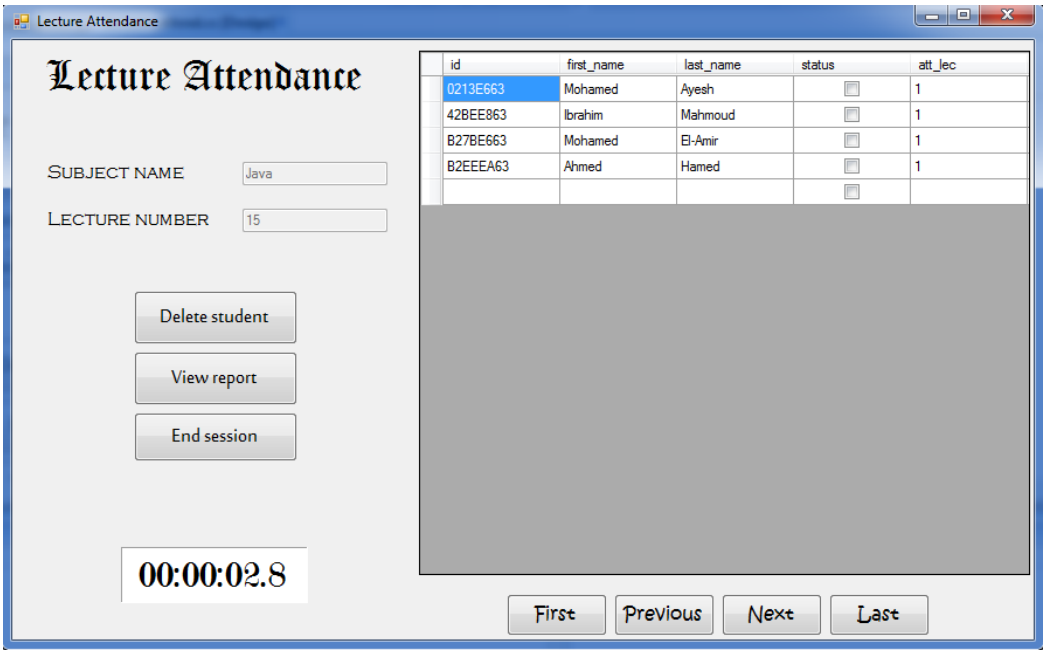


Figure 3-12a

This form will be used to show employee attendance history in the whole year and that will be useful for setting their wages and fines.

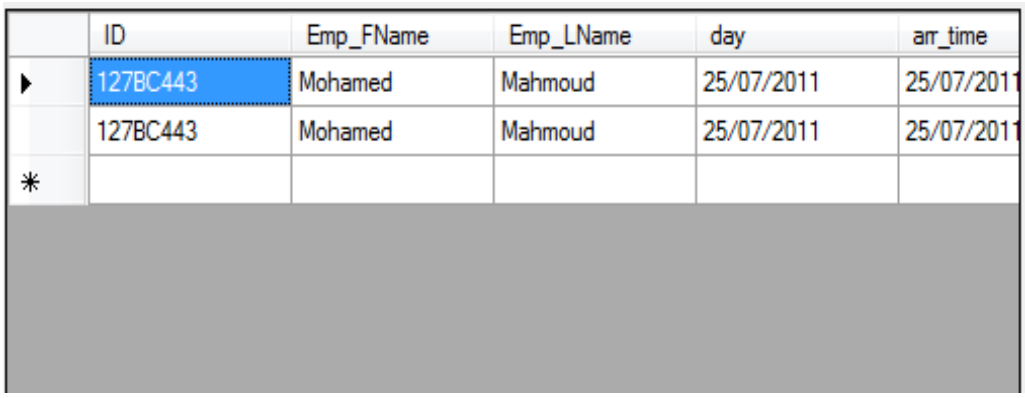


Figure 3-12b

- This is the datagridview that will show all employees' data and it will be connected to tables in the database.

- It will read from the database using two different buttons.

Figure 3-12c

- The text boxes will show the employee first and lastname but the combo box will be set from its properties to read from the calendar in the computer and so by clicking on any date. The datagridview will show all the employees attended in this date.

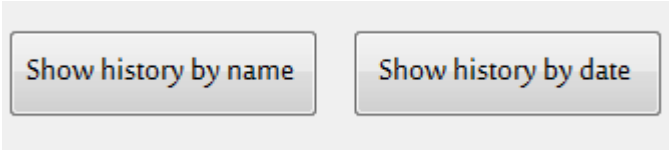


Figure 3-12d

-By clicking on any of these buttons. They will show the employee filtered by name or by date.

- The following code will explain this as:

```
if (textBox2.Text!="" || textBox1.Text!="")
            {
            da.SelectCommand = new SqlCommand("select * from
emp_history where emp_fname='" + textBox2.Text + "' and emp_lname='"+
textBox1.Text+"'   ORDER BY day ", cs);
```

- it means that by clicking the button the datagridview will show the history of all the employees that have the first name equal to some value and then will arrange them by date.

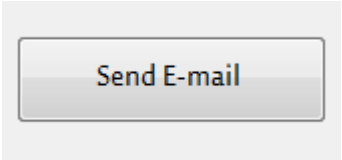


Figure 3-12e

- By clicking on this form, it will show the mail form in which an email will be sent manually.

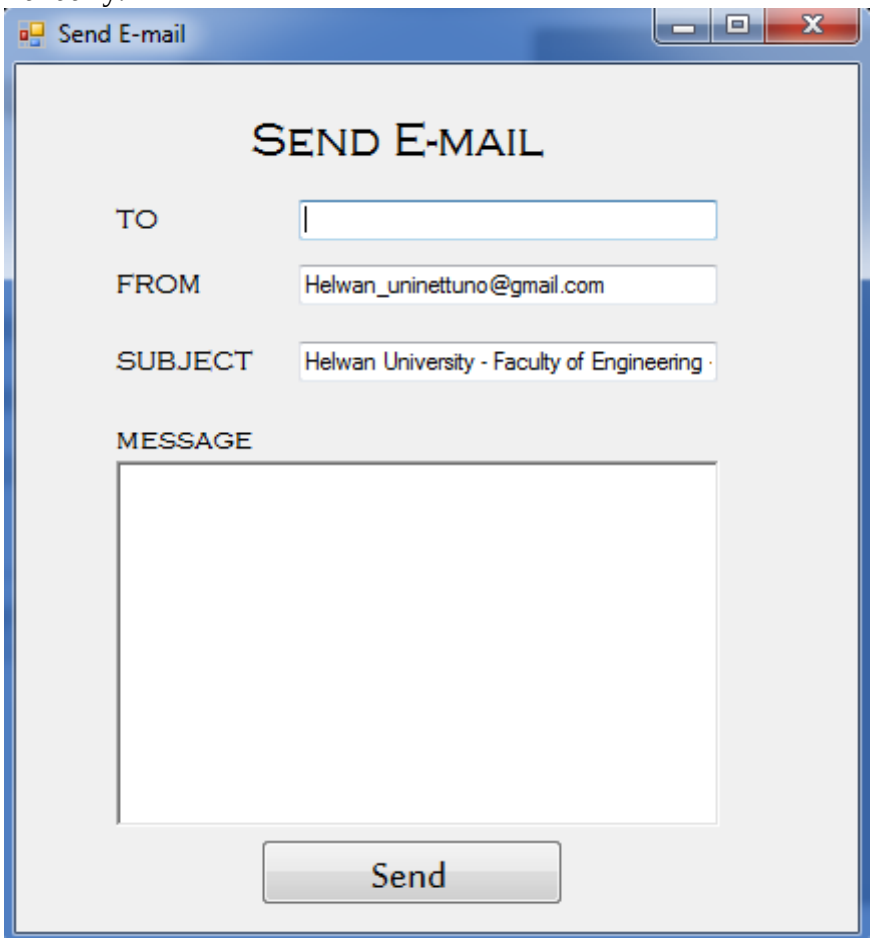


Figure 3-12f

- This is the form of the email that will be used for sending an email to the employee. This one is differ from the student as the one of the student will be automatic mail but this will be controlled manually.

3.5 Taking Student Attendance by Doctor or Tutor

-Now what if the reader read an id which is made for a doctor….

- When the reader reads a doctor's id, it will automatically open the Lecture attendance page and that is where the doctor will take the attendance of his students in the class.

