Informatics College Pokhara

informatics

college . pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: Kriti Gurung   **Submitted To:** Ishwor Sapkota
London Met ID: Enter ID Here   Module Leader

Component Grade and Comments	
A. Implementation of Application	
User Interface and proper controls used for designing	missing controls in the interface
Manual data entry or import from csv	not properly saved or imported data
Data Validation	Only basic validation
Enrollment Report & weekly report in tabular format	very poorly executed reports and data not shown accurately
Course wise enrollment report & Chart display	Very poorly designed and only contains one report format with in appropriate data
Algorithm used for sorting & proper sorting of data	Default sorting provided by .net is used
B. Documentation	
User Manual for running the application	User Manual is below average. Is textual only.

Application architecture & description of the classes ad methods sued	Missing Component
Flow chart, algoriathms and data sctructures used	Missing Component
Reflective essay	Very poorly written

C. Programming Style

Clarity of code,Popper Naming convention & comments	Code is poorly written and lacks comments
System Usability	very poorly developed application

Overall Grade:	D	D

Overall Comment:

Code should be self explainable with less comments. Need some proper naming of the component and require to add comments on required area. Import feature is completely missing. Architecture, workflow and flow chart missing.

In overall the code is working and all the functionality seems working.

informatics
college • pokhara

APPLICATION DEVELOPMENT (CS6004NA)

Coursework 1

Submitted By:
Student Name: Milan Thapa
London Met ID: 17031973
Group: L2C1
Date: 10-Jan-2018

Submitted To:
Mr. Ishwor Sapkota
Module Leader

Table of Contents

Table of figure

Introduction

This is our first coursework of Application Development. We're assigned to make WPF desktop application using c #in this coursework. In visual lab, student information system was created using c#. The application allows the user to input personal details of the student including the registration date, so that a system can generate a student's weekly registration report. System must include details such as the employee's name, address, contact no, email, program enrolment, date of enrolment and daily salary amount. This form is intended to keep track of the data, program enrolment and date of the student's enrollment. In addition, there are apps for displaying the table and map daily and weekly. Other available features are well explained in other sections of the report.

1.1 Current Scenario

This is the 21st century, and this century is called the century of digitalization and technology. Files (paper-based) were used before 10/15 years to record different student data in collage and schools. But now various academic institutes are using desktop application where student records are recorded.

1.2 Proposed System

The system proposed is a digitized system specifically designed to overcome problems in the field of education. With the presence of the login section, the system ensures security. Data entry and data display were made easy with an easy user interface.

User Manual

Different screenshots are mentioned below which will illustrate a user to operate the system.
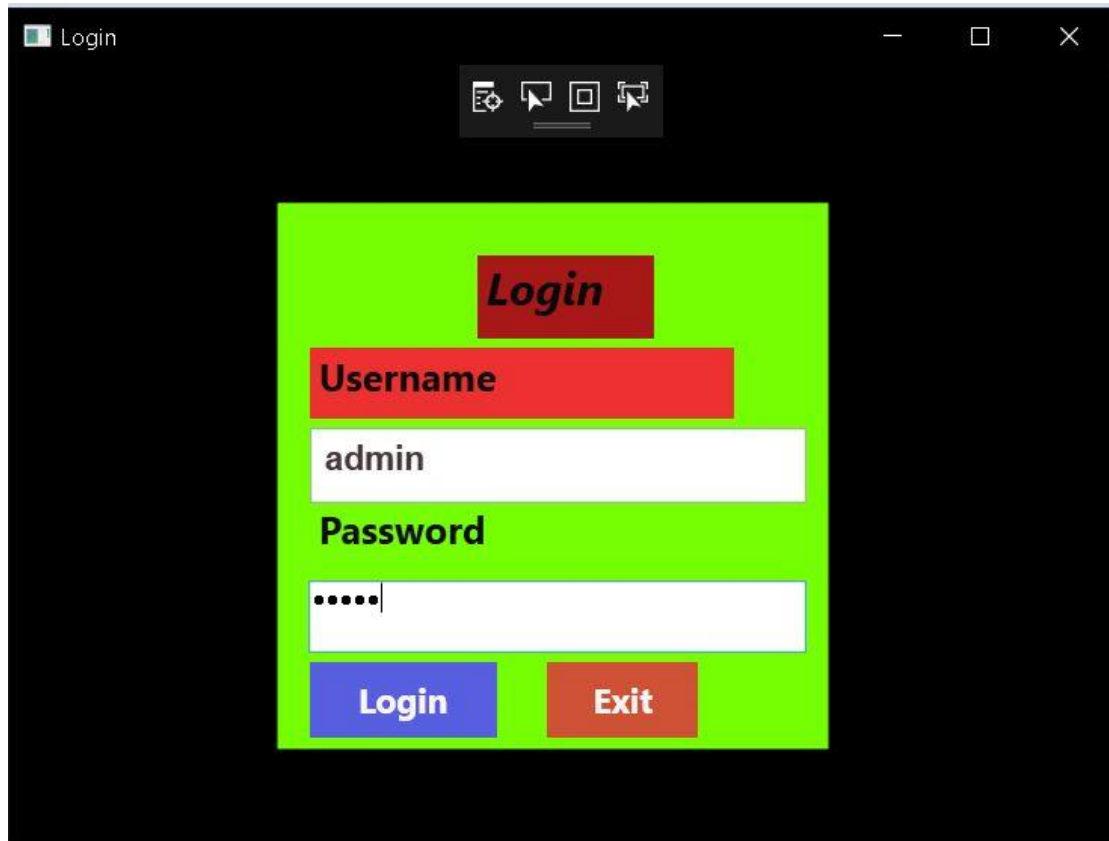

