Informatics College Pokhara

informatics
college   pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: **Niraj 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	User Interface is complete but not separated and have proper use of controls
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	average work with very limited explanation of the classes and methods used
Flow chart, algoriathms and data sctructures used	average work with very limited explanation and missing diagramatic representation.
Reflective essay	Very poorly written

C. Programming Style

Clarity of code,Popper Naming convention & comments	very poorly written code and no comments at all
System Usability	unusable system

Overall Grade:	**D+**	**D+**

Overall Comment:

Code should be self explainable with less comments. Need some proper naming of the componen and require to add comments on required area.

In overall the code is working and all the functionality seems working and system can be used

CS6004NP

Application Development

Coursework 1

Student Name: Niraj Gurung

London Met ID: 17030726

Date: 2020/01/10

Group: L3 C2

Module Leader

Mr. Ishwor Sapkota

Table of Contents

List of Figures

List of Tables

Introduction

Our task for first coursework of Application Development was to develop a Student Management System. This application will allow the users to store the details like name, address, course selected and other important information of the student enrolled in a CSV file. User can also see the students enrolled along with their details and also add the information in the file from another CSV file. This system also shows how many students are enrolled in specific course. So, this application simply allows the user to keep track of all the student details in a systematic way.

This application was developed using Microsoft Visual Studio 2019. As for the programming language C# was used. This application is a WPF based application. The User Interface of the program was designed using tools available for WPF Application.
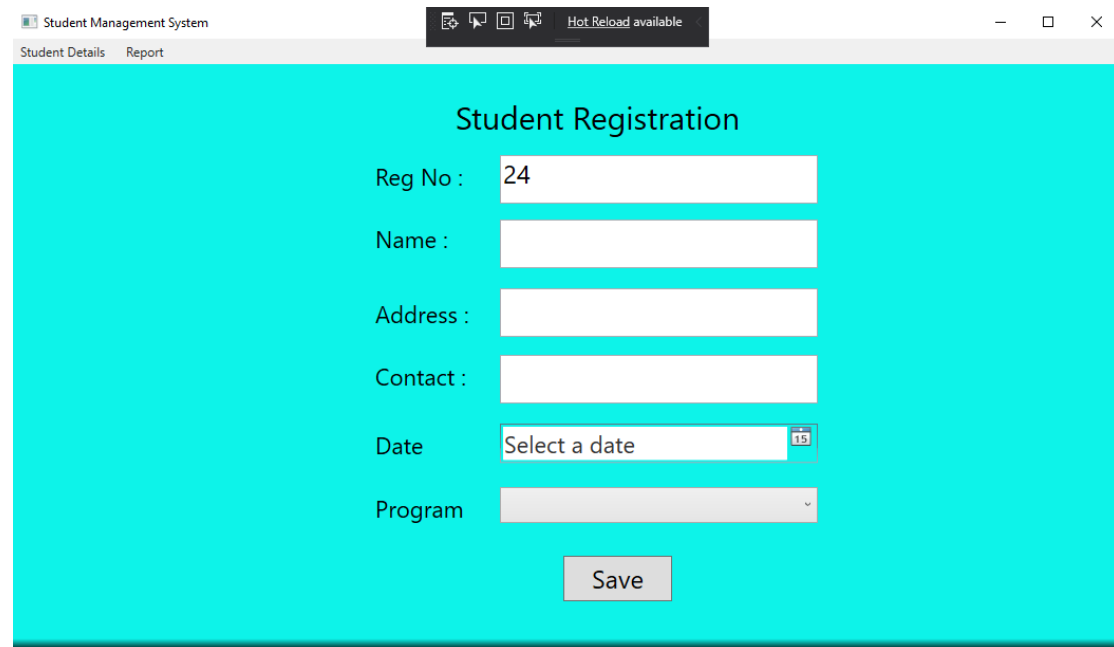
User Manual

The user manual for using the Student Management system is described below:

Introduction to Home Page
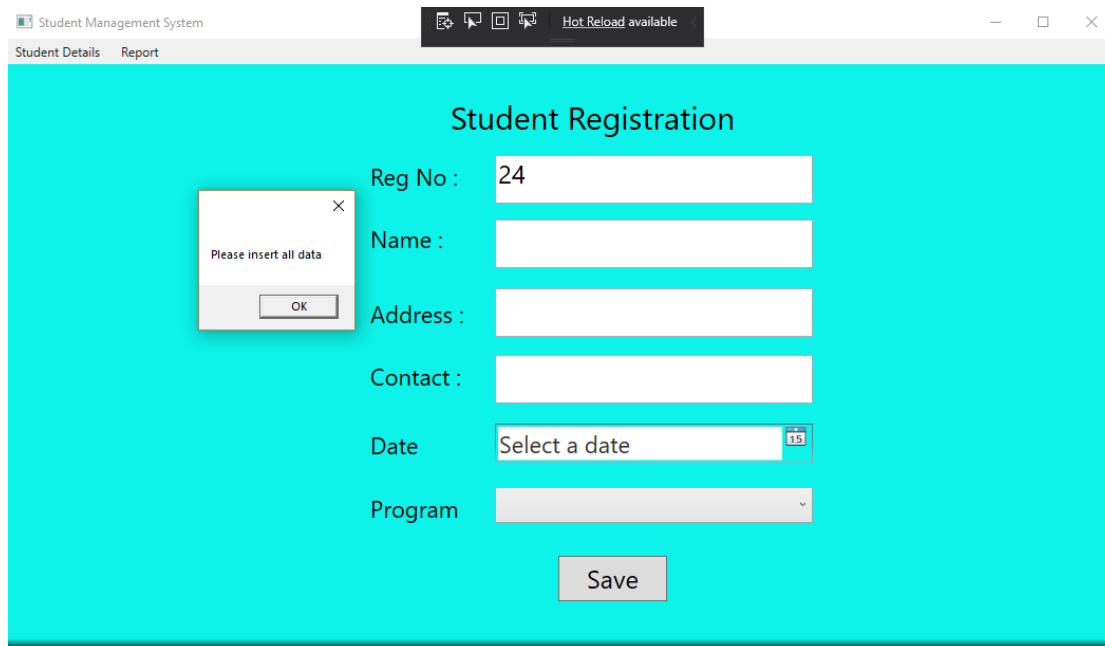
On running the program, the user will see the dashboard.

Figure 1: Home Page

On the top, the system provides menu items like Student Details and Report in the menu bar. These menu items represent the services provided by the system to the user. From the homepage, the user can insert the details of the student and press the submit button to enrol the student into the system.

Figure 2: Empty Fields Validation

Inserting data in the fields is mandatory. If the user misses out to insert the data in the fields of form, the user is asked by the system to insert data in all fields.

