informatics
college • pokhara

Application Development

CS6004NA

Coursework 1

Submitted By:		**Submitted To:**
Student Name:	Sukarna Poudel	Mr. Ishwor Sapkota
London Met ID:	17030530	
Group:	L2C2	
Date:	10-Jan-2018	

Table of Contents

Introduction 1

1.1 Current Scenario 1

1.2 Proposed System 1

User Manual 1

Algorithms of 8

Sorting Algorithm 9

How Bubble Sort Works? 10

Reflection 11

Conclusion 11

Bibliography 12

Appendix 12

Table of figure

Figure 1 login screen 2
Figure 2 formfor inserting data 2
Figure 3 data saved successfully 3
Figure 4 clearing data 3
Figure 5 importing data from xml report 4
Figure 6 Displaying Report 5
Figure 7 sorting by name 5
Figure 8 sorting by date 6
Figure 9 pie chart 6

Introduction

In this coursework we are assigned to make WPF desktop application using c#. Student Information System was made using c# in visual studio. The application allows the user to input the student personal detail including registration date so that a system can generate a weekly enrolment report of the student. System must include detail like Name, address, contact no, email, program enrol, registration date and daily wage amount of the employee. This application is to keep track of the student's details, program enrol and registration date. Furthermore, there is a features to view daily and weekly table and chart. Other available features are well explained in other sections of the report.

1.1 Current Scenario

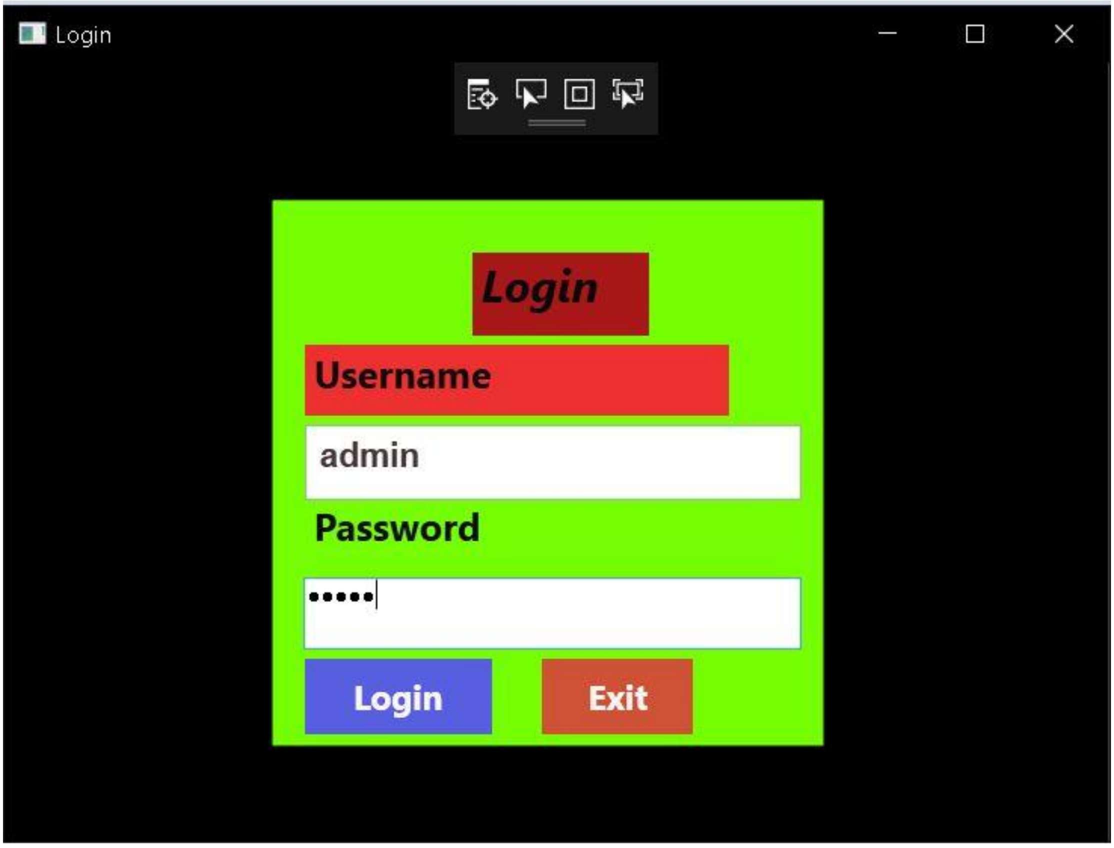
This is the 21st century and this century is called century of technology and digitalization. Before 10/15 years, files (paper based) were used to record different data's of student in collage and schools. But now different academic institutes uses desktop application where records of student are recorded.

1.2 Proposed System

The proposed system is digitized system which is specially designed to overcome problem in educational field. The system ensures security with the presence of login section. Entry of data and display of data have been made easy with the presence of easy user-interface.