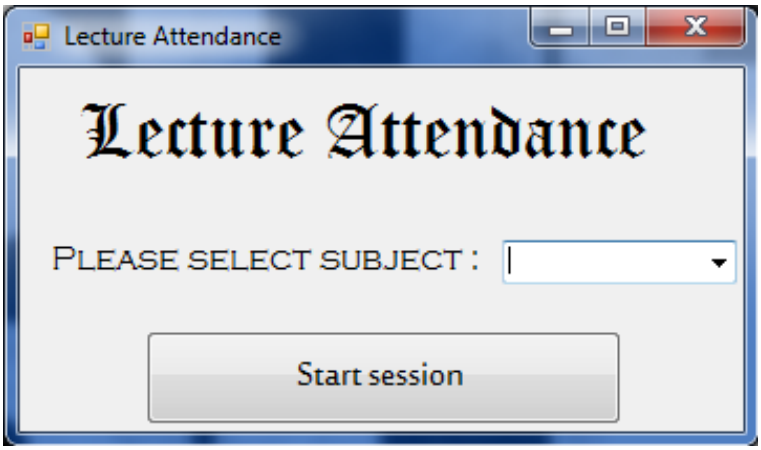


Figure 3-13a

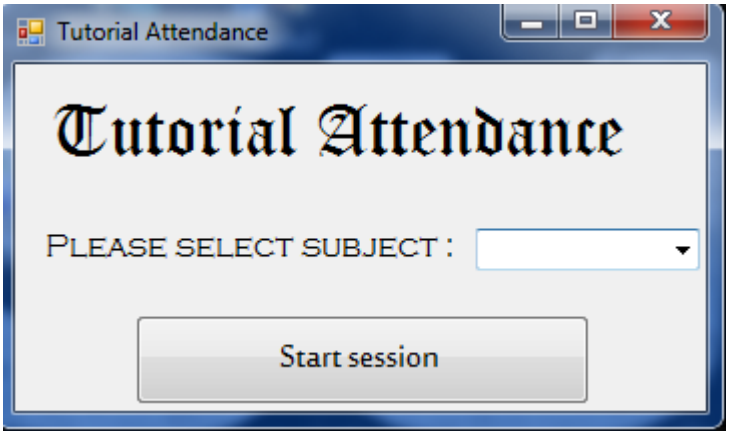


Figure 3-14

-These forms in figure 3-13a and 3-14 will appear automatically after we insert the id in the reader

-Note that: all of the following will be repeated for tutor.
- These forms have a combo box and a button:

First: combo box is used for selecting a specific subject. This combo box is linked to the subjects' table in the database.

Figure 3-13b

- Combo box won't have any codes linked to database but we will connect it with a more simple way to our database and this way will be described in the following figures:

