Marking Scheme

# Informatics College Pokhara

informatics
college pokhara

## Application Development

## CS6004NI

## Course Work 1


**Submitted By:** **Soniya Gurung**
**London Met ID:** Enter ID Here

**Submitted To:** Ishwor Sapkota
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** | appropriate use of data types but missing some properties required or missing CRUD operation |
| **Data Validation** | missing most of the validation |
| **Enrollment Report & weekly report in tabular format** | Any one of the report is missing or not complete |
| **Course wise enrollment report & Chart display** | any one component is missing or inappropriate data is shown |
| **Algorithm used for sorting & proper sorting of data** | Sorting not implemented |
| **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** | 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.

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



**Application Development**

**CS6004NP**

**Coursework 1**

**Year and Semester**

**2018-19 Autumn**

**Submitted by:**

Student Name: Soniya Gurung

Student ID: 17030747

Group: L3C1

Submission Date: 10 Jan 2020

**Submitted to:**

Mr. Ishwor Sapkota

# Table of Contents

Table of Figure

# 1. Introduction

The coursework given to us is all about designing and implementing Student Information System in C# and should be desktop application. Here, I have use Visual Studio 2019 to complete my coursework. The application is used to keep track of the student's details, program enrol and registration date. The application allows the user to input student personal detail including course enrol and registration date so that a system can generate a weekly enrolment report of system. System also includes detail information of Student like name, address, contact no., email, program and registration date. The application also imports CSV file and student details are sort by name and registration date. Chart showing total number of students on each enrol program is also displayed.

Now-a-days almost every schools, colleges and universities use such type of application to keep the record of the students. It makes easier and faster to keep track of students. This type of application should be implement in every field like in government offices, hospitals etc. to keep track of all the details easily.

## 2. User manual

In Visual Studio 2019, I have design and implement Student Information System application. The screenshots are shown below which will show how the system operate.

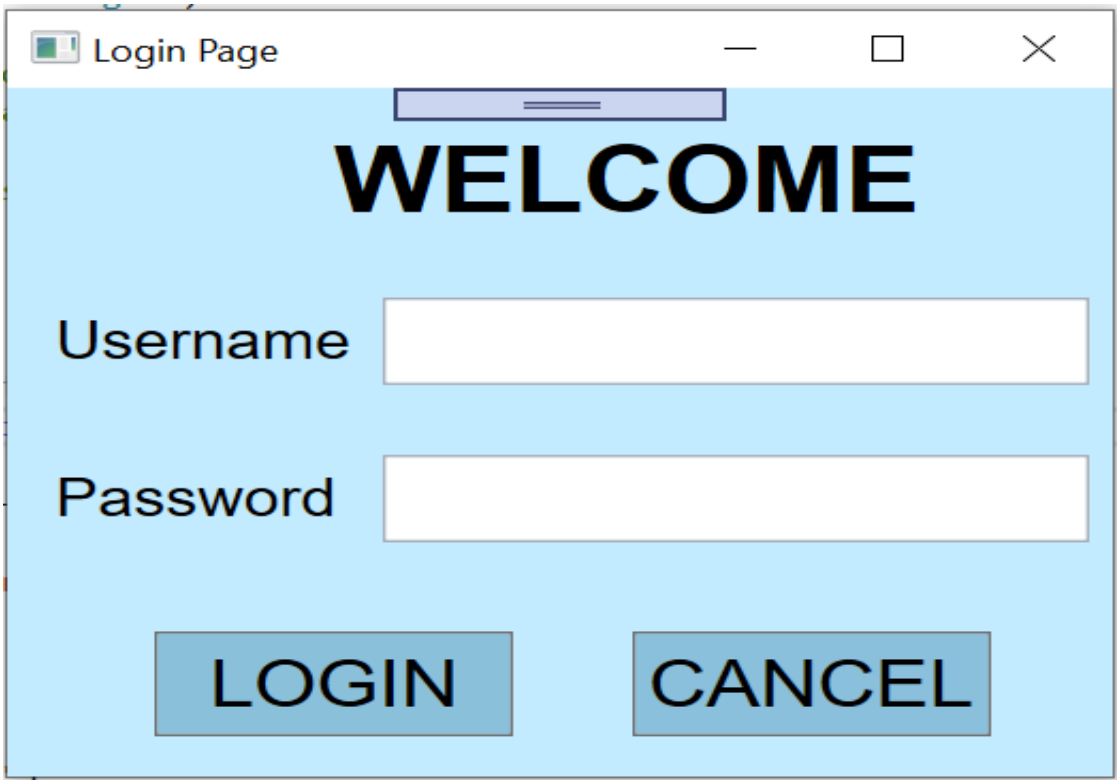



*Figure 1: Login Screen*

After running the application in Visual Studio 2019, we can see above login screen. Here we have declared username and password as “admin”. So for successful login we have to enter correct username and password. When wrong username and password is entered, it will show invalid message box.