User Manual

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



Login

Username

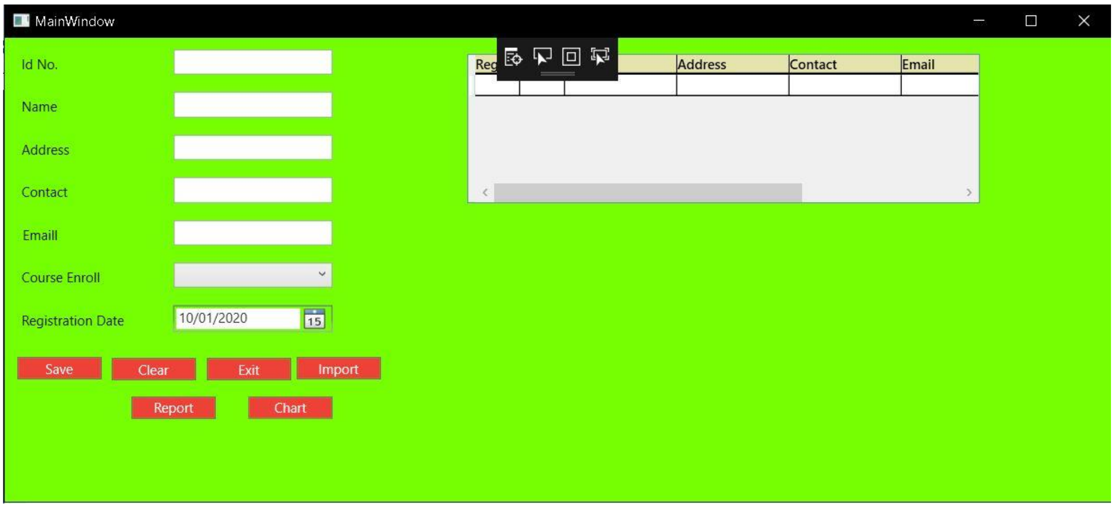
admin

Password

•••••

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.



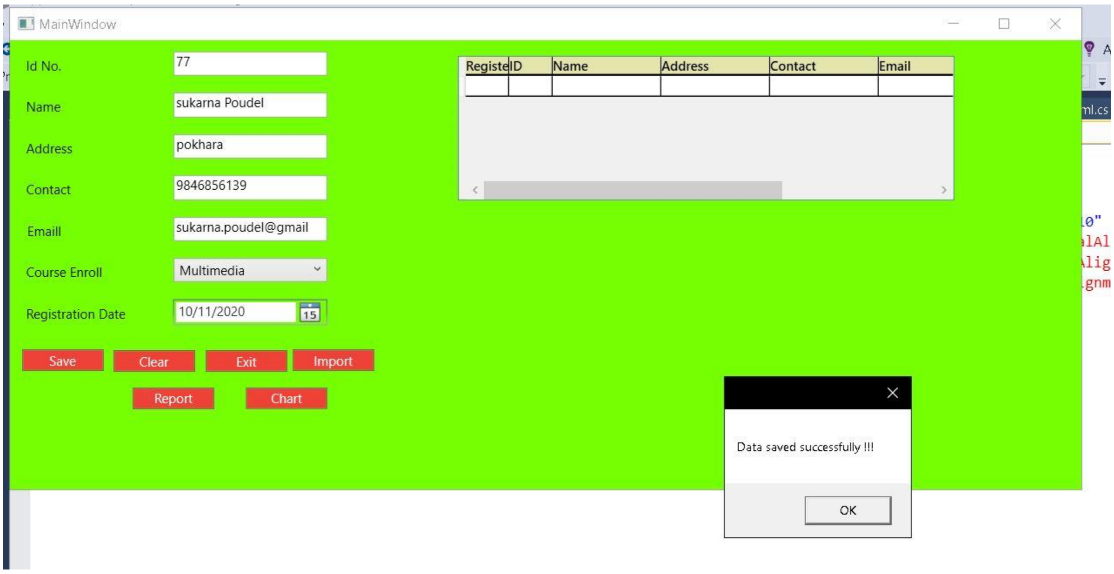
MainWindow

Id No.	
Name	
Address	
Contact	
Emaill	
Course Enroll	
Registration Date	10/01/2020

Reg			Address	Contact	Email

Figure 2 formfor inserting data

In this form students details are filled up.



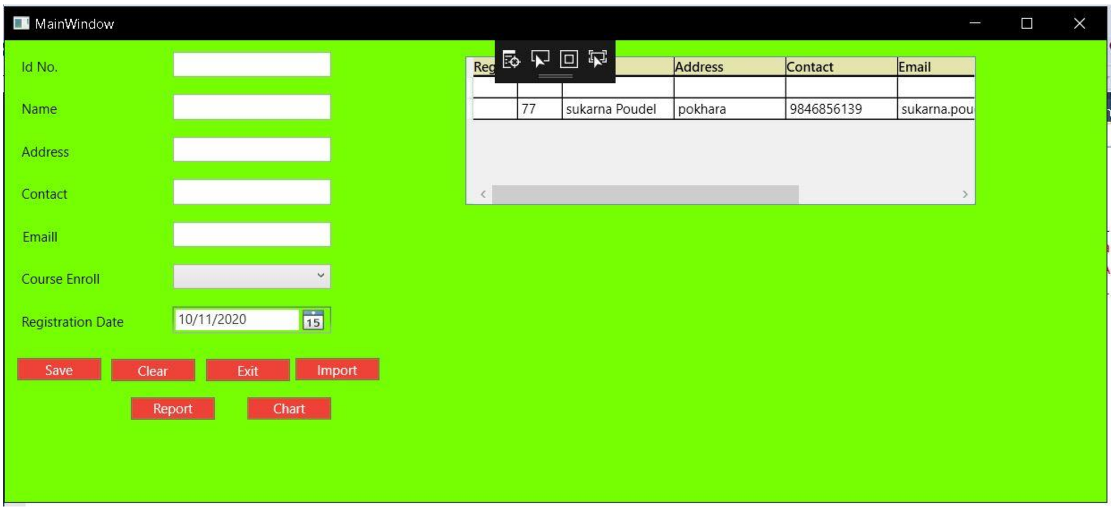
Id No.	77
Name	sukarna Poudel
Address	pokhara
Contact	9846856139
Emaill	sukarna.poudel@gmail
Course Enroll	Multimedia
Registration Date	10/11/2020