Figure 3: Save Student Details

Inside the Student Detail Menu Item, the user can see different UI. Display data button, sort by date, data grid and other components are available in this UI. The content inside the student detail menu item is shown below:
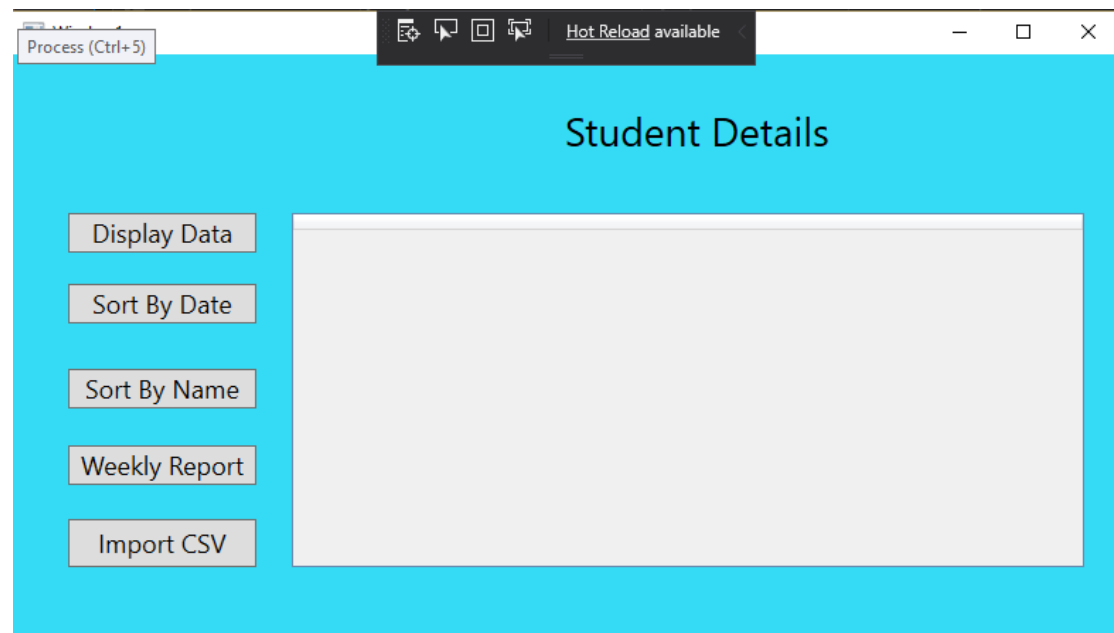
Figure 4: Inside the Student Detail Menu

Display Student Details

Figure 5: Display Available Student Detail

All the students that are enrolled in the system can be retrieved in the datagrid by pressing the display data button.

Sorting
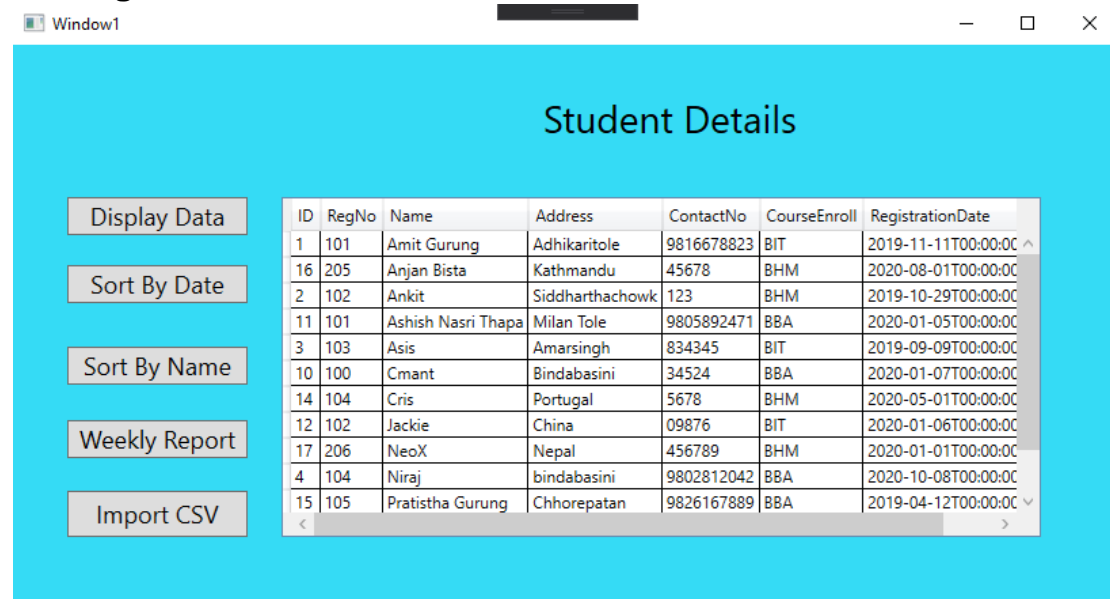
Figure 6: Sort by Name

The user can sort the data in the table in alphabetical order by simply pressing sort by name button.
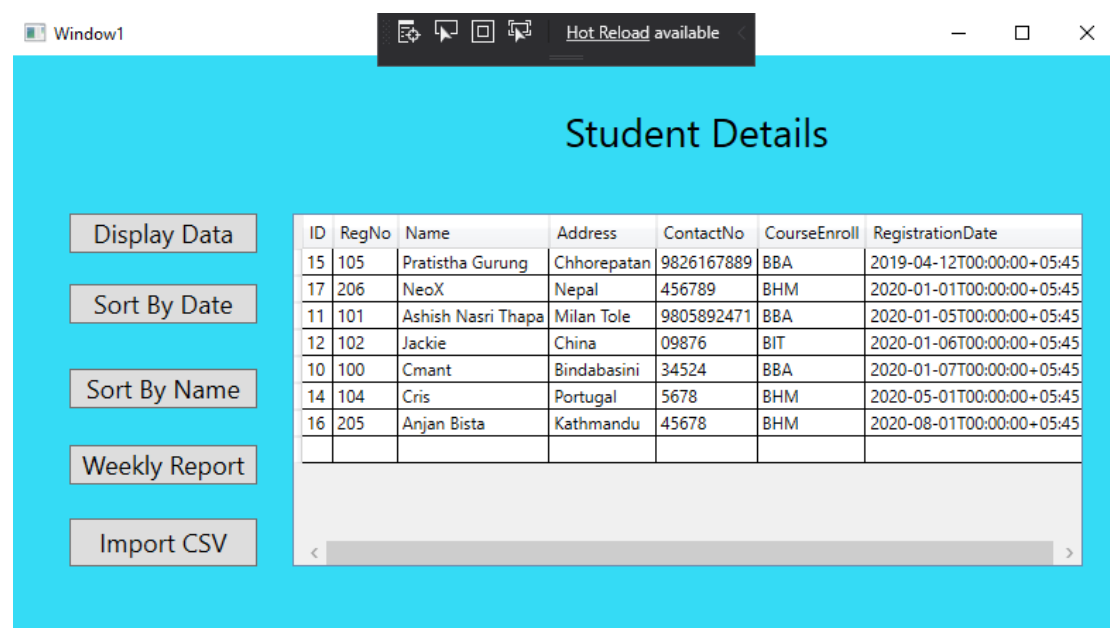
Figure 7: Sort by Date

This system also offers the functionality of sorting the data according to the registration date. For using this functionality, the user has to simply press the sort by date button. This saves a lot of time in professional field.