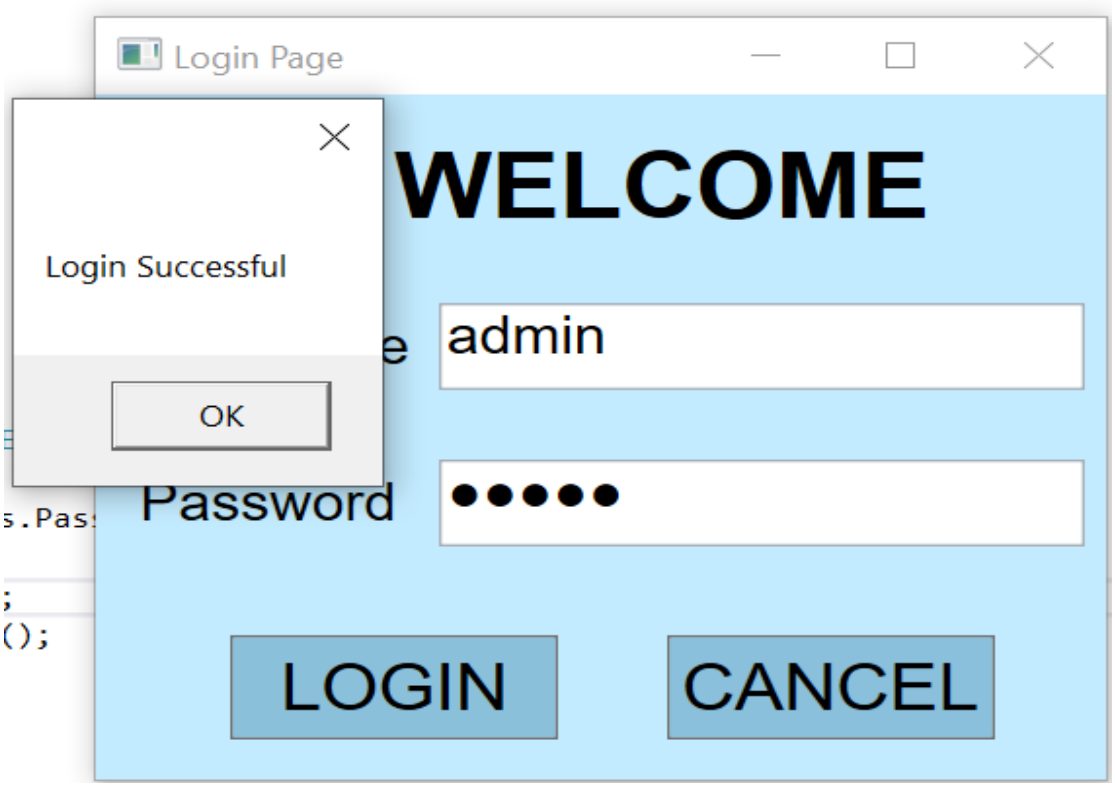



*Figure 2: Login Successful message*

This figure shows login successful message when the user enters the correct username and password.