Figure 1 login screen

This is the login section of the system that requires user name and password for the system. In this system, username is "admin" and password is "admin". If username and passwords matches, user will be able to go towards further part of application.

Figure 2 formfor inserting data

In this form students details are filled up.

Figure 3 data saved successfully
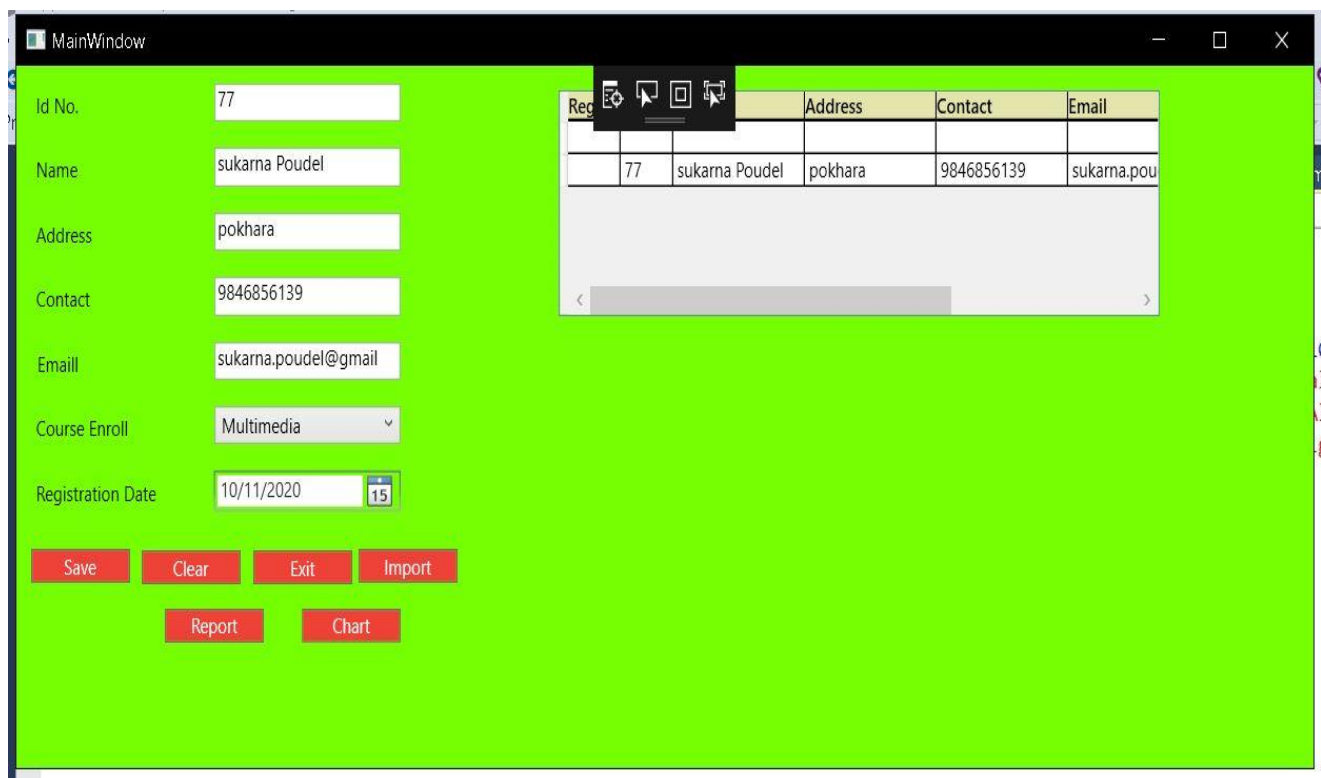
After data is inserted, msg box is generated.