Importing CSV File
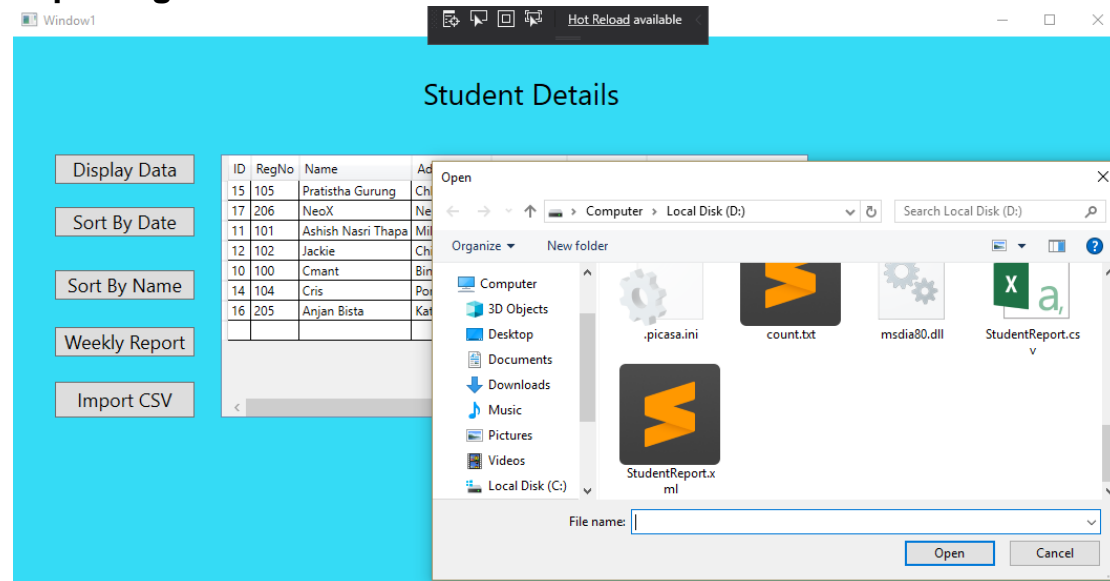
Figure 8: Open File Dialog
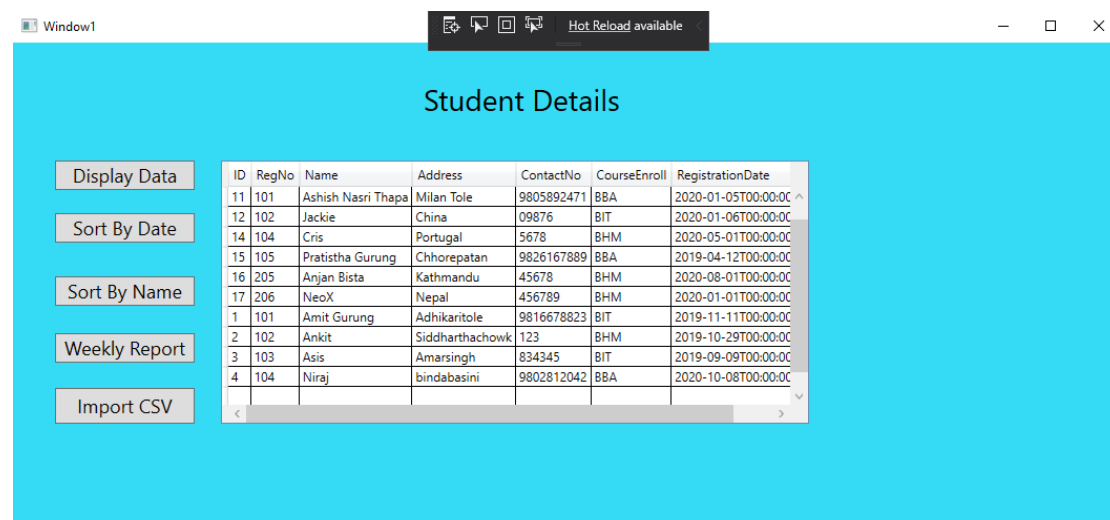
Figure 9: Load CSV File Data in Data Table

For importing the data from CSV file, the user at first has to press import CSV button which opens a dialog box from where the user can select CSV file. Then, the user has to press the display data button to see the result.

Weekly Report
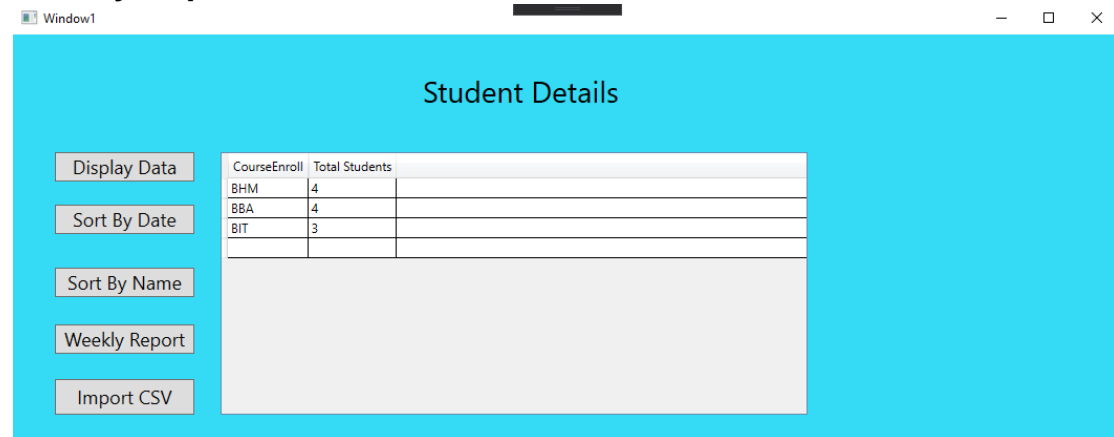
Figure 10: Display Weekly Report