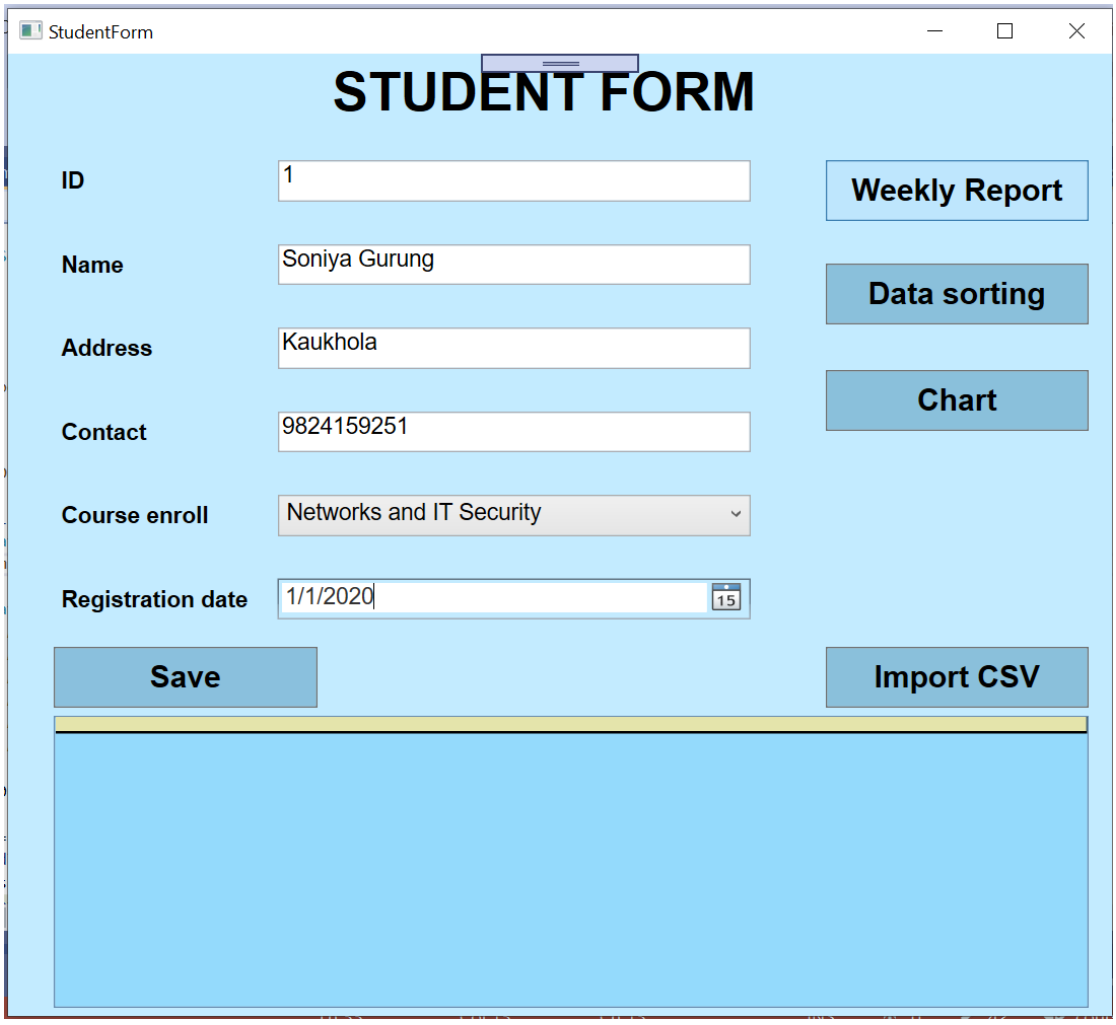



*Figure 3: Student Form*

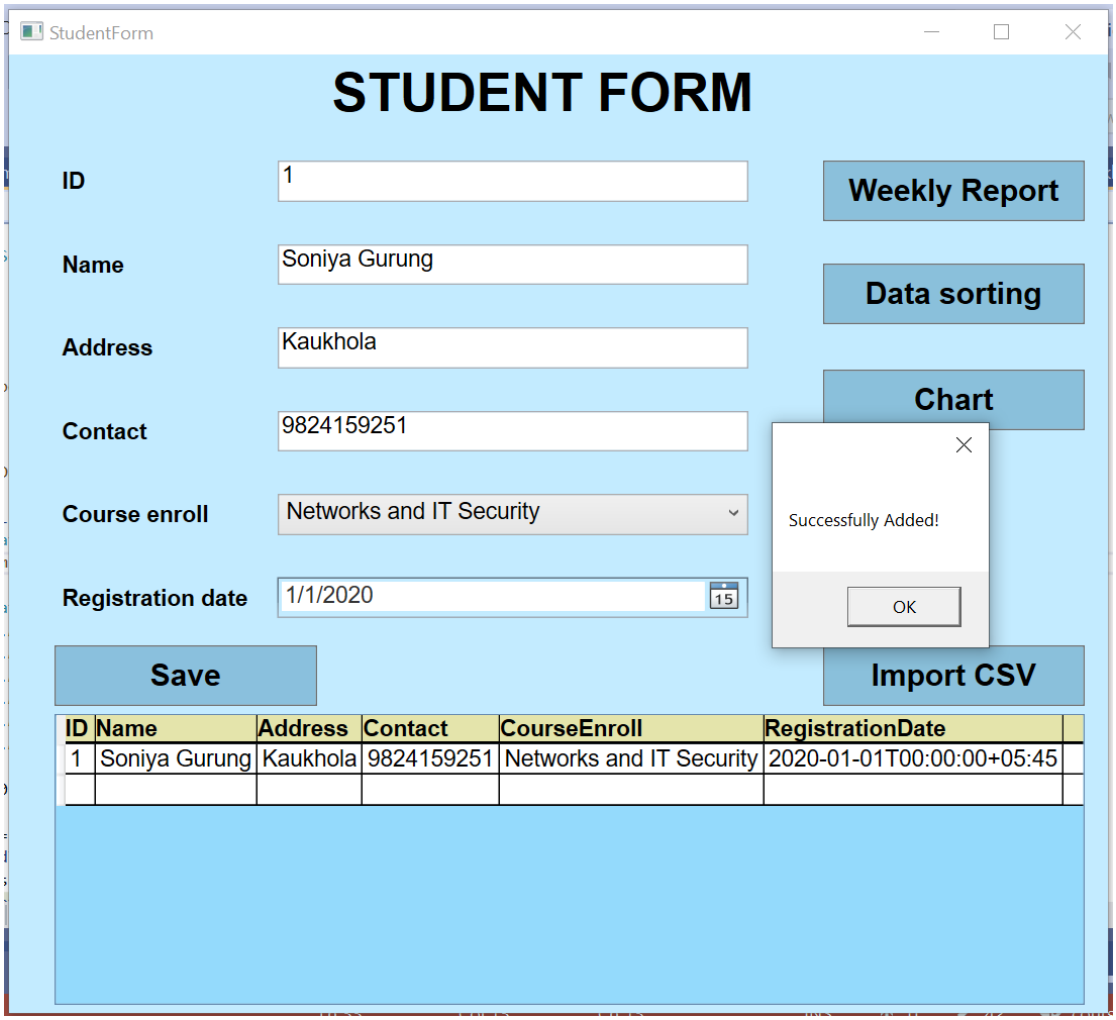



*Figure 4: Student added message*

Student form will appear after successful login. Here, students should include their detail along with course enrol and registration date. Then the details are saved in a grid shown below. There