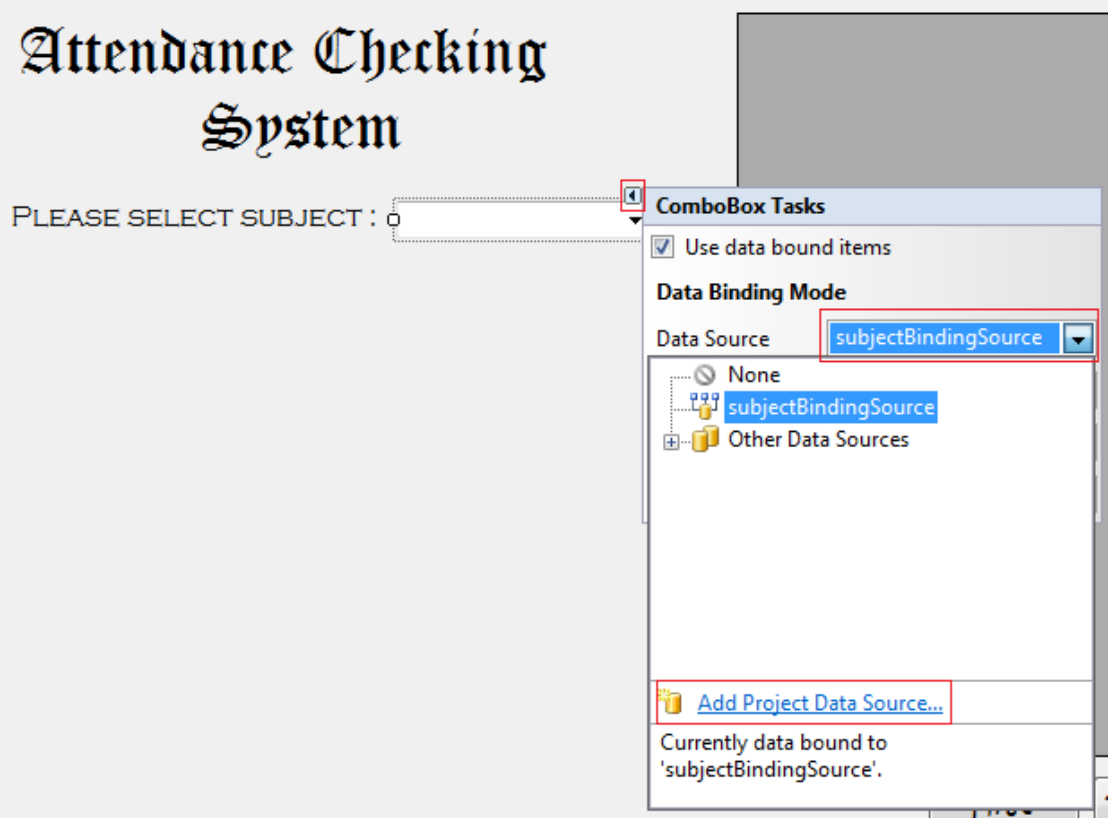


Figure 3-13c

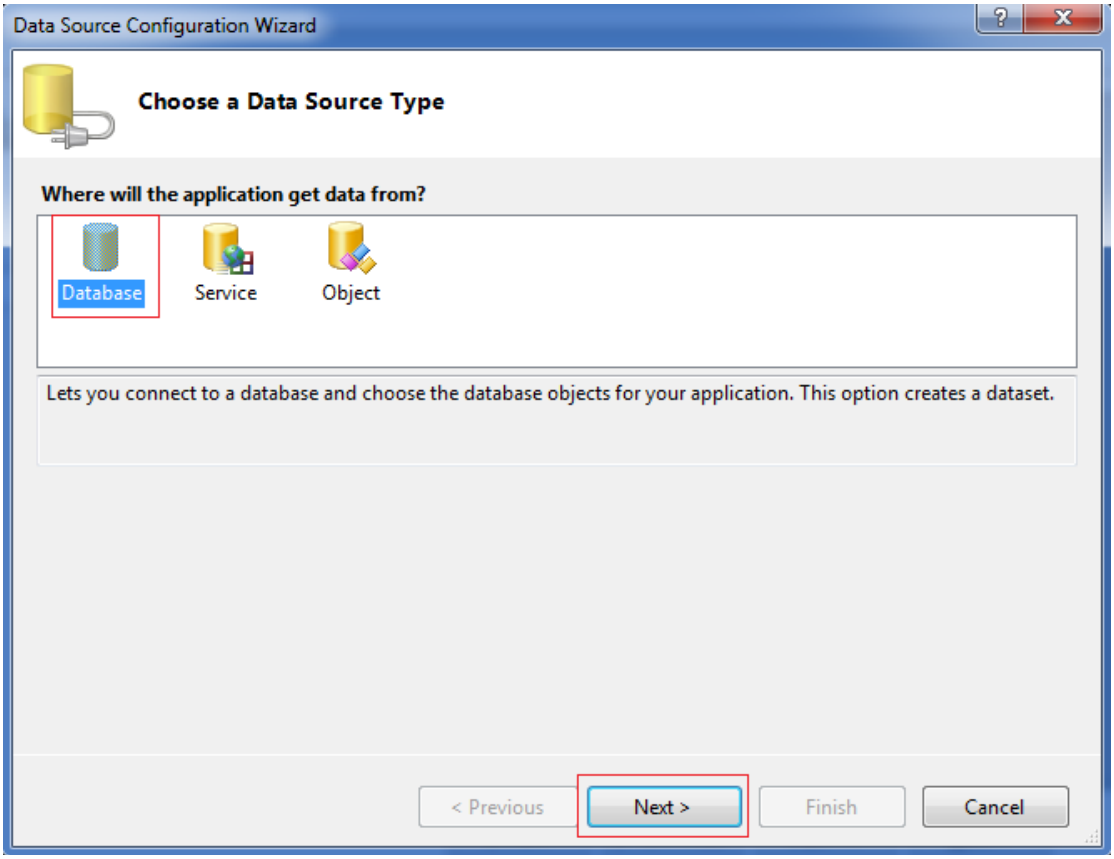


Figure 3-13d

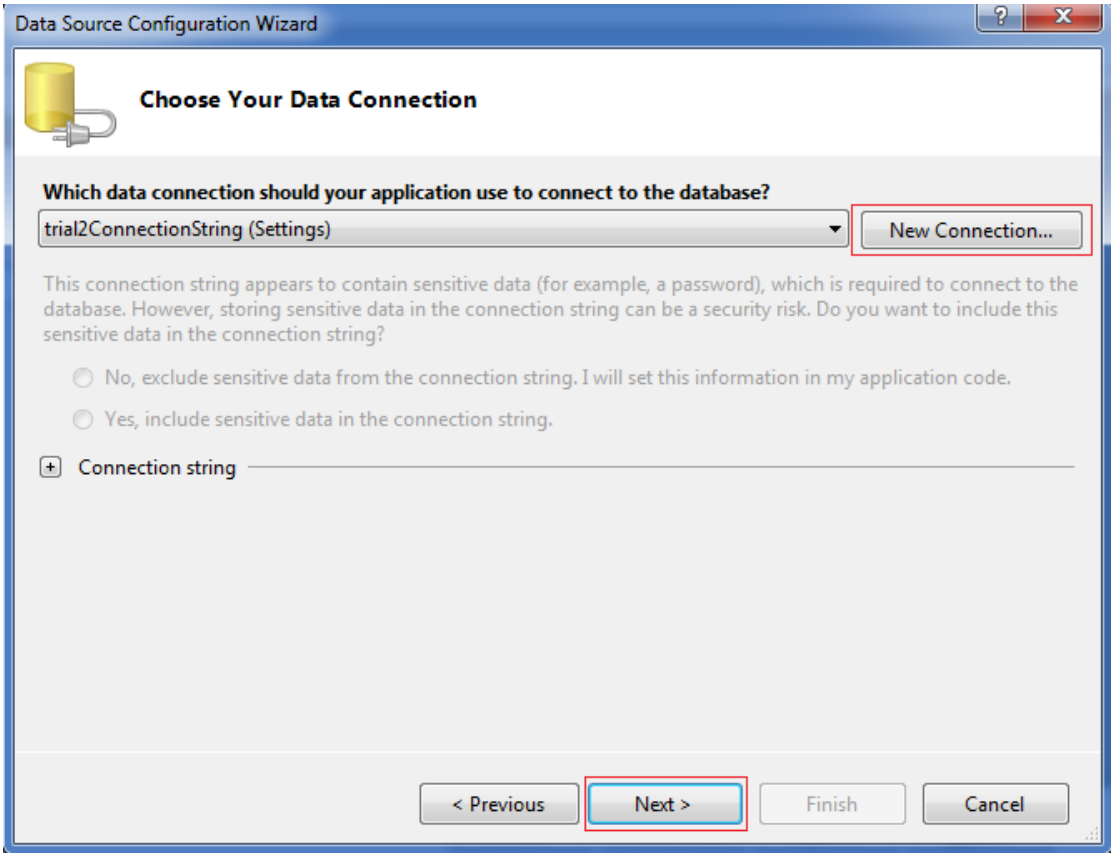


Figure 3-13e

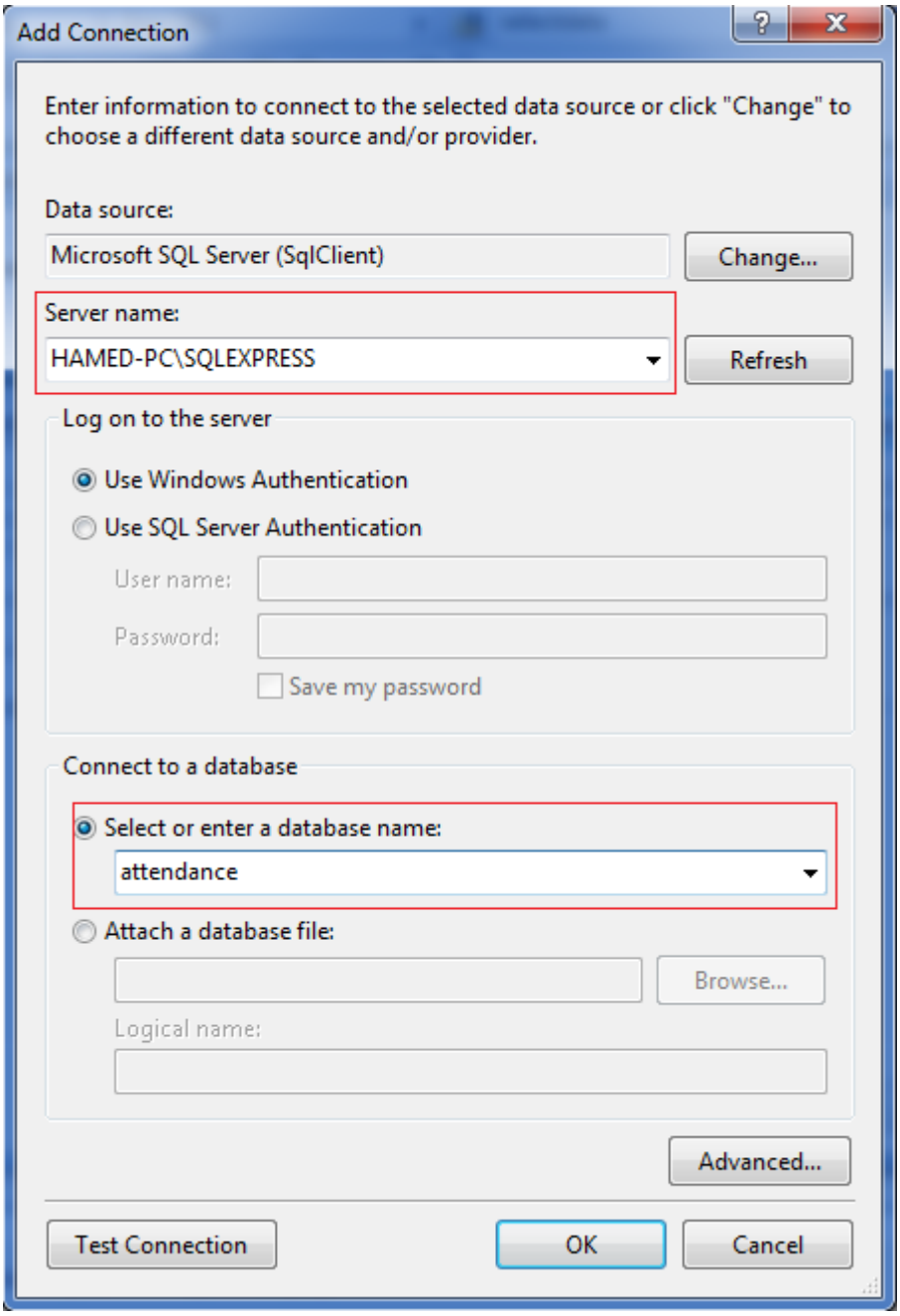

Add Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
HAMED-PC\SQLEXPRESS
Refresh
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
User name:
Password:
Save my password
Connect to a database
Select or enter a database name:
attendance
Attach a database file:
Browse...
Logical name:
Advanced...
Test Connection
OK
Cancel