This allows the user to see the total number of students enrolled in the system. To make it more detailed, the system provides the number of students enrolled in specific course. For using this functionality, the user has to press the weekly report button after pressing display data as it has to get data at first from the grid.
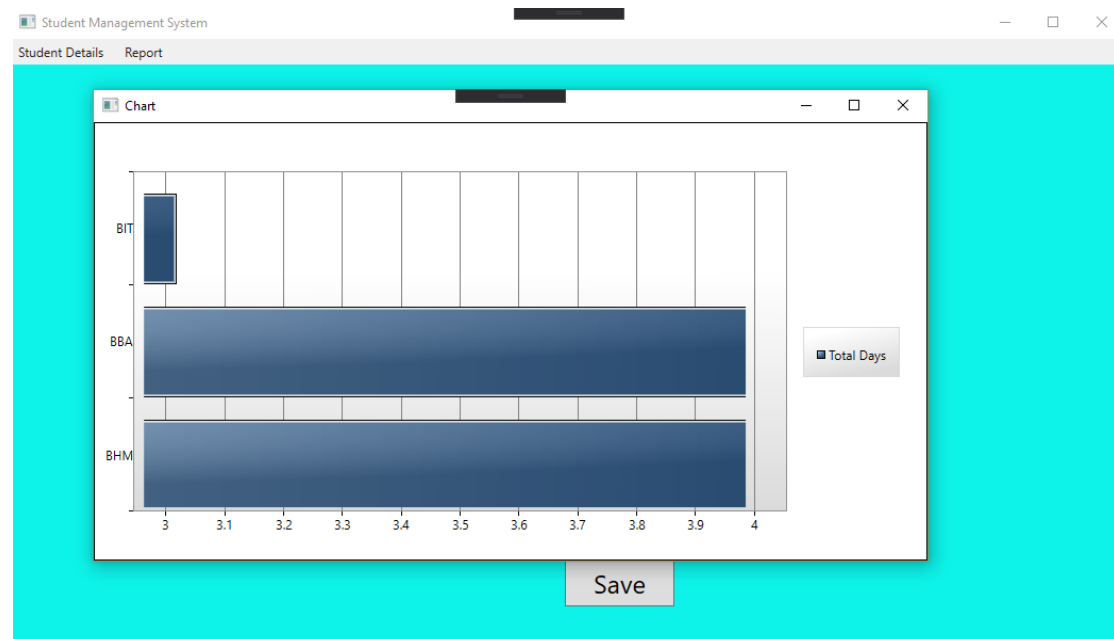
Figure 11: Bar Graph of Weekly Report

The Report Menu on the top of UI displays the weekly report in the form of bar graph. Such diagram helps to easily evaluate the change in the statistics.

Journal Articles

Development of Computerized Enrolment System in rural based higher education system

The purpose of this study was to develop a computerized enrolment system. This system has made the enrolment of student more effective and efficient. By researching in various fields, it was found that the introduction of Enrolment system in school will be a step up in the system of system. This system helps the staff to find record of student faster and easily (Caipang, 2013).

Centralized School Management System for Government School in Ethiopia

Student Management system is used to reduce the burden for teachers and control all the activities in school. But this project was mainly focused to transform the digitalization in government school. Before the use of this type of computerized system, paper and documented materials were used in the school. As the system is not centralized, it is not possible to access the data from other school. This system open ups the path to other school to access the data (S, 2018).

Managing the Digitalization of Schools

In nowadays increasingly ICT-related investments in an overgrowing competitive school setting, the deployment, maintenance, and particularly the effective use of ICT is in many ways a complex multifaceted managerial task involving several stakeholder groups. In a pilot study of a single school district in a Swedish municipality, we have interviewed representatives from the municipal board of education, the municipal IT-support for schools, and two principals. We have used Technology Acceptance Model (TAM3) as our analytical lens to explore how school principals' and municipal IT-managers' perceive ICT adoption, usefulness, and the potential role of ICT. We conclude that the barriers for a successful integration of ICT into school-related activities requires a holistic managerial thinking in order to overcome the lack of coordinated ICT investment strategy and tracking of ICT (Babaheidari, 2014).

System Architecture

Architecture Diagram
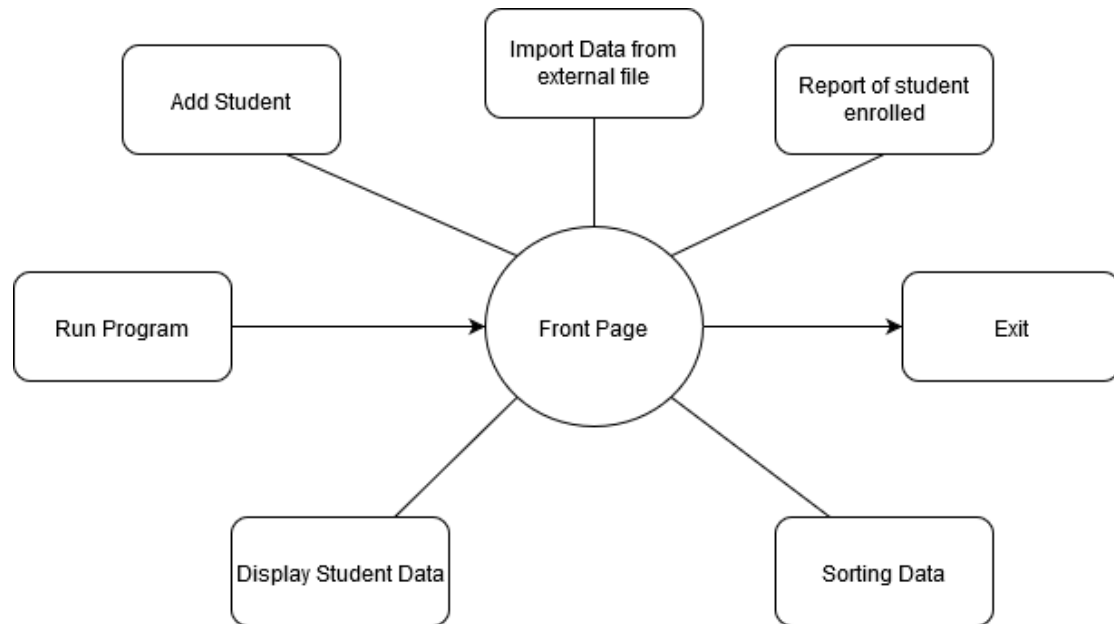
Figure 12: Architecture Diagram