Figure 4 clearing data

After pressing clear button, all the information's are cleared.

Figure 5 importing data from xml report

After we import data, all data's are imported from xml file towards table grid.
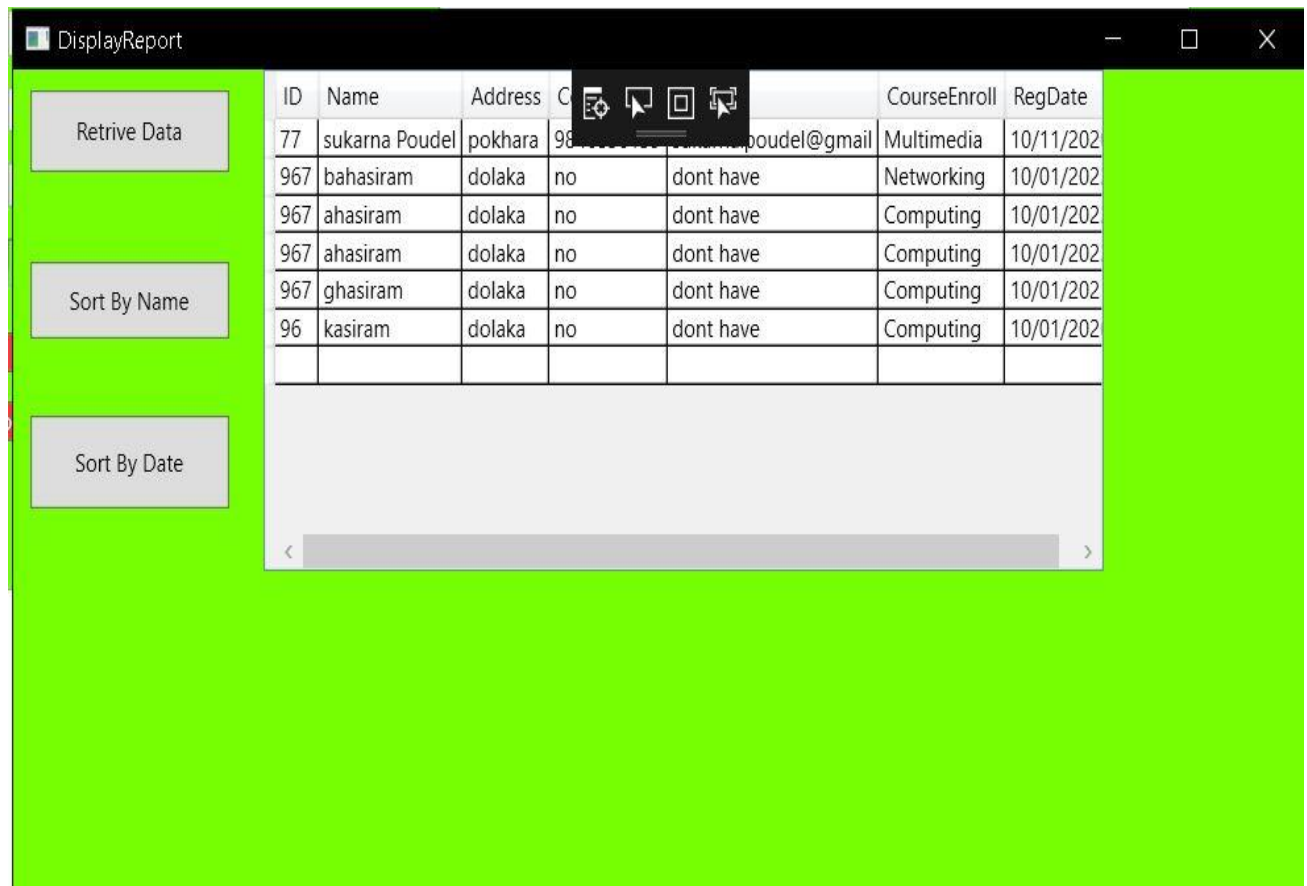
Figure 6 Displaying Report