RegisteID	Name	Address	Contact	Email

Data saved successfully !!!

Figure 3 data saved successfully

After data is inserted, msg box is generated.

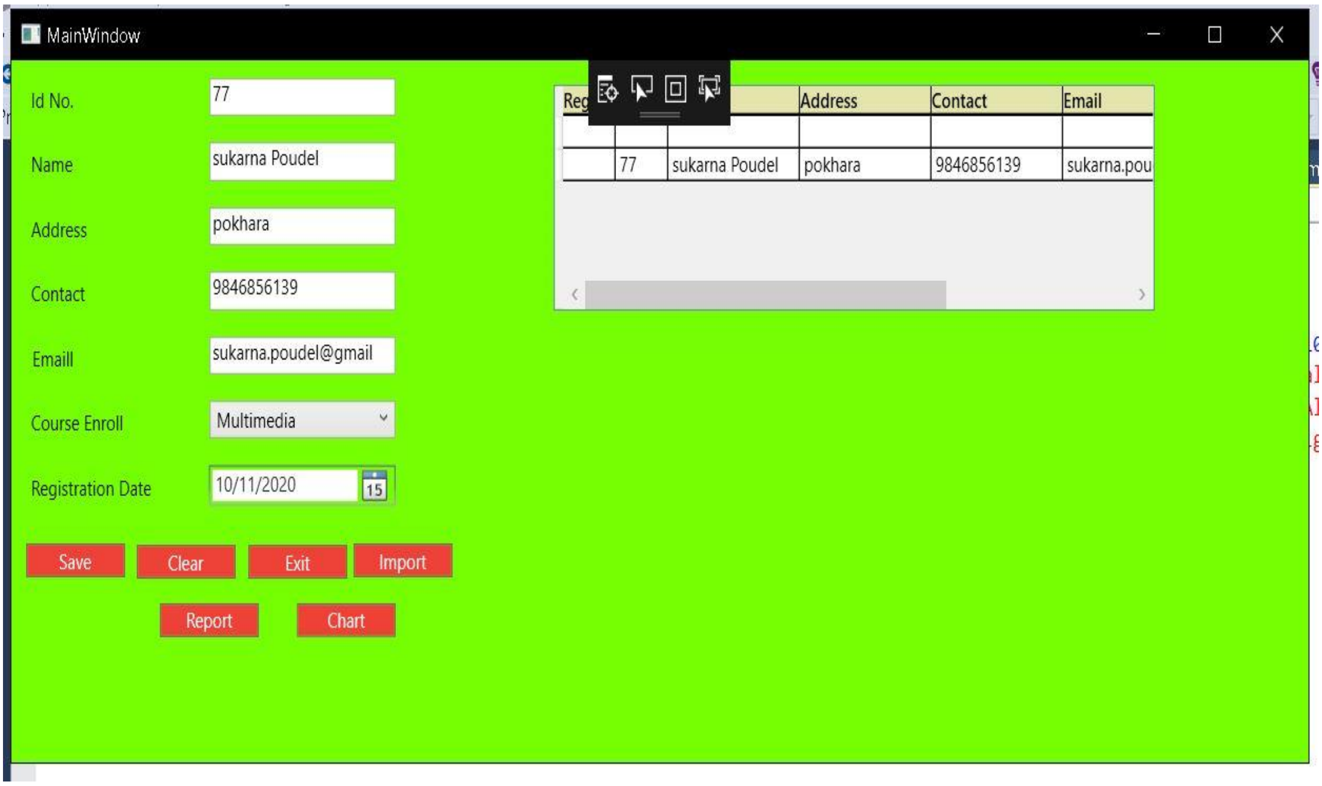


Id No.	
Name	
Address	
Contact	
Emaill	
Course Enroll	
Registration Date	10/11/2020