Class Diagram
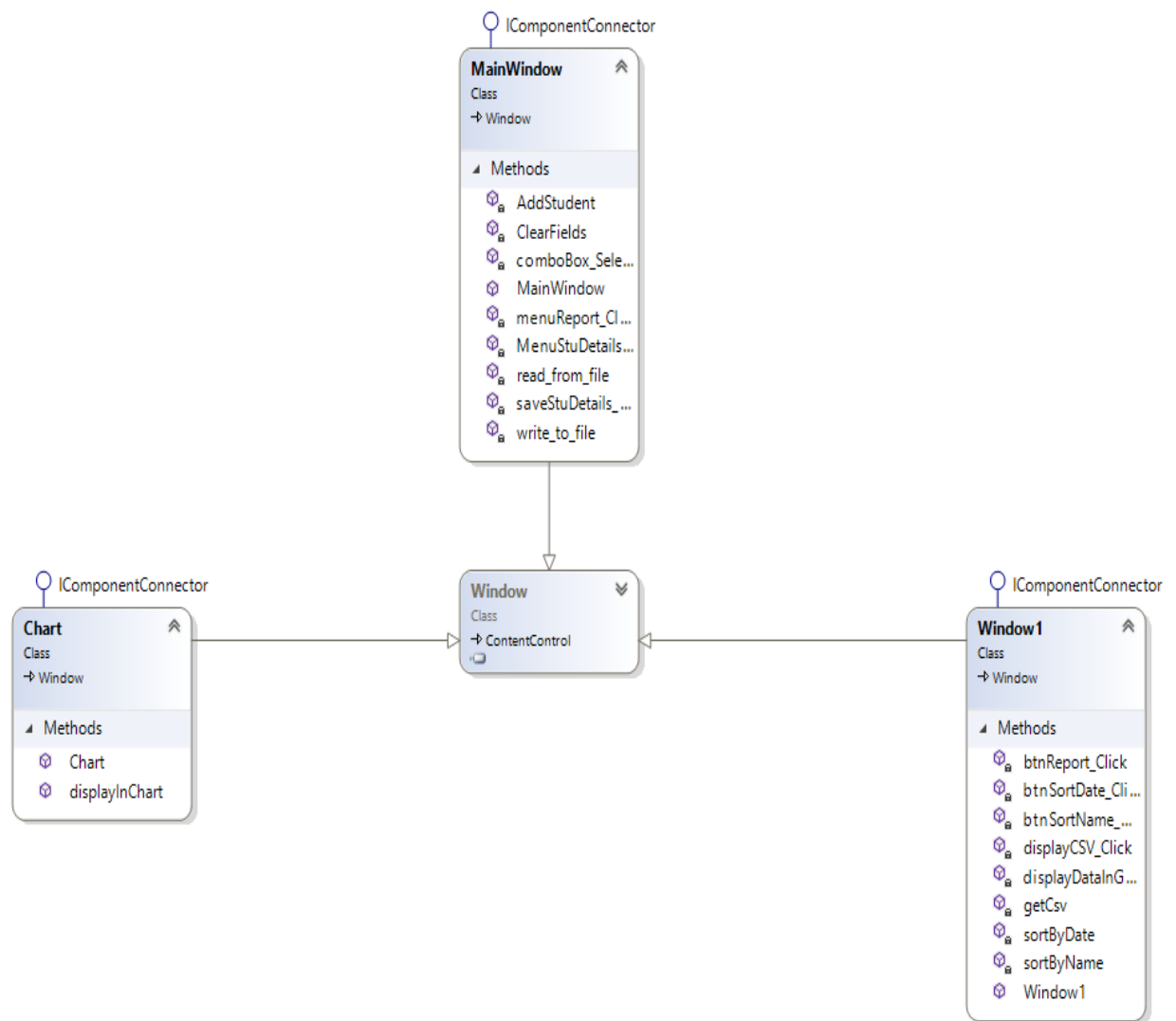
Figure 13: Class Diagram

Individual Diagram

MainWindow
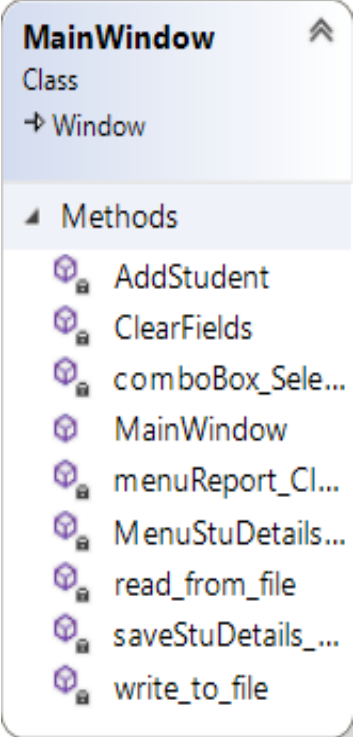
Methods	Description	
AddStudent	Contains validation of the text field and if all text field are filled then the details are stored in xml file.	**MainWindow** Class ↦ Window ▲ Methods 　⊕ AddStudent 　⊕ ClearFields 　⊕ comboBox_Sele... 　⊕ MainWindow 　⊕ menuReport_Cl... 　⊕ MenuStuDetails... 　⊕ read_from_file 　⊕ saveStuDetails_... 　⊕ write_to_file
ClearFields	Clears the text field after inserting the value in xml.	
menuStuDetails_Click	This method opens Windows1.xaml	
read_from_file	This method increments the registration number so that no students get same registration number.	
write_to_file	This method is for writing content in files.	
menuReport_Click	Opens chart.xaml	
saveStuDetails_Click	This methods calls other methods like AddStudent, ClearFileds, etc.	

Table 1: MainWindow

StudentDetails

Methods	Description	
displayDataInGrid_Click	Reads xml file and displays the data in datagrid	**Window1** ⌃ Class → Window
SortByDate	Sorts the date in ascending order	◢ Methods
SortByName	Sorts the name in alphabetical order.	⬡ btnReport_Click
btnSortDate_Click	Calls the sortByDate method	⬡ btnSortDate_Cli...
btnSortName_Click	Calls sortByName function	⬡ btnSortName_... ⬡ displayCSV_Click
displayCSV_Click	Open File Dialog box and calls getCsv method to append the data in datagrid.	⬡ displayDataInG... ⬡ getCsv
GetCsv	Select data from the csv file and append it in data table.	⬡ sortByDate ⬡ sortByName
btnReport_Click	Reads the data in the table and gives the number of student enrolled in specific course.	⬡ Window1
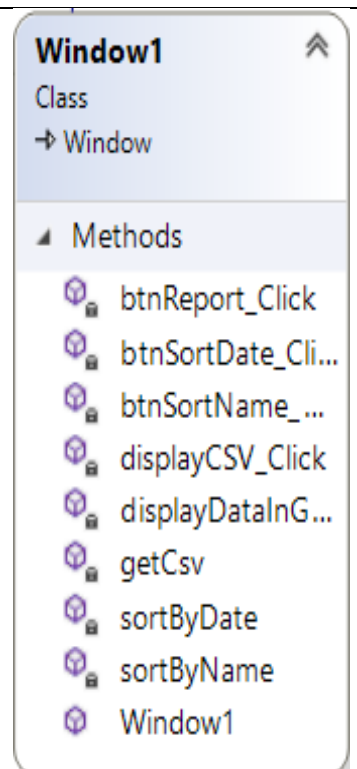
Table 2: StudentDetails