In this part, all the information's that are filled up are displayed in table grid.
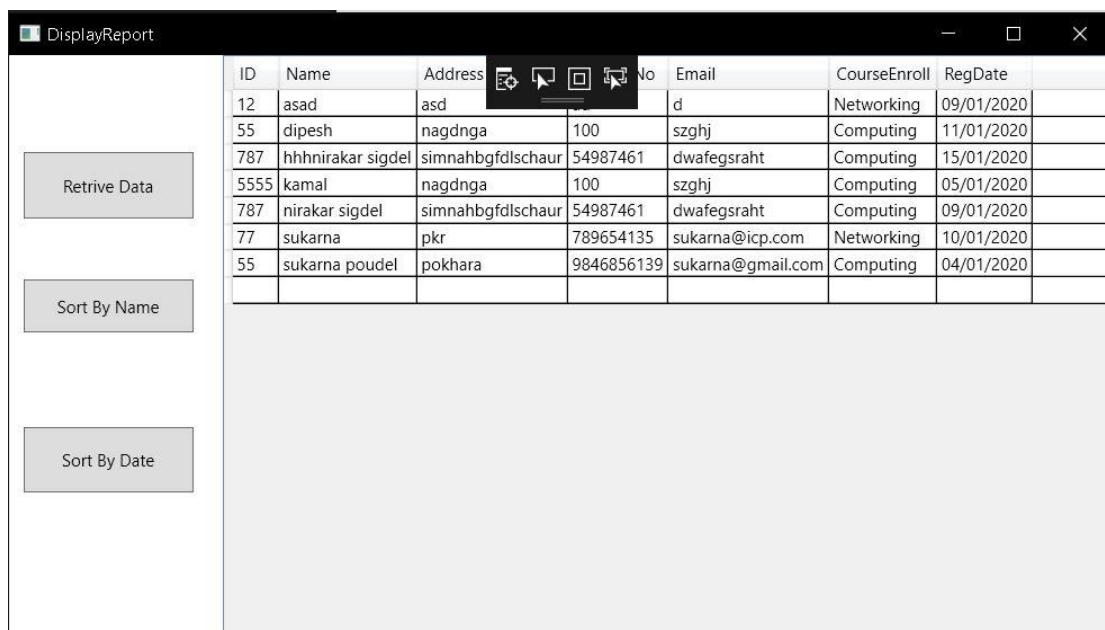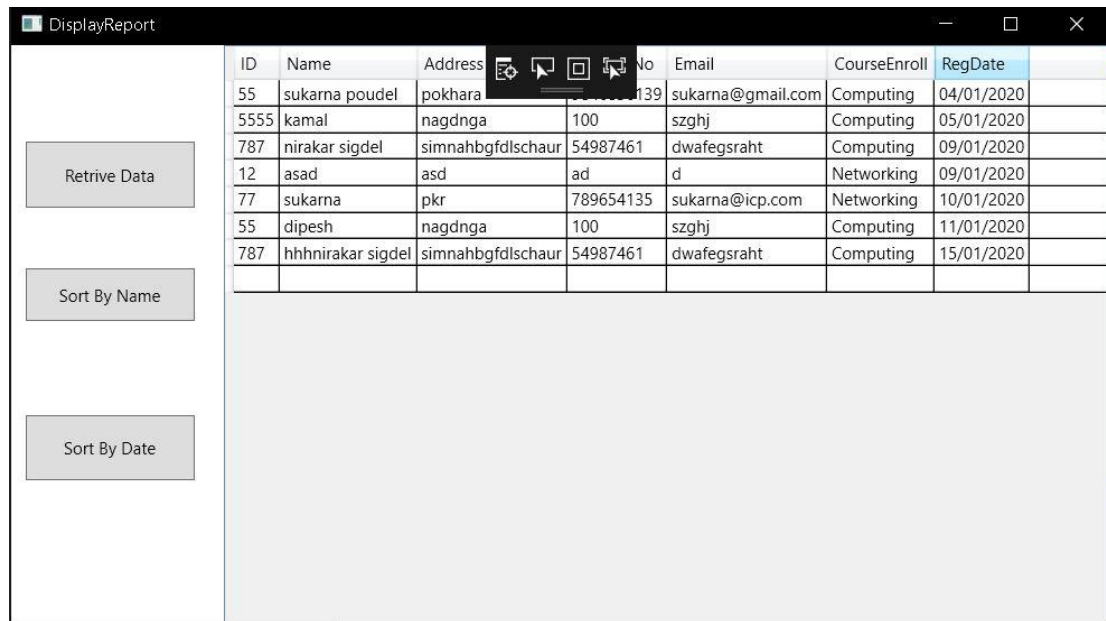
Figure 7 sorting by name

In this part, data's are shown by sorting names.