Reg			Address	Contact	Email
	77	sukarna Poudel	pokhara	9846856139	sukarna.pou

Figure 4 clearing data

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

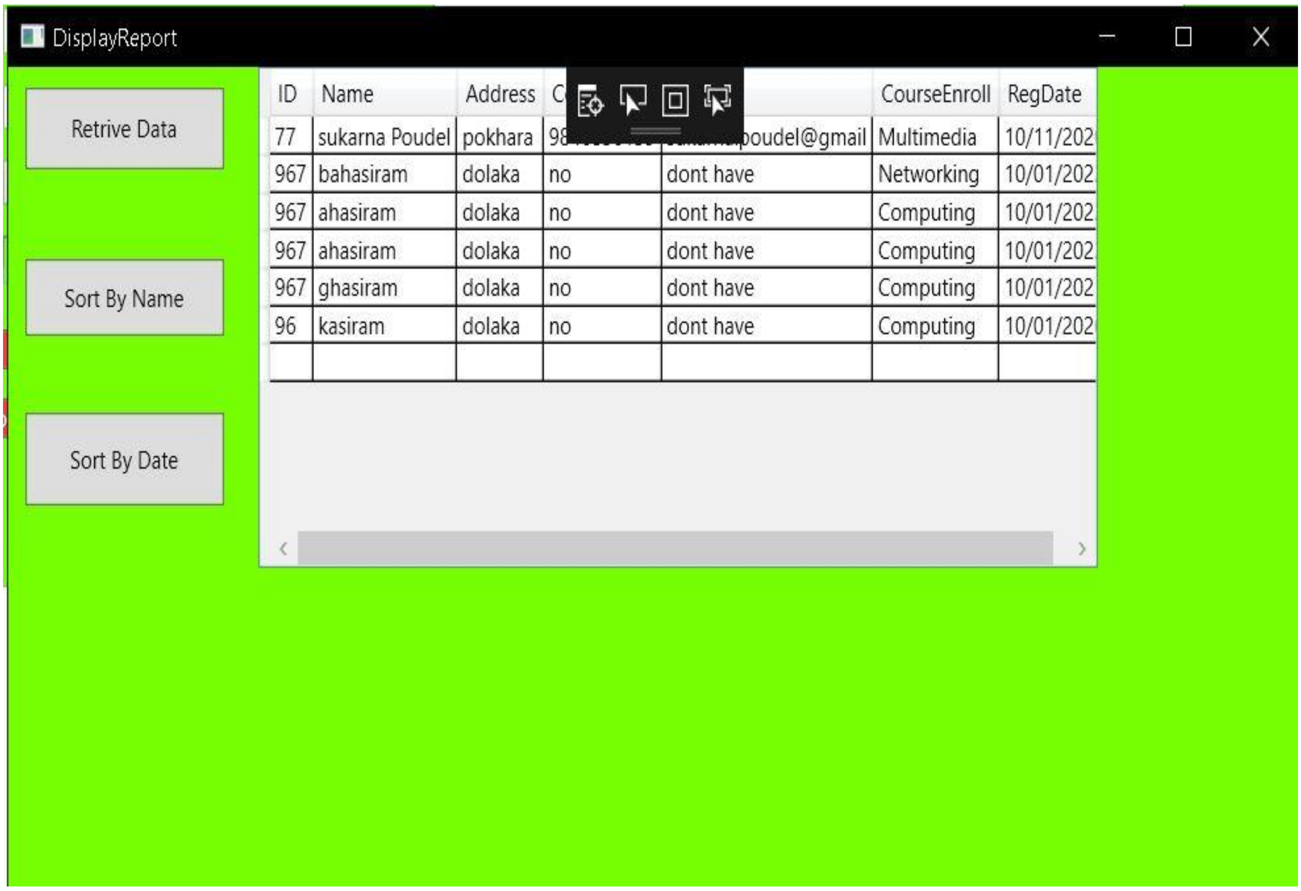


Id No.	77
Name	sukarna Poudel
Address	pokhara
Contact	9846856139
Emaill	sukarna.poudel@gmail
Course Enroll	Multimedia
Registration Date	10/11/2020

Reg			Address	Contact	Email
	77	sukarna Poudel	pokhara	9846856139	sukarna.pou

Figure 5 importing data from xml report

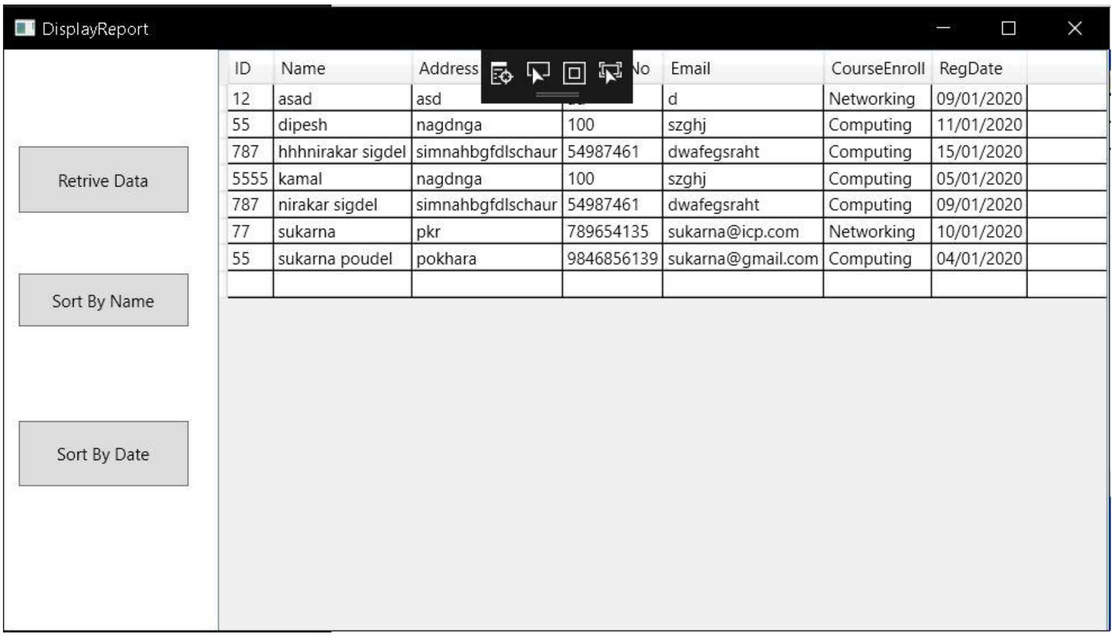
After we import data, all data’s are imported from xml file towards table grid.