are other buttons like weekly report, sorting, chart and import CSV. There explanation is further given below in next figure.

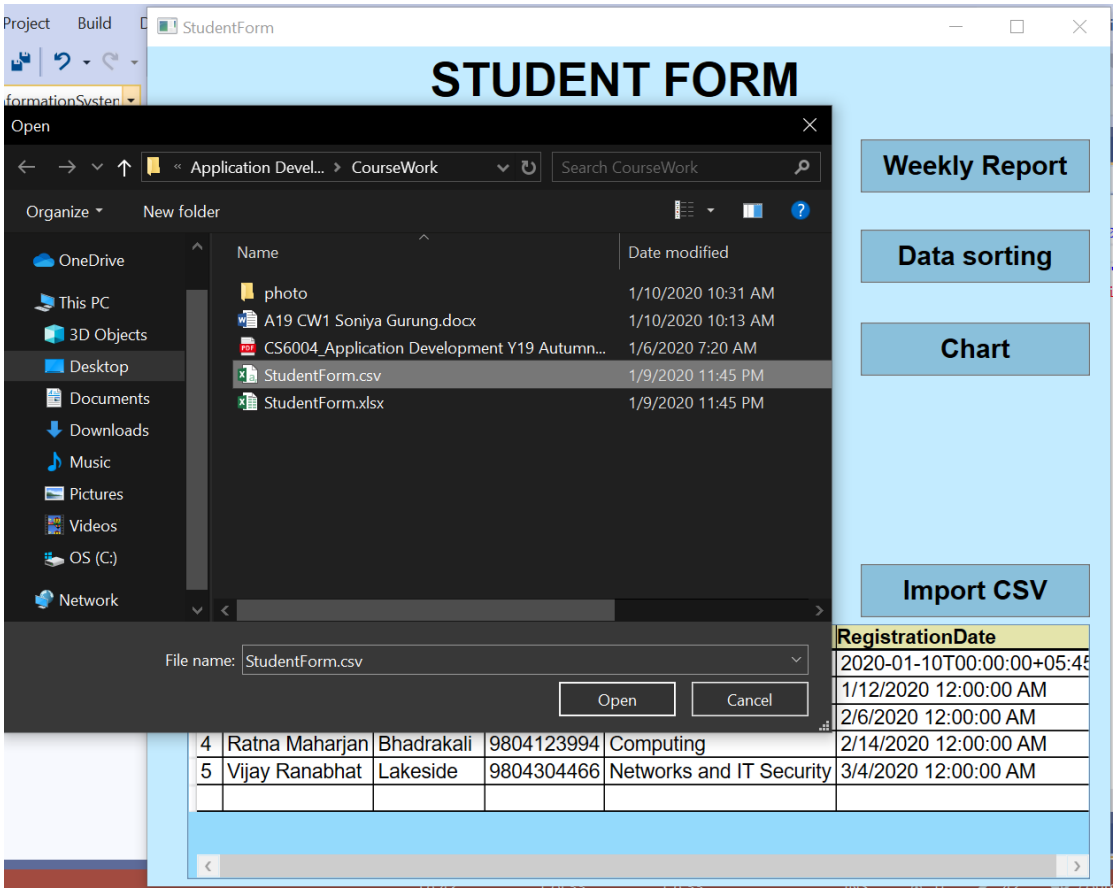



*Figure 5: Importing CSV*

We can import CSV by clicking on import CSC button. Dialog box containing csv file will appear and we have to select the csv file and open it.