Figure 8 sorting by date

In this part, data's are shown by sorting date. Student show are enrolled before are shown above.
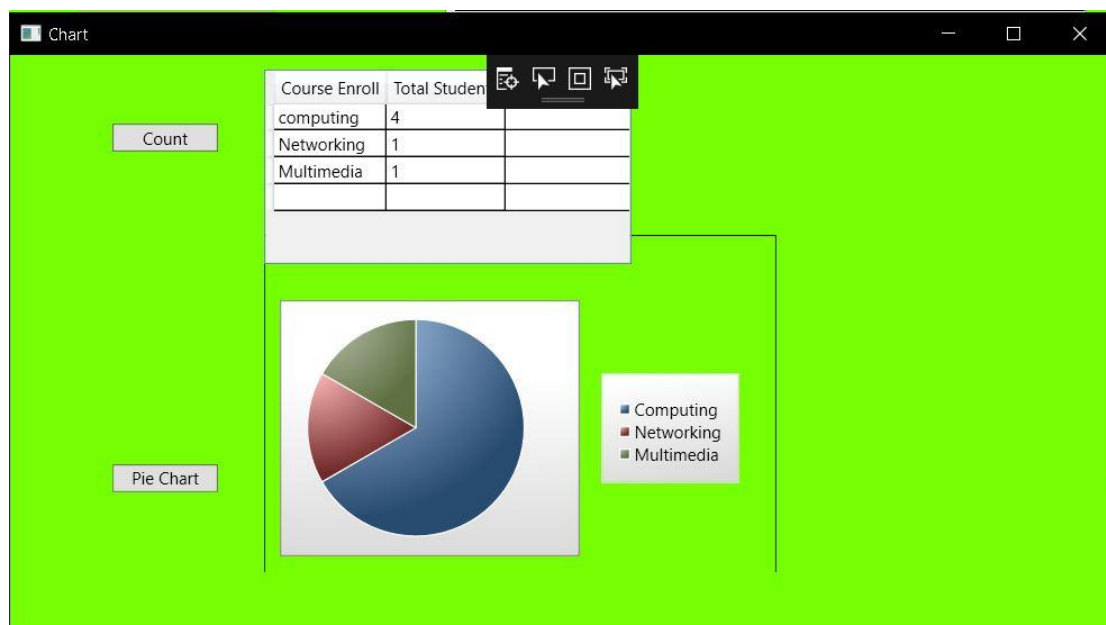
Figure 9 pie chart

This is the chart section in which students that are enrolled in different faculties are shown in pie chart. And number of students enrolled in different faculties are also shown here.

6

2. Journals Articles

C# Language Specification

C# is a simple, modern, object-oriented, and type-safe programming language that combines the high productivity of rapid application development languages with the raw power of C and C++. Written by the language's architect and design team members, The C# Programming Language is the definitive technical reference for C#. Moving beyond the online documentation, the book provides the complete specification of the language along with descriptions, reference materials, and code samples from the C# design team. (Anders Hejlsberg, 2003)

Microsoft Visual C# (Core Reference)

The definitive guide to using Visual C# .NET to develop stand-alone applications for Microsoft Windows and Web-enabled Microsoft .NET applications get the complete guidance you need to use the Visual C# .NET language to produce stand-alone Windows-based applications and Web-enabled .NET applications with this comprehensive reference. It thoroughly covers the language's structure, syntax, code wizards, and the Microsoft Visual Studio® design environment, paying close attention to both the client and server sides of the .NET environment. (Williams, 2002)

Visual C# 2010 How to Program

Appropriate for all basic-to-intermediate level courses in Visual C# 2010 programming. Created by world-renowned programming instructors Paul and Harvey Detail, Visual C# 2010 How to Program, Fourth Edition introduces all facets of the C# 2010 language hands-on, through hundreds of working programs. This book has been thoroughly updated to reflect the major innovations Microsoft has incorporated in Visual C# 2010 and .NET. all discussions and sample code have been carefully audited against the newest Visual C# language specification. Students begin by getting comfortable with the C# Express 2010 IDE and basic Visual C# syntax. Next, they build their skills one-step at a time, mastering control structures, classes, objects, methods, variables, arrays, and the core techniques of object-oriented programming. (Harvey Deitel, 2010)