Chart
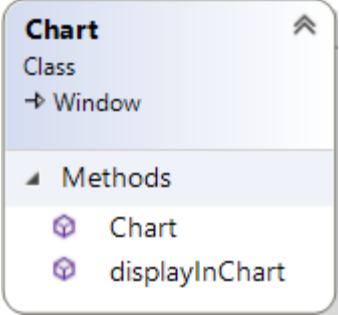
Methods	Description	

Chart	Calls displayInChart method and generates bar diagram	**Chart**
		Class
		↦ Window
		
		▲ Methods
		  ◉ Chart
		  ◉ displayInChart

displayInChart	Converts the students enrolled in specific course into bargraph.	

Table 3: Chart

Flow Chart for Report

Enrol Via Input Form
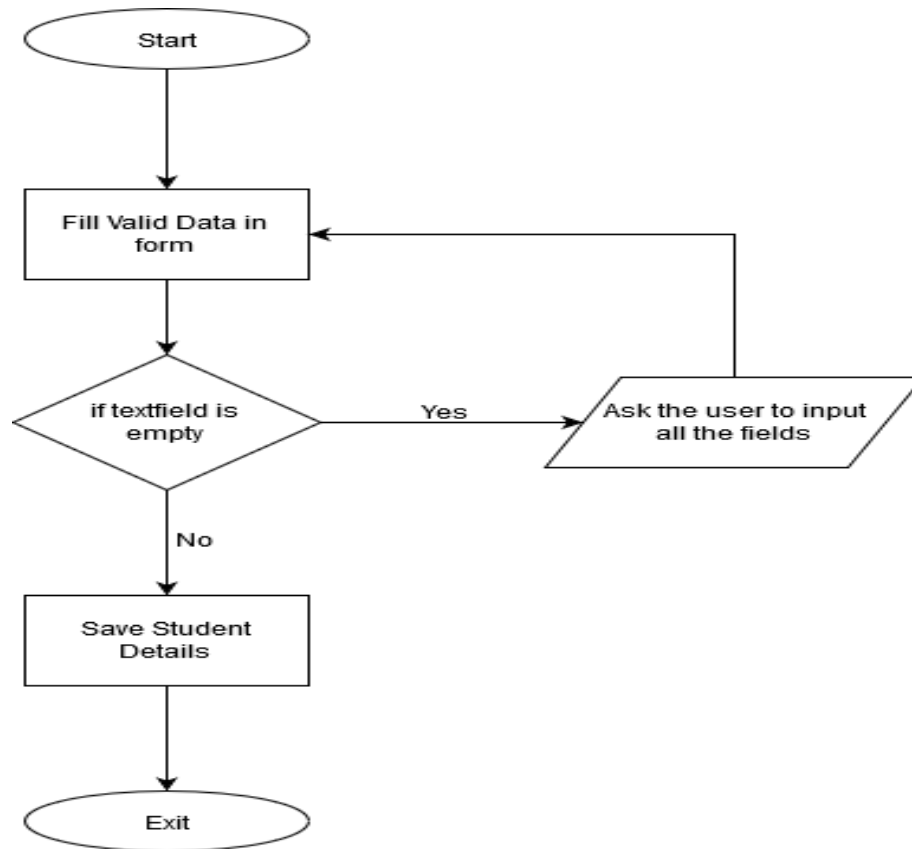
Figure 14: Enrol Via Student Registration Form

Get Student details from CSV File
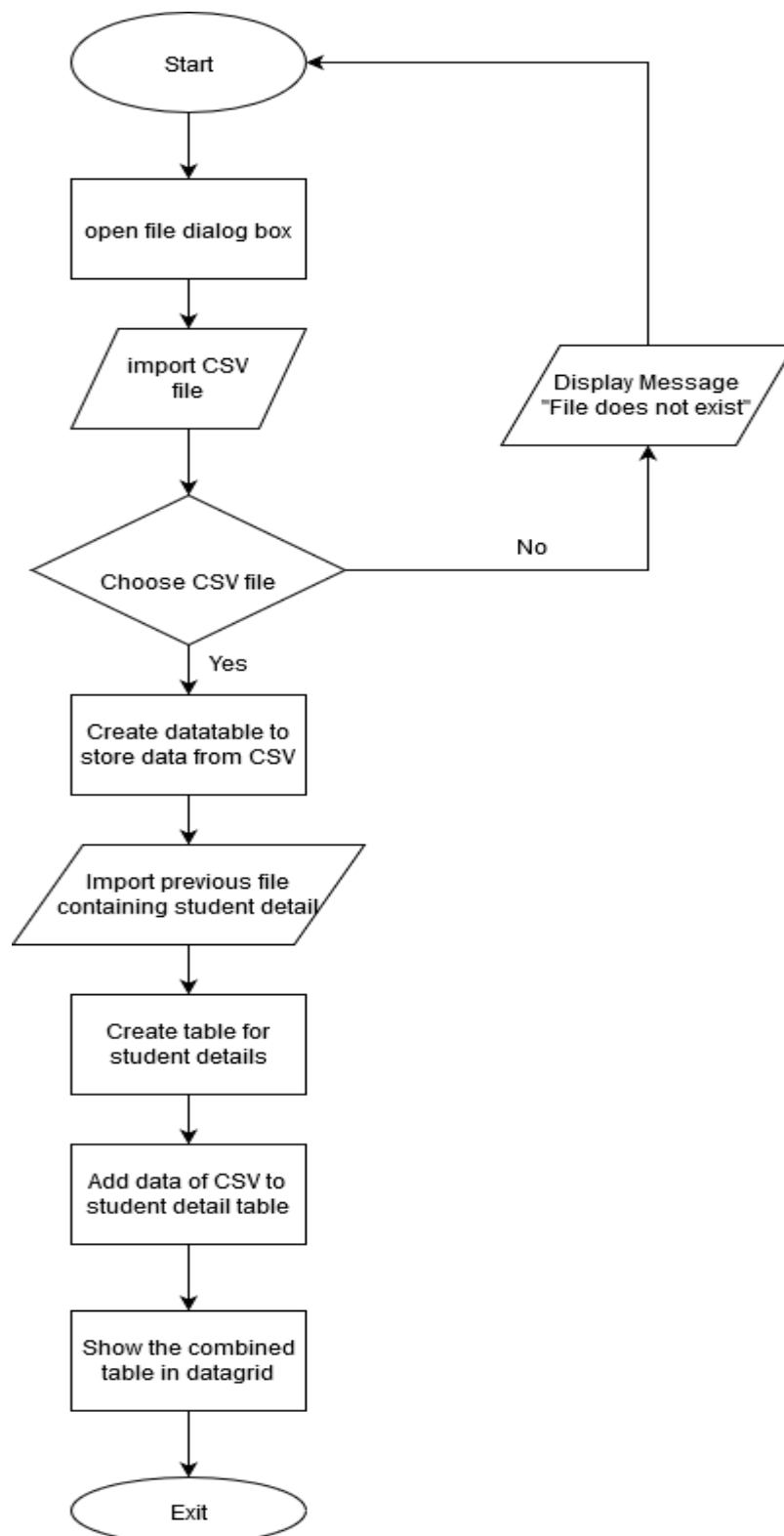
Figure 15: Data Import from CSV