ID	Name	Address	C		CourseEnroll	RegDate
77	sukarna Poudel	pokhara	98	poudel@gmail	Multimedia	10/11/202
967	bahasiram	dolaka	no	dont have	Networking	10/01/202
967	ahasiram	dolaka	no	dont have	Computing	10/01/202
967	ahasiram	dolaka	no	dont have	Computing	10/01/202
967	ghasiram	dolaka	no	dont have	Computing	10/01/202
96	kasiram	dolaka	no	dont have	Computing	10/01/202

Figure 6 Displaying Report

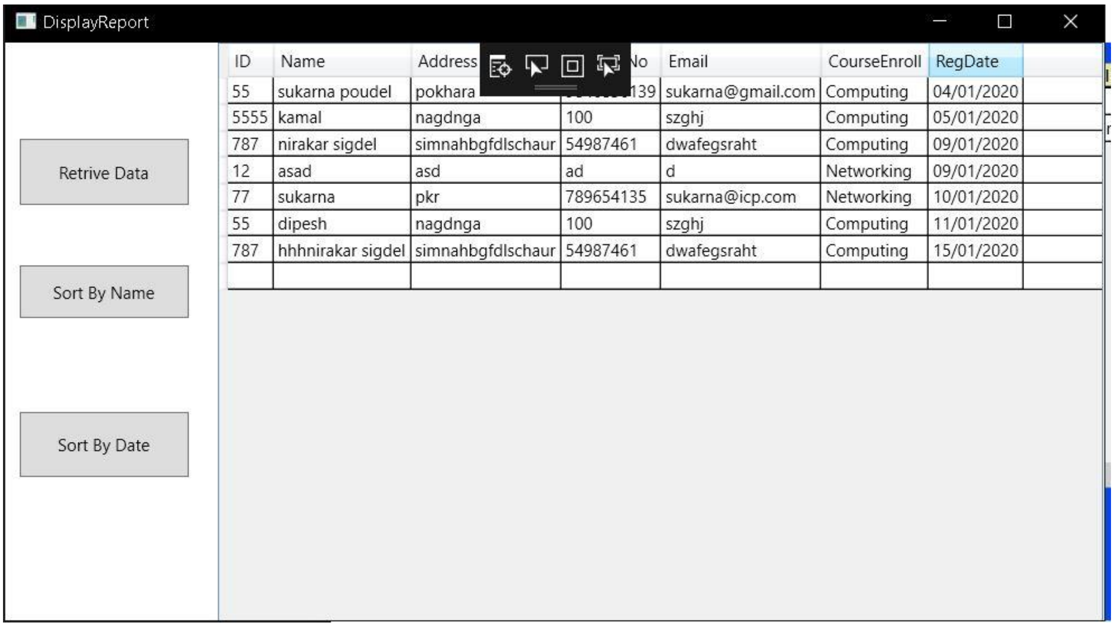
In this part, all the information's that are filled up are displayed in table grid.



ID	Name	Address	No	Email	CourseEnroll	RegDate
12	asad	asd		d	Networking	09/01/2020
55	dipesh	nagdnga	100	szghj	Computing	11/01/2020
787	hhhnirakar sigdel	simnahbgfdlschaur	54987461	dwafegsraht	Computing	15/01/2020
5555	kamal	nagdnga	100	szghj	Computing	05/01/2020
787	nirakar sigdel	simnahbgfdlschaur	54987461	dwafegsraht	Computing	09/01/2020
77	sukarna	pkr	789654135	sukarna@icp.com	Networking	10/01/2020
55	sukarna poudel	pokhara	9846856139	sukarna@gmail.com	Computing	04/01/2020