7

Beginning Visual C# 2010

C# is a relatively new language that was unveiled to the world when Microsoft announced the first version of its .NET Framework in July 2000. Since then its popularity has rocketed, and it has arguably become the language of choice for both Windows and Web developers who use the .NET Framework. Part of the appeal of C# comes from its clear syntax, which derives from C/C++ but simplifies some things that have previously discouraged some programmers. Despite this simplification, C# has retained the power of C++, and there is now no reason not to move into C#. The language is not difficult and it's a great one to learn elementary programming techniques with. This ease of learning, combined with the capabilities of the .NET Framework, make C# an excellent way to start your programming career. (KARLI WATSON, 2010)

Similar books
- The C# yellow book (Miles, 2015)
- Fundamental of computer programming with C# (Svetlin Nakov, n.d.)

Algorithms of
Weekly Report

Steps:

1. Start

2. Check student's details

3. If it doesn't exist, display error message and restart

4. If exists, read file

5. find data count

6. Display report

7. Retrieve the data

8. Display Report

9. Display data in chart

10. Stop

Sorting Algorithm

Bubble sort is a calculation which is used to sort a given arrangement of component furnished in type of an exhibit with n number of component. Bubble sort valuate all component one by one and sort them depending on their qualities. In the event, that the given exhibit must be sorted in rising request, bubble sort will begin by looking at the main component of the cluster with the second component. If the principal component is more noteworthy than the second component, it will swap both the components, and after that proceed onward to analyse the second and the third component, etc.

In the event that we have complete n components, we have to rehash this procedure for multiple times. It is known as bubble sort, in light of the fact that with each total emphasis the biggest component in the given cluster, bubbles up towards the last place or the most noteworthy file, much the same as a water bubble ascends to the water surface. Sorting happens by venturing through every one of the components one-by-one and contrasting it and the contiguous component and swapping them whenever required. (Anon., n.d.)

How Bubble Sort Works?

We take an unsorted array for our example. Bubble sort takes $O(n^2)$ time so we're keeping it short and precise.

Bubble sort starts with very first two elements, comparing them to check which one is greater.

In this case, value 33 is greater than 14, so it is already in sorted locations. Next, we compare 33 with 27.

We find that 27 is smaller than 33 and these two values must be swapped.
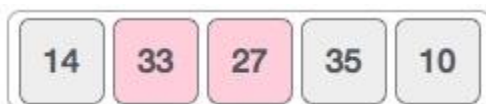
The new array should look like this −

Next we compare 33 and 35. We find that both are in already sorted positions.

| 14 | 27 | 33 | 35 | 10 |

Then we move to the next two values, 35 and 10.

| 14 | 27 | 33 | 35 | 10 |

We know then that 10 is smaller 35. Hence they are not sorted.

| 14 | 27 | 33 | 35 | 10 |

We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this −

| 14 | 27 | 33 | 10 | 35 |

To be precise, we are now showing how an array should look like after each iteration. After the second iteration, it should look like this −

| 14 | 27 | 10 | 33 | 35 |

Notice that after each iteration, at least one value moves at the end.

| 14 | 10 | 27 | 33 | 35 |
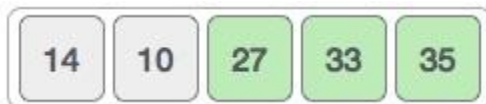
And when there's no swap required, bubble sorts learns that an array is completely sorted.

| 10 | 14 | 27 | 33 | 35 |
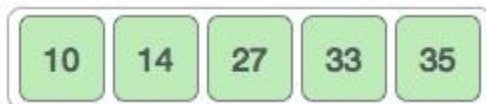
(tutorialsport.com, n.d.)

Reflection

The created framework is Digitalized student information System. It is created utilizing Visual Studio 2017 with the C# language variant 7.3. The GUI planned is profoundly UI and client with fundamental framework organization can work the framework The application can perform the entire task that was proposed. C# was used as a programming language to develop the application. While doing this project, I did a lot of research and got chance to learn many things. I get to know how windows application work and how it is made. I referred many websites, watched YouTube videos to do this project. I faced many problem and errors while codding and somehow I manage to fix the error. Sometime I used to take help of module teacher and sometime my friends used to help me. Every time I make mistake, I used to get chance to learn new thing.

After the completion of project, I found myself to a next level. I was enjoying while doing this projects. It was very new to me. The project has shown me a way to increase my creativity and thinking problem.This project was very beneficial to me and it increased my programming skills..