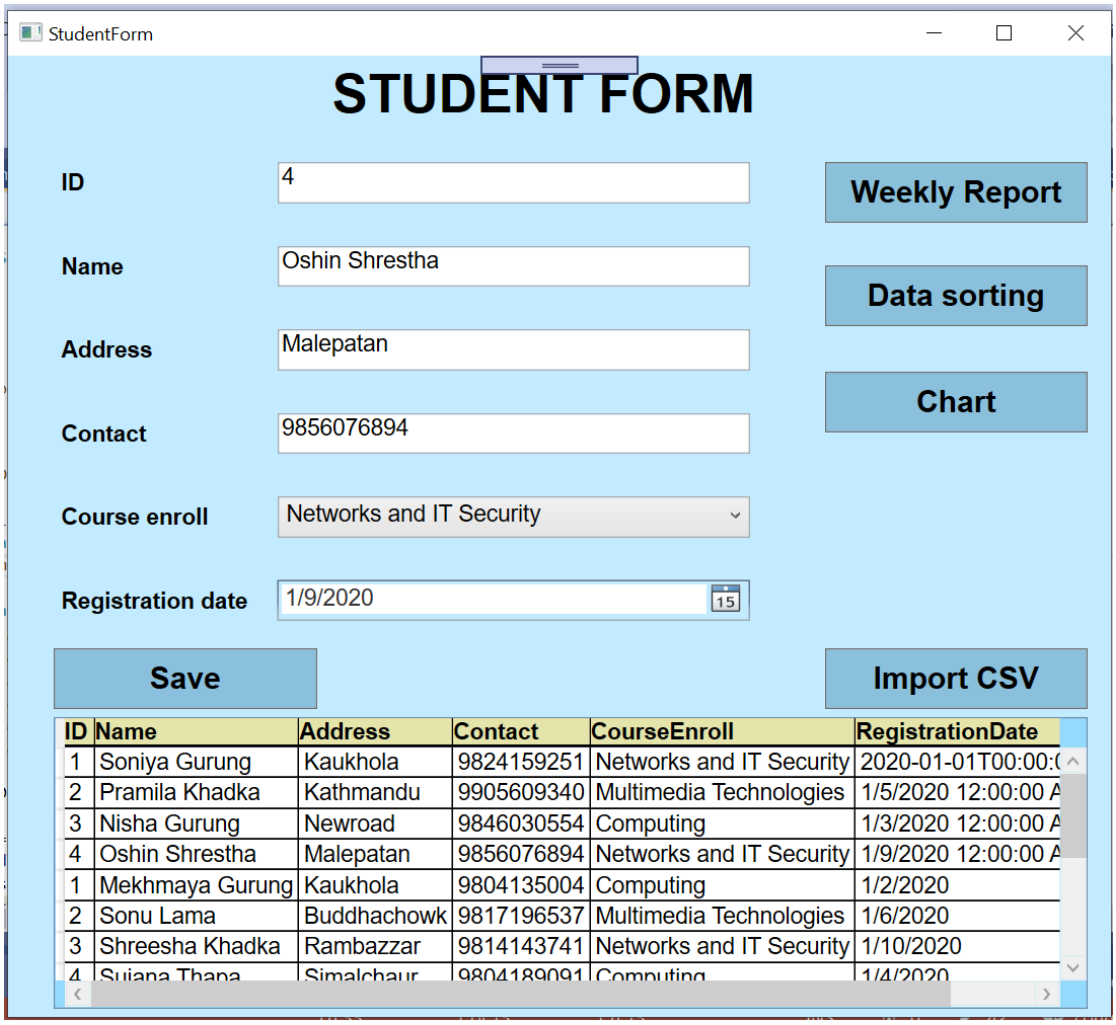



*Figure 6: After importing CSV*

After import CSV, it will show in the grid along with the data we entered from student form.

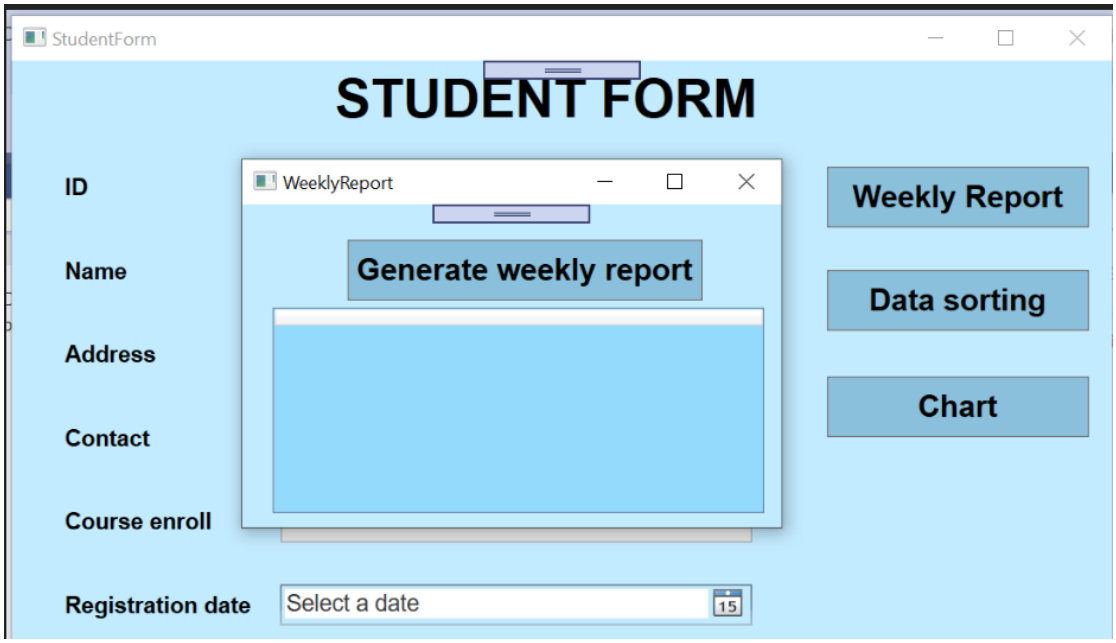



*Figure 7: Weekly report*

We have to create a weekly report showing total number of students enrolled. Here when we click on weekly report button, it will show weekly report generating screen.