Figure 3-13f

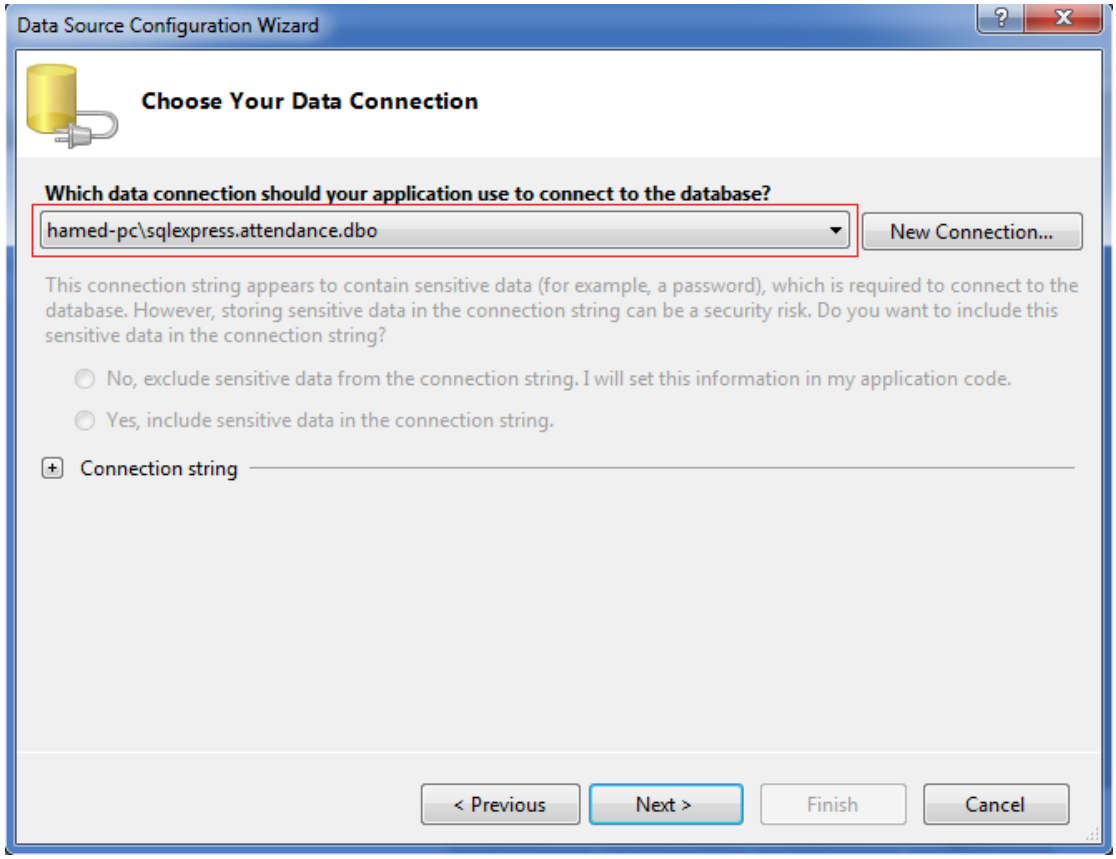


Figure 3-13g

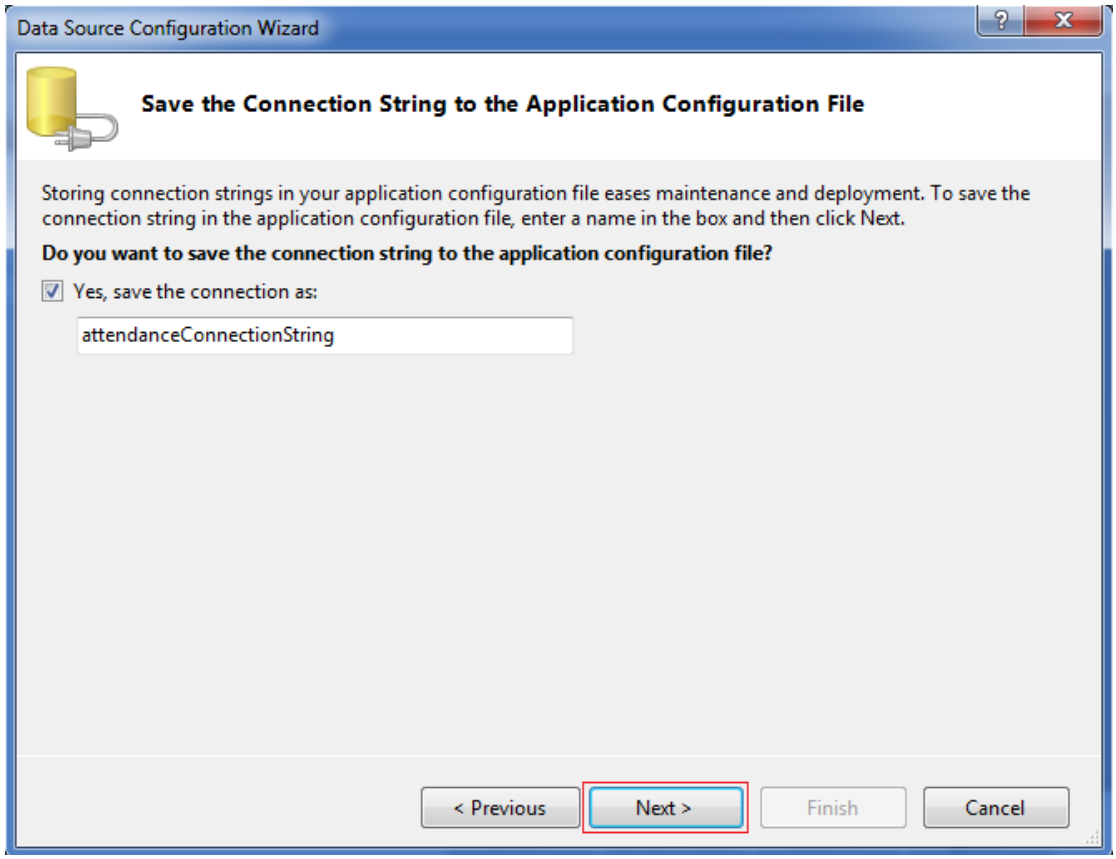


Figure 3-13h

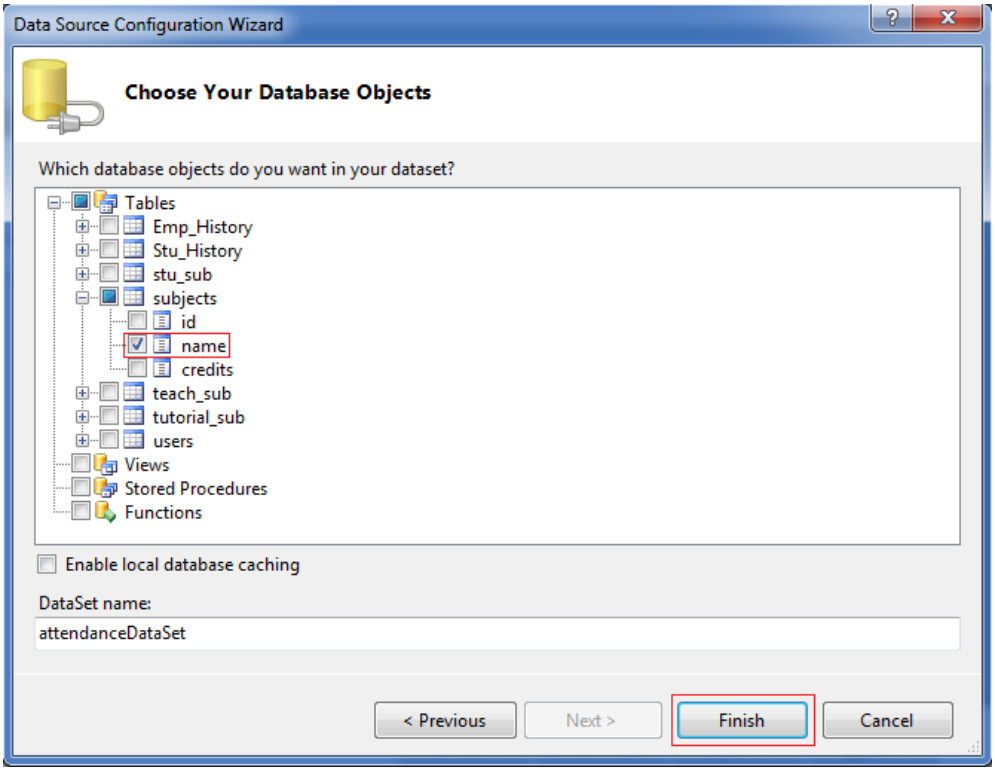


Figure 3-13i

Secondly: button will be used to start the session which means start the lecture in which the doctor will take the attendance of his students. This button will show the grid view containing the students that are registered to the chosen subject.

-Note: clicking on the start session button won't show another form but it will show the same form but it is a bit tricky as clicking on the button will make the rest of the form appear which is pretty good for the doctor or tutor to have a design that is very organized and also have a strong structure. We will explain what will happen after clicking on the button with code to have a clear idea of how we did it.

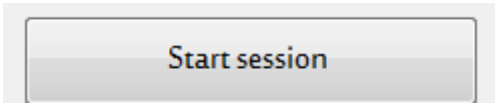


Figure 3-13j

- This button will have the following combination of the code:

```
label1.Visible = false;
            comboBox1.Visible = false;
            start.Visible = false;
            add.Visible = true;
            dataGridView1.Visible = true;
            end.Visible = true;
            delete.Visible = true;
            lblTotal.Visible = true;
```

- This is apart of the code for showing the rest of the form as we notice that we used (Visible function) to control what will appear and not appear on the screen. So for example the datagridview its visibilty function set to be true. An opposite example is the start button visibilty is set to be false after clicking the button.

```
            da.SelectCommand = new SqlCommand("select
users.id,first_name,last_name,att_lec,presence,ref,sub_id from
users,subjects, stu_sub where users.id=stu_sub.stu_id AND
subjects.id=stu_sub.sub_id AND subjects.name='" + comboBox1.Text +
"'", con);
```

- We should consider the start session as multifunction button after it set the visibilty of the buttons and labels etc… now it will use an sql command to show students registered in a specific subject in a datagridview. The sql command tell us that it will select and show the student id, firstname, lastname, number of attended lectures, his presence in the lecture whether it is true or false who are registered in a specific subject that will be determined from the combo box that shows the subject name and is connected to the subject id in the stu_sub table in the database.

```
            //Timer
            startTime = DateTime.Now;
            timDisplay.Enabled = true;
            textBox5.Text = comboBox1.Text;
```

- This portion of code tells that the start time of the lecture will be connected to the computer timer and the timer will be displayed during the session.