Conclusion

In this way, I have developed student information system Every function required for the application was implemented and they all work perfectly. It was difficult because it was our first time working in Visual Studio with C# language.

While doing this project I did a lot of research, which was very useful. I came to lean many new thing. This was completely a new project to us. However, without losing hopes I completed it in purposed time.

I faced many difficulties while doing this project in coding phrase. Designing phrase was easy. We just need to drag and drop the item. While doing coding there were many bugs and error. Some time I used to loss my patient, I used to be frustrate if I cannot find error in the code.

I overcome every error one by one. I used to do online research to find solution. I used to ask module teacher about my error. Sometimes friends also help me doing the project.

Bibliography

ejmste.com. (n.d.). Retrieved from http://www.ejmste.com/Important-Factors-Affecting-
        Student-Information-System-Quality-nand-Satisfaction,81147,0,2.html

manihutu, m. (2018).

McLean, D. a. (n.d.).

mukharjee, s. (2015).

tutorialsport.com. (n.d.). Retrieved from
        https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorith
        m.htm

william, w. (2017).

Appendix

Main window.xaml.cs

```csharp
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace ApplicationDevelopmentCW
{

    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            Student student = new Student();

            DataGridXAML.Items.Add(student);
        }
        public class Student
```

```csharp
        {
            public string ID
            {
                get; set;
            }

            public string Name { get; set; }

            public string Address { get; set; }

            public string Contact { get; set; }

            public string CourseEnroll { get; set; }

            public string RegDate { get; set; }

            public string Email { get; internal set; }
        }
        private void btnSave_Click(object sender, RoutedEventArgs e)
        {

            //empty input validation
            if (textId.Text == "")
            {
                MessageBox.Show("Empty Id!");

            }
            else if (textName.Text == "")
            {
                MessageBox.Show("Name is required");
            }


            else if (textAddress.Text == "")
            {
                MessageBox.Show("Empty Address!");

            }
            else if (textContact.Text == "")
            {
                MessageBox.Show("Empty Contact Number!");

            }
            else if (textEmail.Text == "")
            {
                MessageBox.Show("Empty email id!");

            }

            else if (textCourse.Text == "")
            {
                MessageBox.Show("Invalid Level!");

            }



            else
            {
                var handler = new DataHandler();
                var dataSet = handler.CreateDataSet();
                AddSampleData(dataSet);
```

```csharp
                MessageBox.Show("Data saved successfully !!!");
                if (File.Exists(@"G:\student.xml"))
                {
                    dataSet.ReadXml(@"G:\student.xml");
                    dataSet.WriteXml(@"G:\student.xml");
                }
                else
                {
                    dataSet.WriteXml(@"G:\student.xml");
                }
            }



        }
        private void AddSampleData(DataSet dataSet)
        {
            var dr1 = dataSet.Tables["Student"].NewRow();
            dr1["ID"] = textId.Text;
            dr1["Name"] = textName.Text;
            dr1["Address"] = textAddress.Text;
            dr1["Contact"] = textContact.Text;
            dr1["Email"] = textEmail.Text;
            dr1["CourseEnroll"] = textCourse.Text;
            string text = textDate.Text;
            dr1["RegDate"] = text;
            dataSet.Tables["Student"].Rows.Add(dr1);

        }

        private void btnImport_Click(object sender, RoutedEventArgs e)
        {
            if (textId.Text == "")
            {
                MessageBox.Show("Empty ID!");

            }
            else if (textName.Text == "")
            {
                MessageBox.Show("Name is required");
            }


            else if (textAddress.Text == "")
            {
                MessageBox.Show("Empty Address!");

            }
            else if (textContact.Text == "")
            {
                MessageBox.Show("Empty Contact Number!");

            }
            else if (textEmail.Text == "")
            {
                MessageBox.Show("Empty email id!");

            }

            else if (textCourse.Text == "")
            {
                MessageBox.Show("Invalid Course!");
```

```
            }

            else
            {
                Student dataStudent = new Student();
                dataStudent.ID = textId.Text;
                dataStudent.Name = textName.Text;
                dataStudent.Address = textAddress.Text;
                dataStudent.Contact = textContact.Text;
                dataStudent.Email = textEmail.Text;
                dataStudent.CourseEnroll = textCourse.Text;
                dataStudent.RegDate = textDate.Text;

                DataGridXAML.Items.Add(dataStudent);
            }
        }

        private void btnClear_Click(object sender, RoutedEventArgs e)
        {
            textId.Clear();
            textName.Clear();
            textAddress.Clear();
            textContact.Clear();
            textCourse.SelectedIndex = -1;
            textEmail.Clear();

        }

        private void btnExit_Click(object sender, RoutedEventArgs e)
        {
            MessageBox.Show("Window is being exited.");
            this.Close();
        }

        private void btnReport_Click(object sender, RoutedEventArgs e)
        {
            DisplayReport displayReport = new DisplayReport();
            displayReport.Show();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            Chart chart = new Chart();
            chart.Show();

        }

    }


}



Login.xaml.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```csharp
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Window
    {
        public Login()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            string username = textBox1.Text;
            string password = textBox2.Password;

            if (username == "")
            {
                MessageBox.Show("Username cannot be empty!");
            }
            else if (password == "")
            {
                MessageBox.Show("Password cannot be empty!");
            }
            else if (password == "admin" && username == "admin")
            {
                this.Hide();
                MainWindow mainWindow = new MainWindow();
                mainWindow.Show();
            }

            else
            {
                MessageBox.Show("Incorrect username or password!");
            }

        }

        private void Exit_Click(object sender, RoutedEventArgs e)
        {
            MessageBox.Show("Window is being exited.");
            this.Close();
        }
    }
}
```