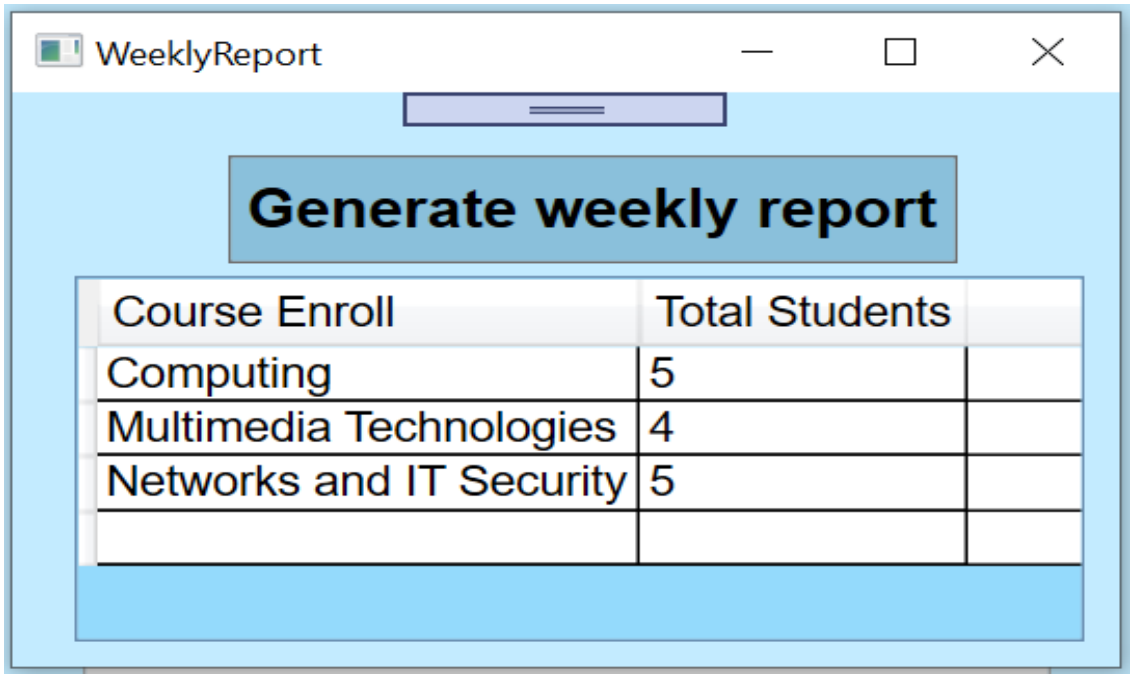



*Figure 8: Showing Weekly report*

When we click on generate weekly report button, it will show the total students enrolled in courses.