Figure 7 sorting by name

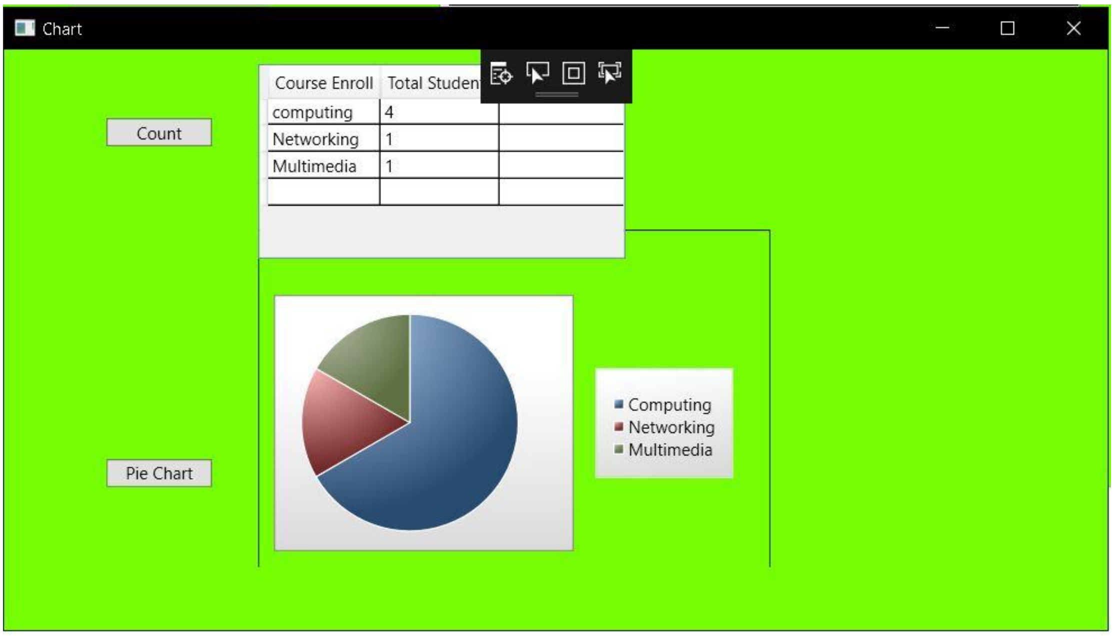
In this part, data's are shown by sorting names.



ID	Name	Address	No	Email	CourseEnroll	RegDate
55	sukarna poudel	pokhara	[illegible]139	sukarna@gmail.com	Computing	04/01/2020
5555	kamal	nagdnga	100	szghj	Computing	05/01/2020
787	nirakar sigdel	simnahbgfdlschaur	54987461	dwafegsraht	Computing	09/01/2020
12	asad	asd	ad	d	Networking	09/01/2020
77	sukarna	pkr	789654135	sukarna@icp.com	Networking	10/01/2020
55	dipesh	nagdnga	100	szghj	Computing	11/01/2020
787	hhhnirakar sigdel	simnahbgfdlschaur	54987461	dwafegsraht	Computing	15/01/2020

Figure 8 sorting by date

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



Course Enroll	Total Studen
computing	4
Networking	1
Multimedia	1

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.

2. Journals /Articles

1. With advanced technology, it seems, comes more and more paperwork. Ironically, this is exactly the opposite of the "paperless society" predicted for years. Fortunately for educators, there are some products available right now that help ease paperwork-intensive tasks. Keeping detailed records of each and every student's grades, class schedules and personal information - including medical, parental and disciplinary information - has long been the bane of educators everywhere. Now add the monumental task of keeping all such records updated district-wide. And every year seems to bring even more regulations and laws that, in turn bring even more paperwork. To keep track of this plethora of data, software firms have responded with a wide range of Student Information System (SIS) programs, gradebook software and miscellaneous administrative packages. (william, 2017)

2. Student Information System is one of the key systems for facilitating the management and development of Higher Education Institutions. Its use for academic decision-making purposes as well as other academic tasks is crucial. Therefore, this paper aims to understand the impact of System Quality, Information Quality and Information Presentation on Student Information System satisfaction of academic and administrative staff. In this study, System satisfaction survey is carried out and factor analysis and regression tests are applied to interpret the collected data. The results show that only Information Quality has direct effect on satisfaction. Then the impact of decision-making as a mediator factor on system satisfaction is measured and the results reveal that System Quality and Information Quality has indirect significant effect whereas Information Presentation does not have direct nor indirect effect on system satisfaction. (ejmste.com, n.d.)

3. Educational institutions should be able to use and utilize information technology as a supporter of operational activities in producing accurate information. Victory University is one of higher education institutions that its operational works process is still manual, it is seen with the condition of data process of student academic value which is still processed offline. By this condition, the problem that arises is a complicated process because of the excessive use of time and paper, causing overhead to the students and the University. Therefore, the researcher conducts research with the title, "Academic Information System for Student (manihutu, 2018)

4. This paper presents a practitioner's view on student system implementations in the Australian university sector. A student information system is a core system of any university and integral to its operations and services to students. These systems are constantly on the list of major projects and at any point in time, a university is either implementing a new system or upgrading an existing one, or planning for either or both. These projects are costly, time-consuming and share common challenges that can be attributed to a combination of factors including software implementations, peculiarities of the individual

institution, the sector, the software supplier and the environment in which it operates. Themes underpinning these challenges are explored and discussed with a view to creating greater understanding of the many facets that come into play. Questions are posed on future needs and directions given the challenges ahead, particularly the major sector reforms. (mukharjee, 2015)