Display Report.xaml.cs

```csharp
using System;
using System.Collections.Generic;
```

```csharp
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for DisplayReport.xaml
    /// </summary>
    public partial class DisplayReport : Window
    {
        DataTable buffer;

        public DisplayReport()
        {
            InitializeComponent();
        }

        private void show_data()
        {
            String dataXML = @"G:\student.xml";
            DataSet dataset = new DataSet();
            dataset.ReadXml(dataXML);

            buffer = new DataTable("dt");
            buffer.Columns.Add("ID", typeof(String));
            buffer.Columns.Add("Name", typeof(String));
            buffer.Columns.Add("Address", typeof(String));
            buffer.Columns.Add("ContactNo", typeof(String));
            buffer.Columns.Add("Email", typeof(String));
            buffer.Columns.Add("CourseEnroll", typeof(String));
            buffer.Columns.Add("RegDate", typeof(String));

            for (int i = 0; i < dataset.Tables[0].Rows.Count; i++)
            {
                string s = dataset.Tables[0].Rows[i][6].ToString();
                DateTime dtime = DateTime.Parse(s);
                buffer.Rows.Add(
                    dataset.Tables[0].Rows[i][0].ToString(),
                    dataset.Tables[0].Rows[i][1].ToString(),
                    dataset.Tables[0].Rows[i][2].ToString(),
                    dataset.Tables[0].Rows[i][3].ToString(),
                    dataset.Tables[0].Rows[i][4].ToString(),
                    dataset.Tables[0].Rows[i][5].ToString(),
                    dtime.ToShortDateString());
            }
            DataView dataV = new DataView(buffer);
            DataGridReport.ItemsSource = dataV;
        }

        private void buttonRetrive_Click(object sender, RoutedEventArgs e)
        {
            show_data();
```

```csharp
        }

        private void buttonSName_Click(object sender, RoutedEventArgs e)
        {
            DataView dataV = new DataView(buffer);
            dataV.Sort = "Name ASC";
            DataGridReport.ItemsSource = dataV;
        }

        private void buttonSD_Click(object sender, RoutedEventArgs e)
        {
            DataView dataV = new DataView(buffer);
            dataV.Sort = "RegDate ASC";
            DataGridReport.ItemsSource = dataV;
        }


        private void buttonChart_Click(object sender, RoutedEventArgs e)
        {
            var dataSet = new DataSet();
            dataSet.ReadXml(@"D:\student.xml");
            DataTable dtStudentReport = dataSet.Tables[0];

            int total_Computing = 0;
            int total_Networking = 0;
            int total_Multimedia = 0;

            DataTable dt = new DataTable("newTable");
            dt.Columns.Add("Course Enroll", typeof(string));
            dt.Columns.Add("Total Students", typeof(int));

            for (int i = 0; i < dtStudentReport.Rows.Count; i++)
            {
                string col =
dtStudentReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    total_Computing++;
                }
                else if (col == "Networking")
                {
                    total_Networking++;
                }
                else if (col == "Multimedia")
                {
                    total_Multimedia++;
                }

            }
            dt.Rows.Add("computing", total_Computing);
            dt.Rows.Add("Networking", total_Networking);
            dt.Rows.Add("Multimedia", total_Multimedia);
            DataGridReport.DataContext = dt.DefaultView;


        }
    }
}
```