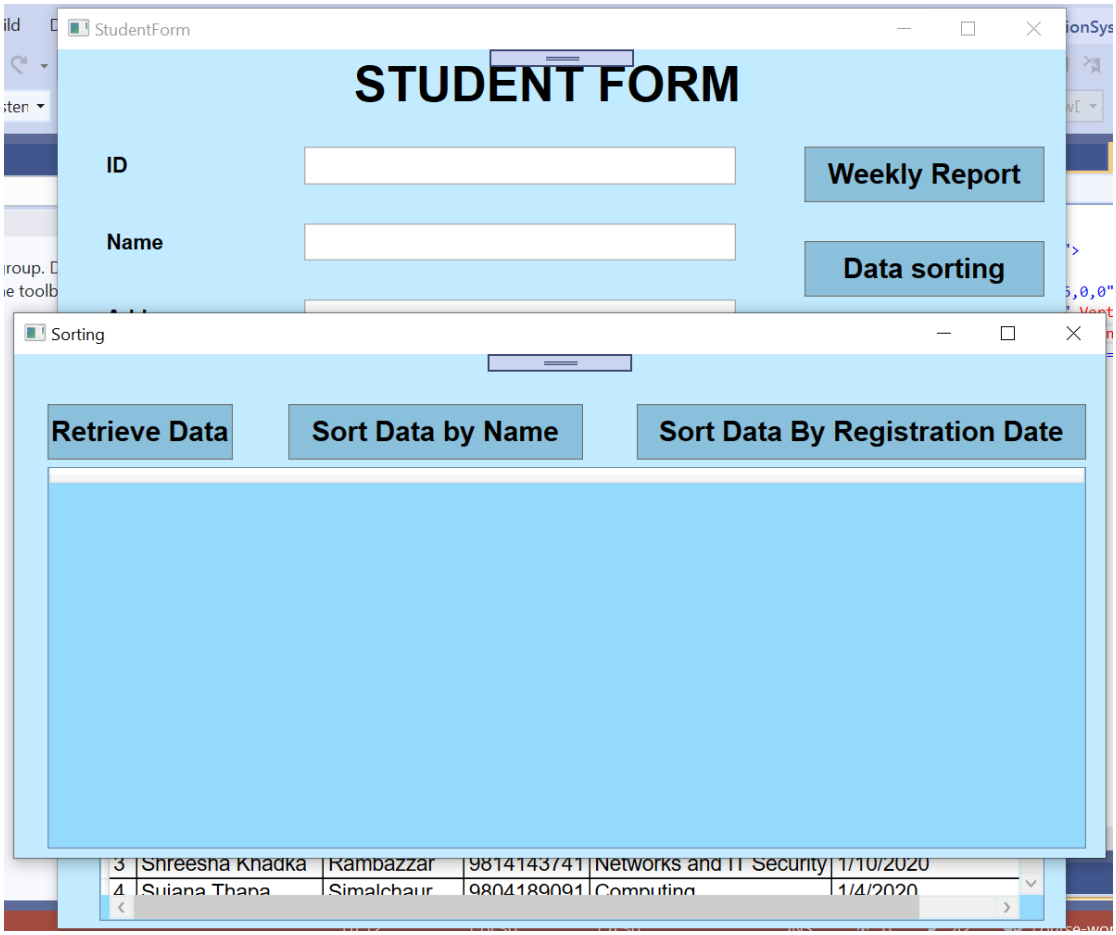
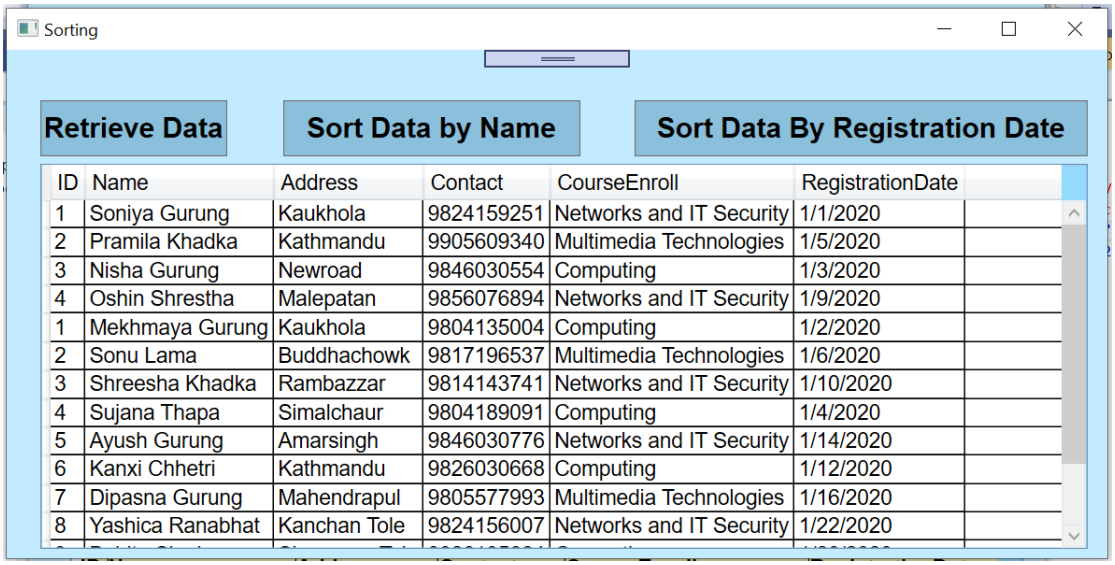