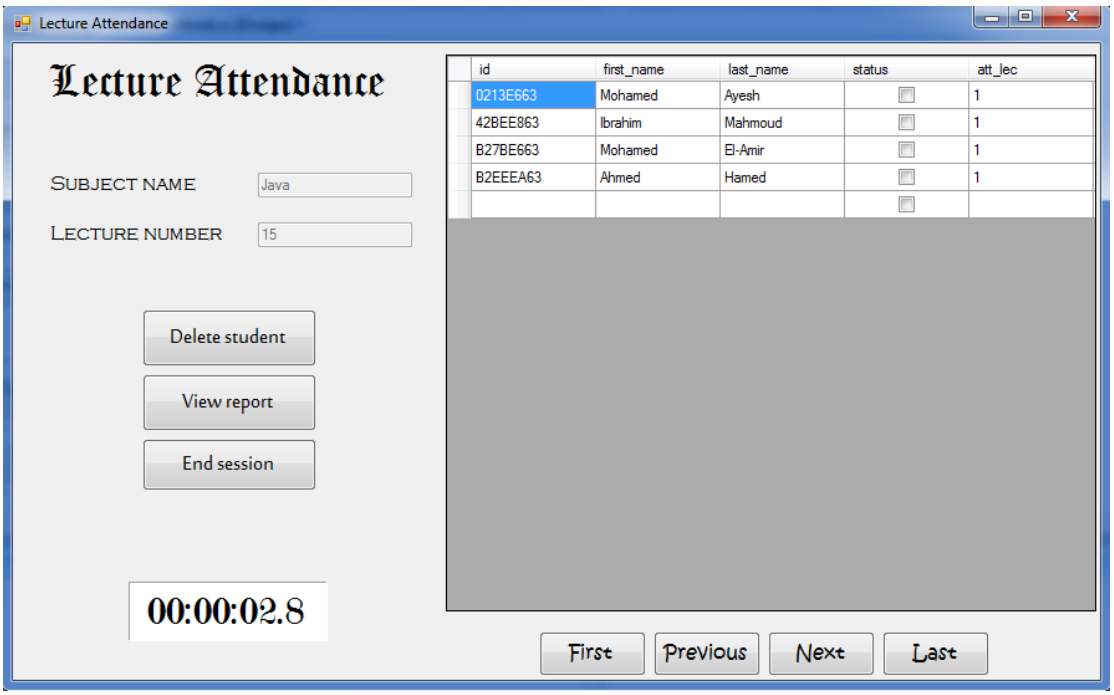


Figure 3-13k

- This figure will appear when we press on the start session button.

- Remember that this is the same form

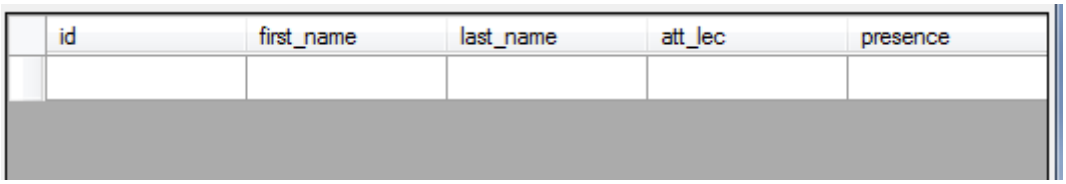


Figure 3-13l

- This is the datagridview that will be displayed after clicking the start session button.

- It is connected directly to the tables in the database system.

- It has the following code:

```
da.SelectCommand = new SqlCommand("select
users.id,first_name,last_name,att_lec,presence,ref,sub_id from
users,subjects, stu_sub where users.id=stu_sub.stu_id AND
subjects.id=stu_sub.sub_id AND subjects.name='" + comboBox1.Text +
"'", con);
```

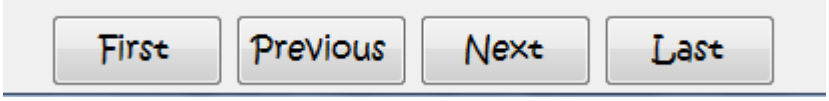


Figure 3-13m

These four buttons is important in navigating through the datagridview. The first button will be used for moving to the first row, the previous button is for moving up in the datagridview, the Next button is for moving down in the datagridview, and finally the Last button is for moving to the last row. For example we have the following code:

```
private void first_Click(object sender, EventArgs e)
        {
            bs.MoveFirst();
            selectdata();
```

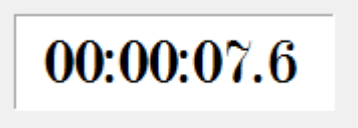


Figure 3-13n

- This code is an open source code.

- It is used to start time counting as soon as the doctor press start session or start the lecture.

- The timer has two functions which are HMS and TimDisplay_Tick.
The timDisplay_Tick is brought from the toolbox that we have in visual basic 2008 and then by clicking on it, it will create a function that we will write the information that we want in it and in our case we will provide it with the current time that is provided by the PC and then it will begin counting.

```
DateTime startTime, stopTime;
        TimeSpan stoppedTime;
private string HMS(TimeSpan tms)
        {

            string s = tms.ToString();
            return (s.Substring(0, s.Length - 6));
        }
private void timDisplay_Tick(object sender, EventArgs e)
        {
            DateTime currentTime;

            currentTime = DateTime.Now;

            lblTotal.Text = HMS(currentTime - startTime -
stoppedTime);
```

- We set the text from its properties to be like what is written in the figure

- Now the operation of the timer is pretty simple as with the opening of the session it will start counting until the doctor press end session button.

- Note: the Counter will be back to zero or in another word will reset as soon as the doctor clicks on end session button.

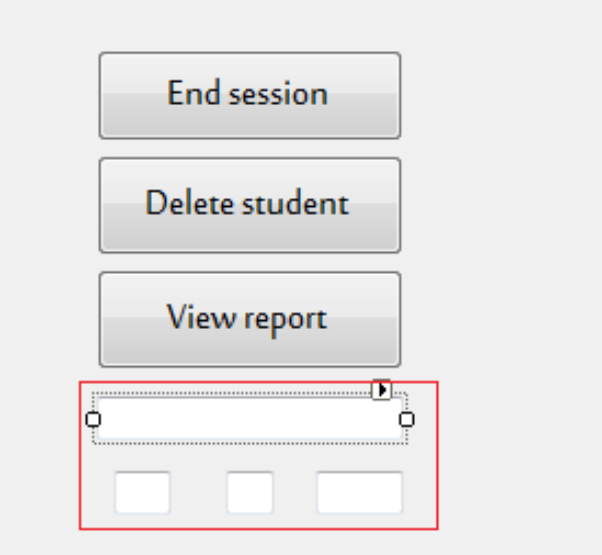


Figure 3-13o

- This a picture of a hidden textboxes. They are hidden as the user won't be in any need for them but they are important for the structure so to keep the design simple and clear of any complications we hid them.

- Note that the students will pass their IDs one by one over the RFID reader, and the reader will read it and make their presence equal to X and that will be discussed in further details.

- Now this figure is vague or enigmatic so we will explain what is going on in this figure.

- The following code will give us a guide through the use of the textboxes and their importance:

- At first we created a textbox that will take id from the reader and then we created an event that when the text changes, it will save it in a string called id:

```
private void id_text_TextChanged(object sender, EventArgs e)
        {

            getsubjectid();
```

- Here we get the subject id first to take the attendance in this specific subject without affecting any other subject and then if the reader reads an id the text will change and after that we execute the following code:

```
SqlCommand com = new SqlCommand("select * from stu_sub where
stu_id='" + textBox2.Text + "'", con);

SqlCommand com2 = new SqlCommand("select * from users where id='" +
textBox2.Text + "'", con2);
```

In which the textbox is connected to the datagridview using binding source, so the code means that it will select id from the stu_sub table and from users table where (con & con2) are the connecetions to the database.

```
w = reader["presence"].ToString();
                FName = reader2["first_name"].ToString();
                LName = reader2["last_name"].ToString();

                if (w == "0")
                {
                    int x;
                    int.TryParse(reader["att_lec"].ToString(), out
x);
                    int y = x + 1;
```

- And now after the reader reads the id of some student, we will check the presence of this person. If it is "0" it means that he/she hasn't been added yet so it will perform a function called TryParse which is a function that will take data in att_lec column in database and put it in an integer called x and then define an integer called y which will take the data of att_lec column in database and increase it by one, in another words, if the student att_lec column equal 5, the integer y will increase it to 6 and so on.

```
SqlCommand com1 = new SqlCommand("update stu_sub set att_lec = '" + y
+ "' where stu_id='"+ textBox2.Text +"' AND sub_id ='"+ subject_id
+"'", con);

SqlCommand com3 = new SqlCommand("update stu_sub set Status = '1'
where stu_id='" + textBox2.Text + "' AND sub_id ='" + subject_id + "'
", con);
```

```
SqlCommand com4 = new SqlCommand("update stu_sub set presence = 'X'
where stu_id='" + textBox2.Text + "' AND sub_id ='" + subject_id + "'
", con);

SqlCommand com5 = new SqlCommand("insert into Stu_History values ('"
+ textBox2.Text + "','" + FName + "','" + LName + "','" +
DateTime.Today.Date + "','" + comboBox1.Text + "','" + comboBox1.Text
+ "')", con2);
```

- This will be the final steps as we begin to update the stu_sub table in the database by setting no of attending lectures to the new number that was updated by y and this is in the condition of that student id is equal to the one in the textbox and that will be written for a specific subject id.

- We also update the stu_sub table by setting the status of the student to be true or false and that is also for a specific student in a specific subject, along with this we update the student's presence to be an X so the doctor won't make the mistake of letting a student attend the lecture twice and will appear a message box telling that the student is already checked.

```
MessageBox.Show("Student is already checked");
```

- The last step will be inserting the student id, firstname, lastname, the date of attended lecture, and finally with the name of the subject the student has checked himself into in the student history table in the database.

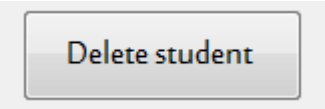


Figure 3-13p

What if an absent student was mistakenly put on the list of students who attended a lecture?