5. The purpose of this study is to explore the role of visual aesthetics as a key in generating satisfaction in student information system (SIS) users and to discover relationships to other antecedents. This work has also studied how gender discriminates those relationships. DeLone and McLean's model of information systems success, visual aesthetic and gender socialisation theory are used as a theoretical framework for the study. An explanatory model was proposed based on the previous literature, and then this model was validated using a sample of undergraduate students. Partial Least Squares was chosen as the approach to conduct the statistical analysis. (McLean)

Algorithms

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

Flowchart

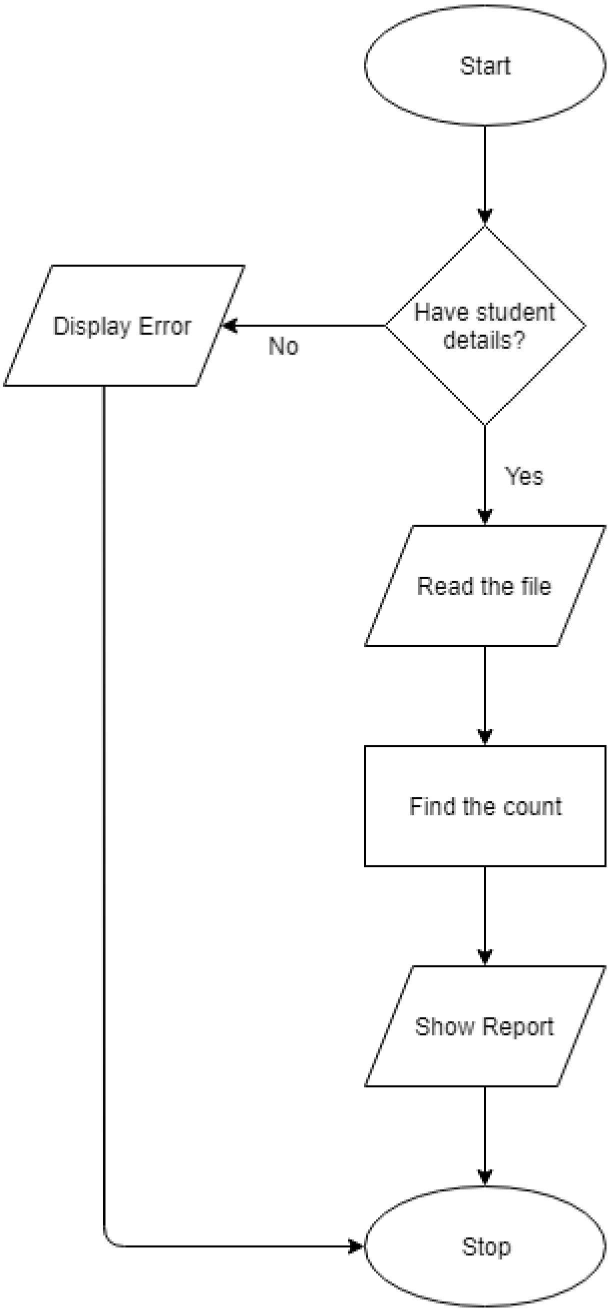


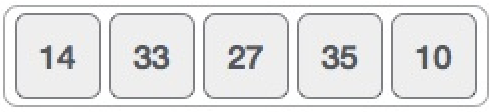
Figure 10 Flowchart for application

Sorting Algorithm

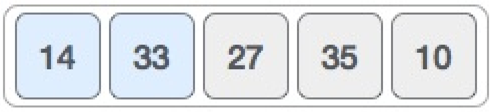
Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of O(n^2) where **n** is the number of items.

How Bubble Sort Works?

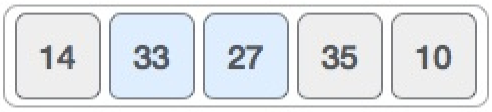
We take an unsorted array for our example. Bubble sort takes O(n^2) time so we're keeping it short and precise.



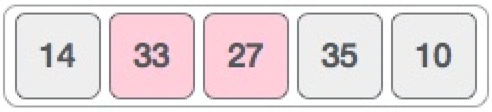
Bubble sort starts with very first two elements, comparing them to check which one is greater.



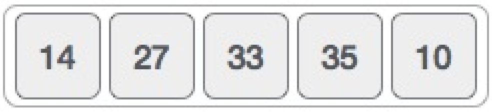
In this case, value 33 is greater than 14, so it is already in sorted locations. Next, we compare 33 with 27.



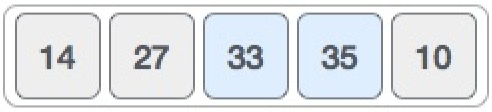
We find that 27 is smaller than 33 and these two values must be swapped.



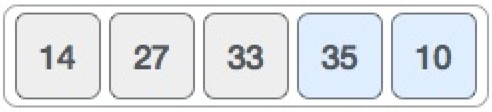
The new array should look like this −



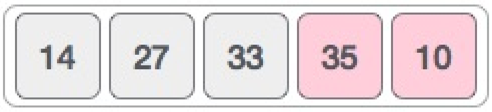
Next we compare 33 and 35. We find that both are in already sorted positions.



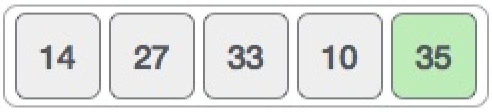
Then we move to the next two values, 35 and 10.