Data Sorting Algorithm

While doing the project, we used bubble sort algorithm in datagrid. It is a simple algorithm that is used to sort set of n elements in form of an array with n number of elements. This algorithm works by comparing the data one by one and sorting them based on their values.

If an array is to be sorted in ascending order then the bubble sort starts comparing the 1st and 2nd item in the array. If the 1st item is greater than the 2nd item then they swap places. Then the comparing goes on up to the last item (Anon., n.d.).

Figure 16: Bubble Sort

This is a simple example of how bubble sort works. The same process can be implemented in datagrid for sorting purposes.

Reflection

Visual Studio 2019 Community Edition was used to develop a management system. Most of the professionals prefer Visual Studio to other IDEs as it provide many services. Developing the application helped us to be familiar with the Visual Studio IDE. Designing program in Visual Studio is a whole new experience as it brings new possibilities in the field of programming.

Creating a whole application that manages student was not an easy task. Not being familiar with C# was the main difficulty. The concept of OOP that were gained in first year came in handy in this project. But then the exciting part was also there. Researching and creating new ideas were the exciting part to it. So, this was a great opportunity for us to create our own application and implement knowledge of OOP.

This coursework helped to improve our programming skills and do tasks creatively.

Conclusion

This coursework required us to build a student management system using C#. While building this system, a lot of problems were faced along the way but by taking guidance from the module teacher and fellow classmates this system was finally built.

After finishing up this coursework in effective way, I am confident that can I can do projects related to C# and have hope that this knowledge that I gained while doing this coursework will help me in the future as well.

Bibliography

Anon., n.d. *Study Tonight.* [Online]
Available at: https://www.studytonight.com/data-structures/bubble-sort
[Accessed 10 January 2020].

Babaheidari, S. M., 2014. *Research Gate.* [Online]
Available at:
https://www.researchgate.net/publication/268514565_Managing_the_Digitaliz
ation_of_Schools_an_exploratory_study_of_school_principals'_and_IT_mana
gers'_perceptions_about_ICT_adoption_and_usefulness?fbclid=IwAR04AV1o
i8RudRUEPTRdL1kaTI45dZITG6y7Ing-aRe4UKZ3mV
[Accessed 10 January 2020].

Caipang, C. L. M. A., 2013. [Online]
[Accessed 10 January 2020].

S, A., 2018. *IJETT.* [Online]
Available at: http://ijettjournal.org/archive/ijett-
v60p213?fbclid=IwAR2AWds8z8uAIjUF7O939Fkh1N7yDjHU1yW2Qcelg4jkl8
OEpdG60KqJ3A8
[Accessed 10 January 2020].

Appendix

MainWindow.xaml.cs

```csharp
using DataHandler;
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.Navigation;
using System.Windows.Shapes;
using System.Data;

namespace CourseWorkSample
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            txtRegNo.Text = Read_from_file();
        }

        private void AddStudent(DataSet dataSet)
        {
            if (txtName.Text=="" || txtAddress.Text=="" || txtContact.Text==""
|| comboBox.Text=="") {
                if (txtName.Text == "" && txtAddress.Text == "" &&
txtContact.Text == "" && comboBox.Text == "") {
                    MessageBox.Show("Please insert all data");
                }
                else if (txtName.Text == "")
                {
                    MessageBox.Show("Please insert Name");
                }
                else if (txtAddress.Text == "")
                {
                    MessageBox.Show("Please insert Address");
                }
                else if (txtContact.Text == "")
                {
                    MessageBox.Show("Please insert Contact Number");
                }
                else if (comboBox.Text == "") {
                    MessageBox.Show("Please select a Course");
                }
            }

            else
            {
```

```
                    var handler = new Handler();


dataSet.Tables["StudentReport"].ReadXml("Files/StudentReport.xml");

            var dr2 = dataSet.Tables["StudentReport"].NewRow();
            dr2["RegNo"] = txtRegNo.Text;
            dr2["Name"] = txtName.Text;
            dr2["Address"] = txtAddress.Text;
            dr2["ContactNo"] = txtContact.Text;
            dr2["CourseEnroll"] = comboBox.Text;
            DateTime? selectedDate = date.SelectedDate;
            if (selectedDate.HasValue)
            {
                string date1 = selectedDate.Value.ToString("yyyy/MM/dd");
                dr2["RegistrationDate"] = date1;
            }
            dataSet.Tables["StudentReport"].Rows.Add(dr2);


dataSet.Tables["StudentReport"].WriteXml("Files/StudentReport.xml");

            ClearFields();

            MessageBox.Show("Data Saved Successfully");
        }
    }


    private void saveStuDetails_Click(object sender, RoutedEventArgs e)
    {
            var handler = new Handler();
            var dataSet = handler.CreateDataSet();
            AddStudent(dataSet);

            var regno = txtRegNo.Text;
            var name = txtName.Text;
            //dataSet.Tables["Student"].WriteXml(@"D:\" + name + "CWData" +
regno + ".xml");

            Write_to_file(txtRegNo.Text);

            txtRegNo.Text = Read_from_file();
    }

    private void Write_to_file(string text)
    {
        System.IO.File.WriteAllText("Files/count.txt", text);
    }


    private string Read_from_file()
    {

        string text = System.IO.File.ReadAllText("Files/count.txt");

        int i;

        if (txtName.Text == "" || txtContact.Text == "" || txtContact.Text
== "")
        {
            i = int.Parse(text.ToString());
```

```
            return i.ToString();
        }
        else
        {
            i = int.Parse(text.ToString());
            i = i + 1;
            return i.ToString();
        }
    }

    private void ClearFields()
    {
        txtName.Text = "";
        txtAddress.Text = "";
        txtContact.Text = "";
    }

    private void MenuStuDetails_Click(object sender, RoutedEventArgs e)
    {
        Window1 studentDetails = new Window1();
        studentDetails.Show();
    }

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

StudentDetails.xaml.cs