What if a student was annoying the rest of the class and the professor gets angry with him and he fired him from the class but the attendance was taken already?

This can tell us that the professor will need a way to delete this student from this lecture, so we create a button called delete.

- This button will be used to delete the attendance of a specific student in a specific subject.

- We can say that this button is going to work in reverse of what happened previously.

- So now we will discuss in detail what will happen when perform the event of clicking the Delete Student button.

- We first select the whole stu_sub table when the reader reads the id of the student from the textbox.

- The code is as follows:

```
SqlCommand com6 = new SqlCommand("select * from stu_sub where
stu_id='" + textBox2.Text + "'", con);

string z = reader["presence"].ToString();
```

- And now after the reader reads the id of some student, then checked the presence of this person that we want to delete, That is if this person has his presence checked as an X. A message box will first appear to tell the user if he/she is really sure about performing the deletion process and if yes, we will start the TryParse function.

```
  if (z == "X")
            {
if (MessageBox.Show("Are you sure to delete "+textBox1.Text+"
"+textBox4.Text+" ?", "Are you sure ?", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
```

```
1-      int x;
2-      int.TryParse(reader["att_lec"].ToString(), out x);
3-      int y = x - 1;
```

- The previous three steps was to tell us the beginning and the end of TryParse function

- Previously when we were trying to check the student as attended, we took the data in att_lec column in database and put it in an integer called x and then define an integer called y which will take the data of att_lec column in database and increase it by one but in here we will take the data from att_lec column in database and decrease it by one and that will have an immediate effect on that column in database.

```
SqlCommand com7 = new SqlCommand("update stu_sub set att_lec= '" + y
+ "' where stu_id='" + textBox2.Text + "' AND sub_id ='" + subject_id
+ "' ", con);

SqlCommand com8 = new SqlCommand("update stu_sub set presence = '0'
where stu_id='" + textBox2.Text + "' AND sub_id ='" + subject_id + "'
", con);

SqlCommand com9 = new SqlCommand("update stu_sub set Status = '0'
where stu_id='" + textBox2.Text + "' AND sub_id ='" + subject_id + "'
", con);
```

- These are the last steps to complete the deletion process

1- We begin to update the stu_sub table in the database by setting no of attended lectures to the new number that was updated by y and this is in the condition of that student id is equal to the one in the textbox and that will be written for a specific subject id.

2-We also update the stu_sub table by setting the status of the student to be true or false which in this case will be reset to be false and that is also for a specific student in a specific subject, along with this we update the student's presence to be null.

3- The last step will be inserting the student id, firstname, lastname, the date of attended lecture, and finally with the name of the subject the student has checked himself into in the student history table in the database.

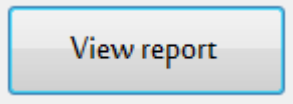


Figure 3-13q

Clicking on this button will open a report that view student attendance and that will be explained in the next figure 3-13r

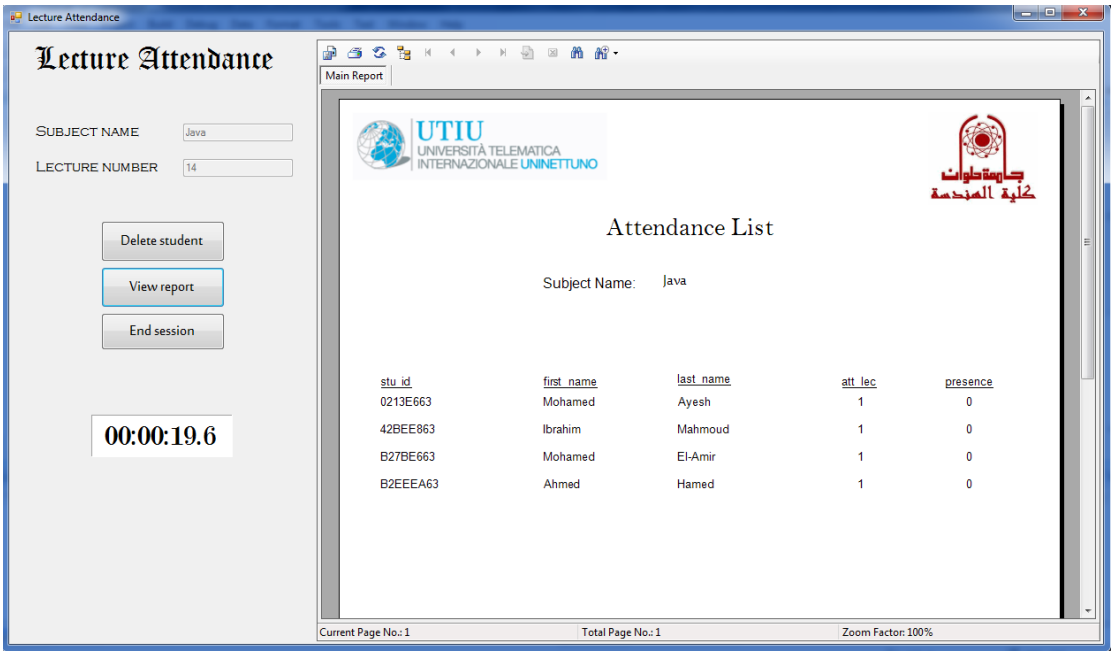


Figure 3-13r

- This report will appear after the doctor presses on the view report button. This is a feature that will allow the doctor to have a hardcopy of the students' attendance in a specific subject. It will contain the required information that the doctor will need to identify every student and his attendance which means that it must be connected to the database and take specific columns needed for this report. The required information are student id, student firstname and lastname and the number of attended lectures and the presence of the student in last lecture. The code of selection will be as follows:

```
da.SelectCommand = new SqlCommand("select
users.id,first_name,last_name,att_lec,presence,ref,sub_id from
users,subjects, stu_sub where users.id=stu_sub.stu_id AND
subjects.id=stu_sub.sub_id AND subjects.name='" + comboBox1.Text +
"'", con);
```

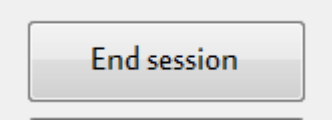


Figure 3-13s

- this button will be used to end the lecture and so it will close the form that is shown in figure 3- and it will show the first form that appeared in the start shown in figure 3- so clicking the button will perform an event that will disable the whole structure that appeared when we click on the start session button and subsequently will return to the beginning of the form which means that we have to perform an event for the button that by clicking on it, it will return to the beginning and we also have to make the button after clicking it to reset all the values in the form and also reset the values in table stu_sub
- the next code is a simple set of the code that will be written:

```
start.Visible = true;

            comboBox1.Visible = true;
            end.Visible = false;
            delete.Visible = false;
            dataGridView1.Visible = false;
            first.Visible = false;
            previous.Visible = false;
            textBox6.Visible = false;
            label2.Visible = false;
            stopTime = DateTime.Now;
            timDisplay.Enabled = false
            SqlCommand reset = new SqlCommand("update stu_sub set
presence = '' ", con);
```

- This sql command will be performed to reset the values of the presence in the stu_sub to zeros and that's will give us the opportunity to take the presence of the student in a certain subject again.
- We should consider the end session button as multifunction button after it set the visibilty of the buttons and labels etc… and it will use an sql command to reset values as we said before.

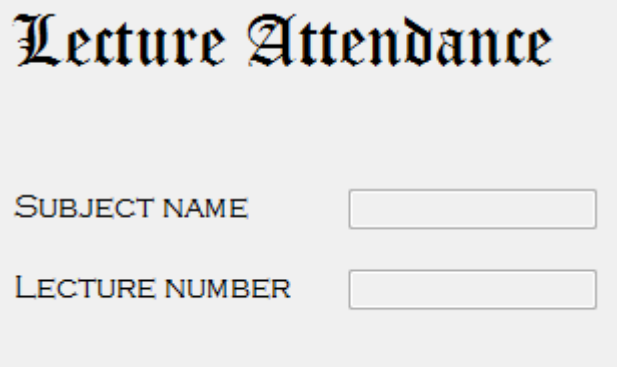


Figure 3-13t

- These two labels and textboxes show the subject name which was selected from a combo box at the start of the lecture and the lecture number determine the number of lectures that the doctor has already given. It could be useful to determine if there is an error in the program as if there is some student that has an attendance higher that the number of lectures we will discover the error and quickly solve it.