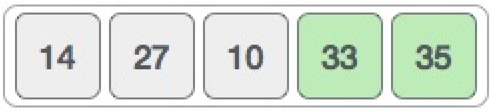
We know then that 10 is smaller 35. Hence they are not sorted.



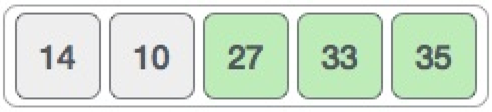
We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this −



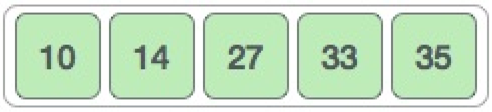
To be precise, we are now showing how an array should look like after each iteration. After the second iteration, it should look like this −



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



And when there's no swap required, bubble sorts learns that an array is completely sorted.



(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 business rationale utilized in the framework mirrors the genuine workplace of the Museum. The GUI planned is profoundly UI and client with fundamental framework organization can work the framework.

An end user can have the offices of include the in Time and Out Time of the student alongside their name. The complete term spend by the student is determined consequently by the framework. The subtleties of the entered guest can be included, for example, Last Name, telephone number, email address and occupation can be included physically by the client. Notwithstanding that, a user can look at the day by day and week by week diagram alongside the rundown. I had some past involvement in Visual Studio. With this experience I had got some in addition to point while doing this coursework. I came to get more working and involvement in the language. Highlights like making graph creating list notwithstanding that, arranging of information structure the framework was another thing for me. Besides, import and sending out to CSV record was new perspective for me.

Conclusion

In this way I have developed student information system which is a desktop application. It took 10 days to complete this task. This coursework taught many things about WPF desktop application. Also this task helped me for improving my programming skills in C# in visual studio. I would like to thank Sachin sir and ishwor sir for guiding me throughout 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_algorithm.htm

william, w. (2017).

Appendix

Main window.xaml.cs

```
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
{
    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);
                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;
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

```
using System;
using System.Collections.Generic;
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();
        }

        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;


        }
    }
}
```

Datahandler.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ApplicationDevelopmentCW
{
    class DataHandler
    {
        public DataSet CreateDataSet()
        {
            var ds = new DataSet();
            ds.Tables.Add(CreateStudentTable());
            return ds;
        }

        private DataTable CreateStudentTable()
        {
            var dt = new DataTable("Student");
            dt.Columns.Add("ID", typeof(string));
            dt.Columns.Add("Name", typeof(string));
            dt.Columns.Add("Address", typeof(string));
            dt.Columns.Add("Contact", typeof(string));
            dt.Columns.Add("Email", typeof(string));
            dt.Columns.Add("CourseEnroll", typeof(string));
            dt.Columns.Add("RegDate", typeof(DateTime));
            return dt;
        }
    }
}
```

Chart.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Controls.DataVisualization.Charting;
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 Chart.xaml
    /// </summary>
    public partial class Chart : Window
    {
        public KeyValuePair<string, int>[] ItemsSource { get; private set; }

        public Chart()
        {
            InitializeComponent();

        }
        private void btnChart_click(object sender, RoutedEventArgs e)
        {
            var dataSet = new DataSet();
            dataSet.ReadXml(@"G:\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);
            DataChartGrid.DataContext = dt.DefaultView;

        }

        private void btnPieChart_Click(object sender, RoutedEventArgs e)
        {
            InitializeComponent();
            LoadPieChartData();
        }


        public void LoadPieChartData()
        {
            var dataSet = new DataSet();
            if (System.IO.File.Exists(@"G:\student.xml"))
            {
                dataSet.ReadXml(@"G:\student.xml");


                DataTable dtStdReport = dataSet.Tables[0];
                int Total_Computing = 0;
                int Total_Networking = 0;
                int Total_Multimedia = 0;

                DataTable Week = new DataTable("WeekTable1");
                Week.Columns.Add("Courses Enrolled", typeof(String));
                Week.Columns.Add("Overall Student", typeof(int));


                for (int i = 0; i < dtStdReport.Rows.Count; i++)
                {

                    String column =
dtStdReport.Rows[i]["CourseEnroll"].ToString();

                    if (column == "Computing")
                    {
                        Total_Computing++;

                    }
                    else if (column == "Networking")
                    {
                        Total_Networking++;

                    }
                    else if (column == "Multimedia")
                    {
                        Total_Multimedia++;

                    }
                }
                Week.Rows.Add("BBA", Total_Computing);
                Week.Rows.Add("BE_IT", Total_Networking);
                Week.Rows.Add("BBS", Total_Multimedia);

                ((PieSeries) chartEnroll).ItemsSource = new
KeyValuePair<string, int>[]{
                        new KeyValuePair<string,int>("Computing",
Total_Computing),
                        new KeyValuePair<string,int>("Networking",
Total_Networking),
                        new KeyValuePair<string,int>("Multimedia",
Total_Multimedia)};

                                 }
            else
            {
                MessageBox.Show("No data to show!");
            }

        }


    }
    }
```

App.cs

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Threading.Tasks;
using System.Windows;

namespace ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}
```

Thank you!!!