```csharp
using DataHandler;
using System;
using System.IO;
using System.Collections.Generic;
using System.Data;
using System.Data.OleDb;
using System.Globalization;
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 CourseWorkSample
{
    public partial class Window1 : Window
    {
        public Window1()
        {
            InitializeComponent();
        }

        private void displayDataInGrid_Click(object sender, RoutedEventArgs e)
        {
            if (System.IO.File.Exists("Files/StudentReport.xml"))
            {
                var handler = new Handler();

                var dataSet = new DataSet();

                dataSet.ReadXml("Files/StudentReport.xml");

                DataTable dtStdReport = new DataTable();
                dtStdReport = dataSet.Tables[0];
                grdStd.DataContext = dtStdReport.DefaultView;
            }
        }

        private void btnSortDate_Click(object sender, RoutedEventArgs e)
        {
            sortByDate();
        }

        private void sortByDate()
        {

            if (System.IO.File.Exists("Files/StudentReport.xml"))
            {
                var dataSet = new DataSet();

                dataSet.ReadXml("Files/StudentReport.xml");

                DataTable dtStdReport = new DataTable();
                dtStdReport = dataSet.Tables[0];
```

```csharp
                dtStdReport.DefaultView.Sort = "RegistrationDate ASC";
                grdStd.DataContext = dtStdReport.DefaultView;
            }
        }

        private void sortByName()
        {
            if (System.IO.File.Exists("Files/StudentReport.xml"))
            {
                var dataSet = new DataSet();

                dataSet.ReadXml("Files/StudentReport.xml");

                DataTable dtStdReport = new DataTable();
                dtStdReport = dataSet.Tables[0];
                dtStdReport.DefaultView.Sort = "Name ASC";
                grdStd.DataContext = dtStdReport.DefaultView;
            }
        }

        private void btnSortName_Click(object sender, RoutedEventArgs e)
        {
            sortByName();
        }

        private void btnReport_Click(object sender, RoutedEventArgs e)
        {
            var dataSet = new DataSet();

            dataSet.ReadXml("Files/StudentReport.xml");

            DataTable dtStdReport = dataSet.Tables[0];

            int BHM = 0;
            int BBA = 0;
            int BIT = 0;

            DataTable dt = new DataTable("Report");

            dt.Columns.Add("CourseEnroll", typeof(String));
            dt.Columns.Add("Total Students", typeof(int));

            for (int i = 0; i < dtStdReport.Rows.Count; i++)
            {
                String col = dtStdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "BHM")
                {
                    BHM++;
                }
                else if (col == "BBA")
                {
                    BBA++;
                }
                else if (col == "BIT")
                {
                    BIT++;
                }
            }
            dt.Rows.Add("BHM", BHM);
            dt.Rows.Add("BBA", BBA);
            dt.Rows.Add("BIT", BIT);
            grdStd.DataContext = dt.DefaultView;
```

```
        }

        private void displayCSV_Click(object sender, RoutedEventArgs e)
        {
            Microsoft.Win32.OpenFileDialog dialog = new
Microsoft.Win32.OpenFileDialog();
            dialog.DefaultExt = ".csv";
            Nullable<bool> result = dialog.ShowDialog();

            if (result == true)
            {
                DataTable tableStd = getCsv(dialog.FileName, true);
                DataTable dataTable = tableStd.Clone();
                dataTable.Columns["ContactNo"].DataType = typeof(String);
                dataTable.Columns["RegNo"].DataType = typeof(String);
                //dataTable.Columns["RegistrationDate"].DataType =
typeof(String);

                foreach (DataRow row in tableStd.Rows) {
                    dataTable.ImportRow(row);
                }
                Handler handler = new Handler();
                DataSet dataSet = handler.CreateDataSet();

dataSet.Tables["StudentReport"].ReadXml("Files/StudentReport.xml");
                dataSet.Tables["StudentReport"].Merge(dataTable);

dataSet.Tables["StudentReport"].WriteXml("Files/StudentReport.xml");

                var dataset1 = new DataSet();
                dataSet.ReadXml("Files/StudentReport.xml");
                DataTable dataTable1 = dataset1.Tables["StudentReport"];
                grdStd.DataContext = tableStd.DefaultView;
            }
        }

        static DataTable getCsv(string path, bool isFirstRowHeader)
        {
            string header = isFirstRowHeader ? "Yes" : "No";

            string pathOnly = System.IO.Path.GetDirectoryName(path);
            string fileName = System.IO.Path.GetFileName(path);

            string sql = @"SELECT * FROM [" + fileName + "]";

            using (OleDbConnection connection = new OleDbConnection(
                    @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
pathOnly +
                    ";Extended Properties=\"Text;HDR=" + header + "\""))
            using (OleDbCommand command = new OleDbCommand(sql, connection))
            using (OleDbDataAdapter adapter = new OleDbDataAdapter(command))
            {
                DataTable dataTable = new DataTable();
                dataTable.Locale = CultureInfo.CurrentCulture;
                adapter.Fill(dataTable);
                return dataTable;
            }
        }
    }
}
```

Chart.xaml.cs

```csharp
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 CourseWorkSample
{
    /// <summary>
    /// Interaction logic for Chart.xaml
    /// </summary>
    public partial class Chart : Window
    {

        public Chart()
        {
            InitializeComponent();
            displayInChart();
        }

        public void displayInChart() {

            var dataSet = new DataSet();

            dataSet.ReadXml(@"D:\StudentReport.xml");

            DataTable dtStdReport = dataSet.Tables[0];

            int BBA = 0;
            int BHM = 0;
            int BIT = 0;

            DataTable dt = new DataTable("Report");

            dt.Columns.Add("CourseEnroll", typeof(String));
            dt.Columns.Add("Total Students", typeof(int));

            for (int i = 0; i < dtStdReport.Rows.Count; i++)
            {
                String col = dtStdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "BHM")
                {
                    BHM++;
                }
                else if (col == "BBA")
                {
                    BBA++;
                }
                else if (col == "BIT")
```

```
            {
                BIT++;
            }
        }

        dt.Rows.Add("BIT", BIT);
        dt.Rows.Add("BHM", BHM);
        dt.Rows.Add("BBA", BBA);

        var item = BHM.ToString();

        ((BarSeries)gridchart).ItemsSource =
        new KeyValuePair<string, int>[]{
            new KeyValuePair<string,int>("BHM", BHM),
            new KeyValuePair<string,int>("BBA", BBA),
            new KeyValuePair<string,int>("BIT", BIT)
        };
    }
  }
}
```