3.6 Employee Attendance

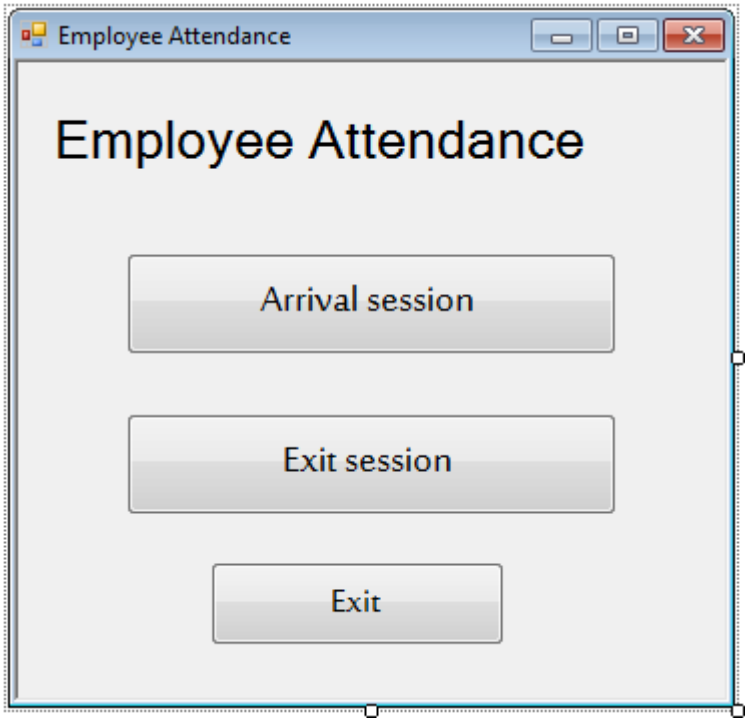


Figure 3-15a

- What we are trying to change by creating an employee attendance system is to replace the old system of taking employees' attendance in some papers and the need for each employee to go to only one place in the college to check on their attendance and dismissal and then sign that paper, but here we will put an RFID reader that will only take the id of the employee and mark him/her as attended in the database and that will save time and effort, also it will take us to a higher level in how to manage an attendance system.

- This form will be used to check the employee attendance; it contains two buttons for attendance and one button to exit this form.

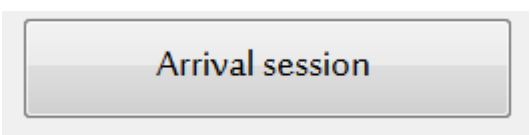


Figure 3-15b

- This button will be used to show the form of taking employee's attendance when the employee arrives to the college.

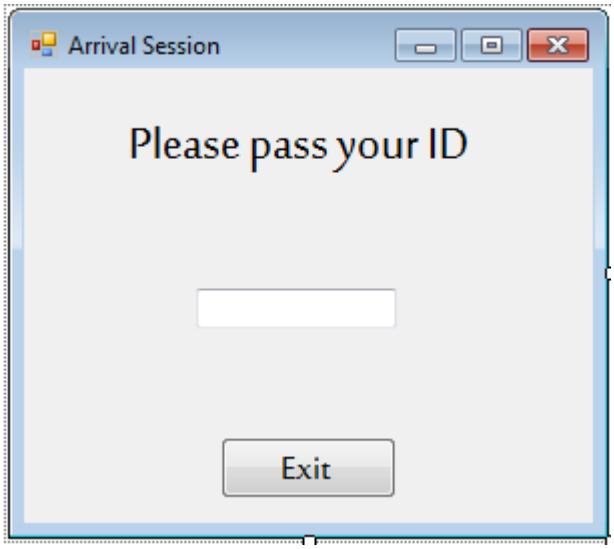


Figure 3-15c

- Here the employee will only pass his id through the RFID reader and he/she will be saved in the database.

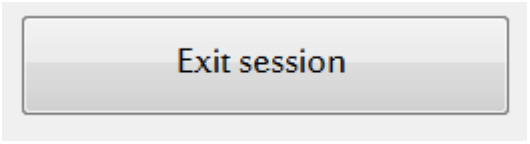


Figure 3-15d

- This button will be used to show the form of taking employee's attendance when the employee exits the college.

Figure 3-15e

- Here the employee will only pass his id through the RFID reader and he/she will be saved in the database.

2011

Chapter Four

RFID Reader

Page no:

101

Helwan University
International Telematic University UNINETTUNO

Chapter four

RFID Reader

4.1 the type of the used RFID reader:

Mifare 13.56Mhz ISO14443A RFID Reader/Writer with USB interface.

4.2 features of the RFID reader:

• Support ISO14443A
• Operation frequency: 13.56 MHz
• Built-in transceiver antenna;
• Reading distance up to ~4-6cm (for ISO card size)
• Less than 100ms decoding time;
• USB 2.0 / USB 1.1 interface
• Power from USB, No need external power adaptor
• Built-in bi-color LED and buzzer

4.3 Protocol Support:

ISO14443A
- Mifare S50
- Mifare S70
- Mifare ultra light

4.4 Software pack:

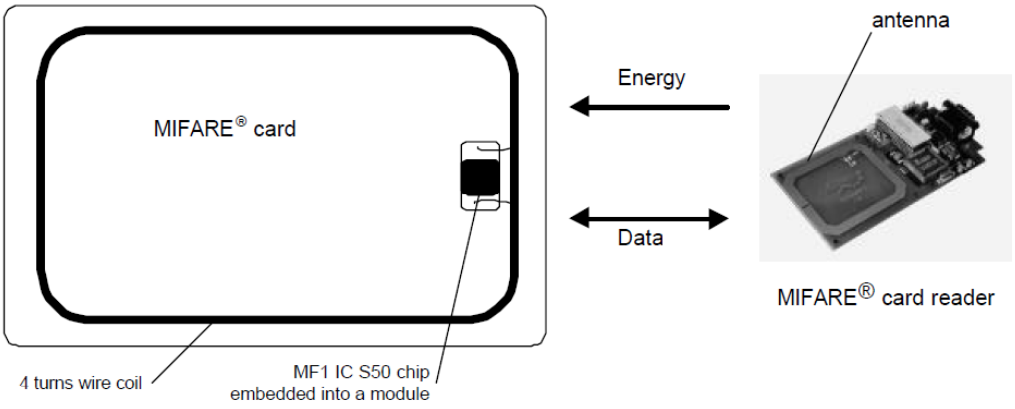
• Demo program - show how to read & write the data into the tag
• Example with source code - includes VB.NET , C# and Delphi
- Speed up software development time
• detail DLL command set and protocol manual
• Data sheet & user manual
• USB to virtual com driver

4.5 the RIFD tags:

5.1 types: Mifare standard IC card
5.2 features: Mifare RF interface (ISO/IEC 14443 A)

- Contactless transmission of data and supply energy (no battery needed).
- Operating frequency: 13.56 MHZ.
- Fast data transfer: 106 Kbit/sec.

- High data integrity: 16 Bit CRC, parity, bit coding, bit counting.
- True anti-collision.



4.6 The used RFID functions:

1. **Automatically connect reader and serial port:** `AutoOpenComPort():`

1.1 Function description:

- This function is used to automatically detect the unoccupied communication port and attached in with the reader. The function tries to establish the connection between them.
- if the connection is established successfully, the function will open the communication port and return a valid handle, otherwise, the function will return an error code with invalid handle (value as -1).

1.2 Usage:

```
public extern static int AutoOpenComPort(ref int Port, ref byte
ComAddr, ref int FrmHandle);
```

1.3 Parameters:

Port: Output. Pointed to the communication port number(COM1~ COM12) that the reader is detected and connected.
ComAddr: Output. Pointed to the address of the reader.
1.4 FrmHandle: Output. Pointed to the handle which is binding with the communication port opened successfully. The application software should use this handle to talk with the reader.
Returns:
Zero value returned when successfully.

2. Disconnect all reader and serial port: `CloseComPort():`

2.1 Function description:

- This fucntion is used to disconnect the reader and release the corresponding communication port reasources. In some development enviroment, the communication port resources must be released before exiting. Otherwise the operating system will become unsuitable.

2.2 Usage:

```
public extern static int CloseComPort();
```

2.3 Parameter:

None

2.4 Return:

- Zero value returned when successful. Please refer to "Function return Value definition" for other value returned.

3. Change reader mode to ISO 14443A: `ChangeTo14443A():`

3.1 Function description:

- This function is used to change the reader mode to ISO 14443A. In this mode, only ISO 1443A tag can be operated.

3.2 Usage:

```
public extern static int ChangeTo14443A(ref byte
ComAddr, int FrmHandle);
```

3.3 Parameter:

ComAddr: Input. Pointed to the address of the reader.
FrmHandle: Input. Handle the communication port the reader is connected to. The handle value is got when calling function Auto Comport.

3.4 Returns:

- Zero value returned when successful.

4. Turn off the RF field: close RF ()

4.1 Function description:

This function is used to turn off the RF field.

4.2 Usage:

Public static extern int closer (ref byte ComAddr, int FrmHandle);

4.3 Parameter:

ComAddr: input. Pointed to the address of the reader.
FrmHandle: input. Handle of the communication port the reader is connected to.
The handle value is got when calling function AutOopenComport or Open Comport.

4.4 Returns:

Zero value returned when successful.

5. Turn on the RFID field: open RF ()

5.1 Function description:

This function is used to turn on the RF field.

5.2 Usage:

Public static extern int OpenRf (ref byte ComAddr, int FrmHandle);

5.3 Parameter: none.

ComAddr: input. Pointed to the address of the reader.
FrmHandle: input. Handle of the communication port the reader is connected to.
The handle value is got when calling function AutOopenComport or Open Comport.

5.4 Returns:

Zero values returned successful.

6. Detected that if any ISO 14443A tag is in RF field: ISO 14443A request ()

6.1 Function description:

This function is used to detect if any ISO 14443A tag is in the RF field and return the tag type if a tag responds.

6.2 Usage:

Public static extern int iso14443ARequest (ref byte ComAddr, byte Mode, byte [] tag type, red byte ErrorCode, int FrmHandle);

6.3 Parameter:

ComAddr: input. Pointed to the address of the reader.
Mode: input. Request mode.
Mode=01: request for all tag.
Mode=00: request for all tag except the tag in the HALT state.
Tag type: output. Pointed to the 2 bytes array describing the tag type with last significant byte first.

Tag type [0] =0x44: Ultra light.
Tag type [0] = 0x04: Mifare one s50.
Tag type [0] = 0x02: Mifare one s70.

Error codes: output. Pointed to an explanation byte when the function return value equals 0x10.
FrmHandle: input. Handle of the corresponding communication port the reader is connected. The handle value is got when calling function AutOopenComport or Open Comport.

6.4 Returns:

Return Value	comment
0	success
0x10	ISO 14443A operation error with error explanation code in error code.

7. ISO14443AAnticoll ()

7.1 Function description:

This function is used to carry out the anti-collision procedure and return the unique serial number (UID) of one tag in the RF field, This function allows only one tag placed in the RF field at the same time.

7.2 Usage:

Public static extern int ISO14443AAnticoll (ref byte ComAddr, byte Reserved, byte [] UID, ref byte ErrorCode, int FrmHandle);

7.3 Parameters:

ComAddr: Input. Pointed to the address of the reader.
Reserved: Input. Reserved for future use with default value 0.
UID: Output. Pointed to 4 bytes tag's UID with least significant byte first.
ErrorCode: Output. Pointed to an explanation byte when the function return value equals to 0x10

Frmhandle: Input. Handle of the corresponding communication port the reader is connected. The handle value is got when calling function AutOopenComport or open comport.

7.4 Returns:

Return Value	Comment
0	Success
0x10	ISO14443A operation error with error explanation code in ErrorCode.

4.7 Problems:

1) We couldn't debug the SDK program included with the reader and it showed this error:

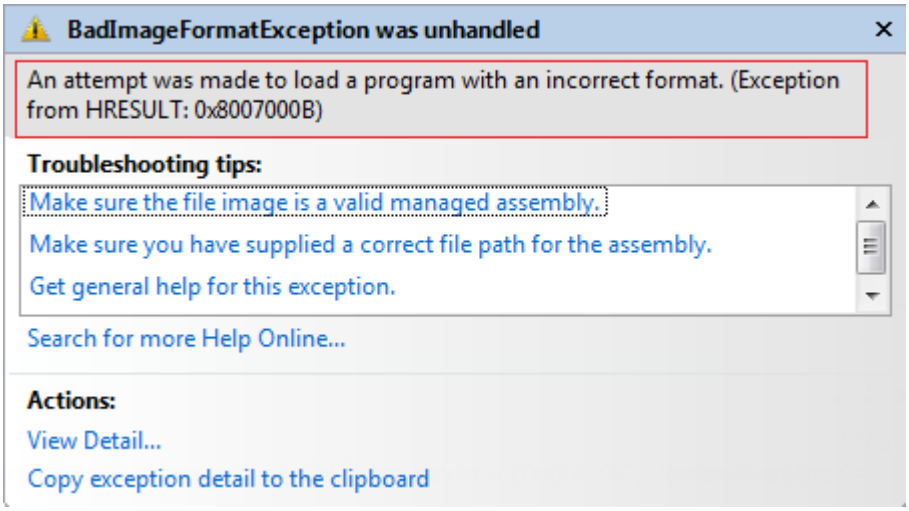


Figure 4-1

We solved this problem by copying error and searched for the solution, Then we found that the DLL file of the RFID Reader is working only on the 32-bit platform, and we were using 64-bit platform so by clicking right click on properties of the solution, then clicking on the build tab and choosing the platform target X86 (32-bit).

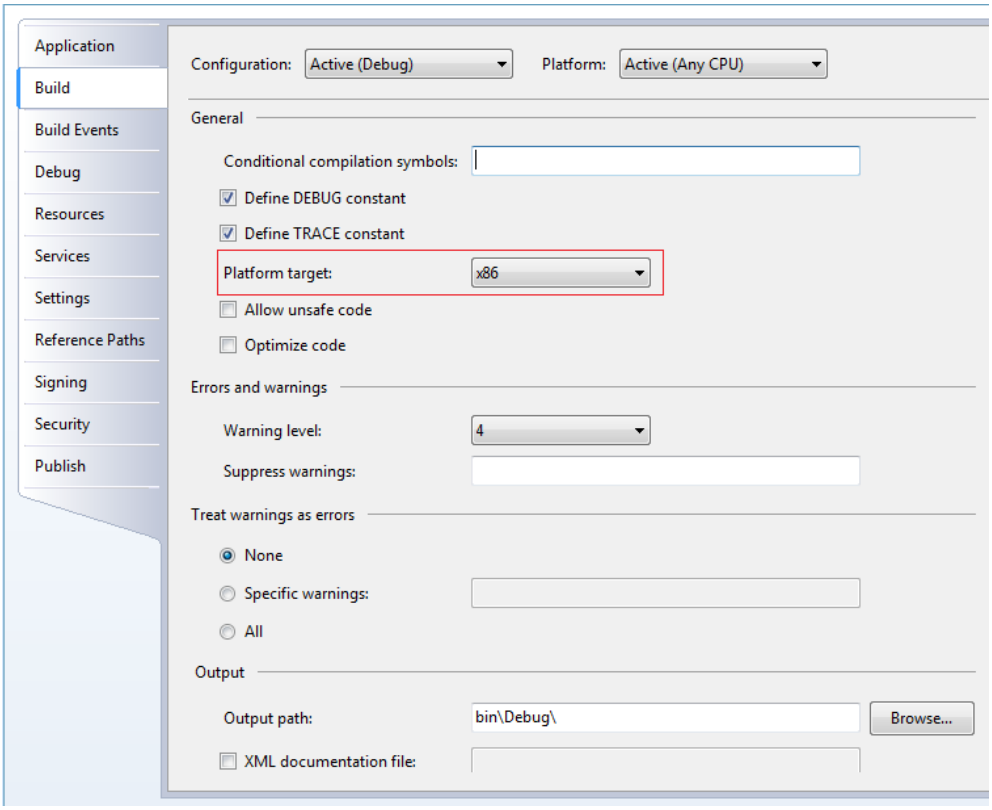


Figure 4-2

2) We needed to continuously read from the reader
So we had to use a continuously working (while) loop which make the program hang, so we a Thread method to solve this problem which make the program work in two phases parallel one is the main program and the other (Thread) in the background.

Then we meet another problem that the Thread is working on a Text-box that is used by the main program so there is an error:

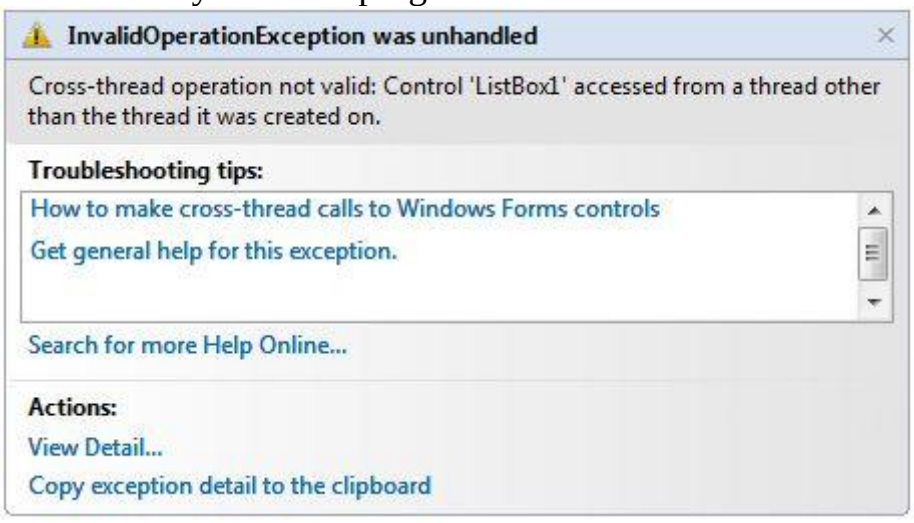


Figure 4-3

And we solved this error with the code which ignores that error.

```
public lecture_attend()
{
    Control.CheckForIllegalCrossThreadCalls = false;
    thread = new Thread(threadfunc);
    InitializeComponent();
}
```

Figure 4-4

Conclusion

This attendance management system is developed using visual studio 2008 C# have met the objectives that have been set for this project. The errors in the system is very much identified and eliminated. This system is operating in a very high level of efficiency and the system will be easy for the students and teachers to learn and understand the system. Finally the system will definitely eliminate the old system of taking attendance by paper if it will be properly operated.