*Figure 9: Data Sorting*

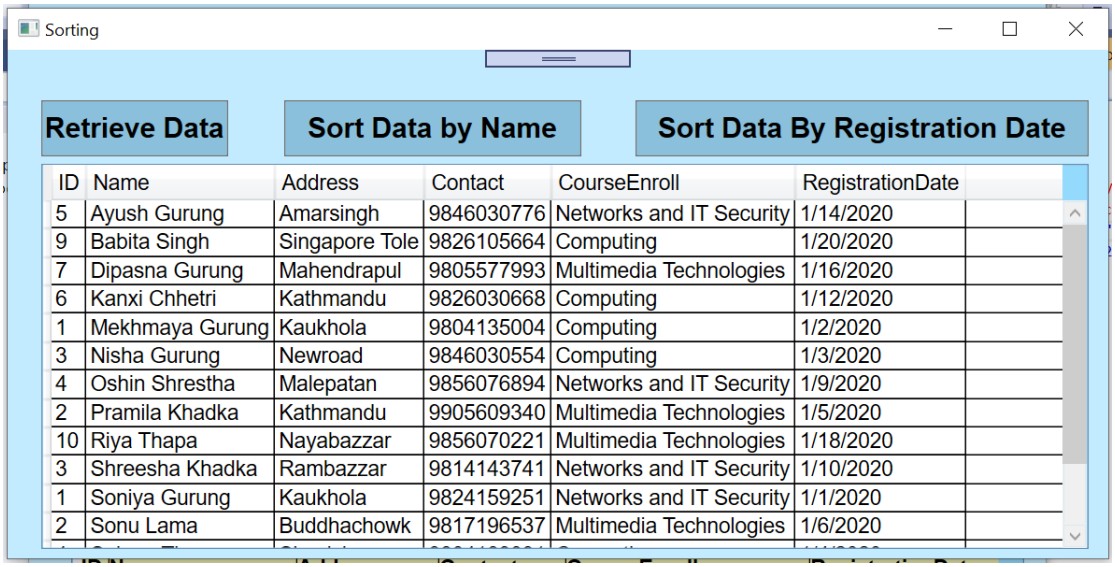
In this project, we have to show the data sort by name and data sort by registration date. So for this process we have to click on data sorting button which will show the page containing retrieve, import data by name and import data by registration date buttons respectively.




| ID | Name | Address | Contact | CourseEnroll | RegistrationDate |
|---|---|---|---|---|---|
| 1 | Soniya Gurung | Kaukhola | 9824159251 | Networks and IT Security | 1/1/2020 |
| 2 | Pramila Khadka | Kathmandu | 9905609340 | Multimedia Technologies | 1/5/2020 |
| 3 | Nisha Gurung | Newroad | 9846030554 | Computing | 1/3/2020 |
| 4 | Oshin Shrestha | Malepatan | 9856076894 | Networks and IT Security | 1/9/2020 |
| 1 | Mekhmaya Gurung | Kaukhola | 9804135004 | Computing | 1/2/2020 |
| 2 | Sonu Lama | Buddhachowk | 9817196537 | Multimedia Technologies | 1/6/2020 |
| 3 | Shreesha Khadka | Rambazzar | 9814143741 | Networks and IT Security | 1/10/2020 |
| 4 | Sujana Thapa | Simalchaur | 9804189091 | Computing | 1/4/2020 |
| 5 | Ayush Gurung | Amarsingh | 9846030776 | Networks and IT Security | 1/14/2020 |
| 6 | Kanxi Chhetri | Kathmandu | 9826030668 | Computing | 1/12/2020 |
| 7 | Dipasna Gurung | Mahendrapul | 9805577993 | Multimedia Technologies | 1/16/2020 |
| 8 | Yashica Ranabhat | Kanchan Tole | 9824156007 | Networks and IT Security | 1/22/2020 |

*Figure 10: Data Retrieve*

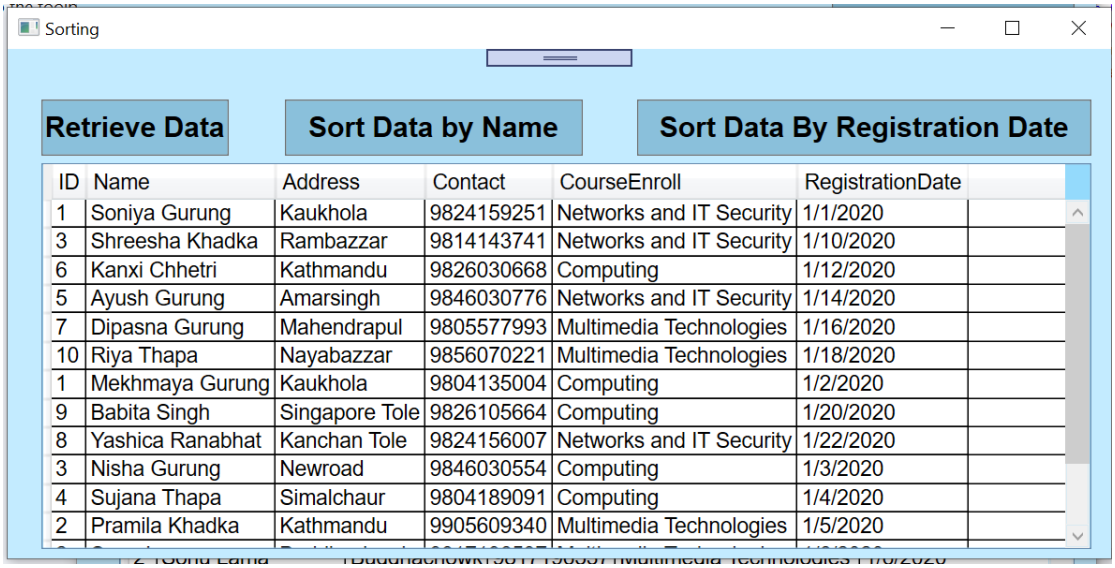
When we click on retrieve button all the data are shown in the grid.




| ID | Name | Address | Contact | CourseEnroll | RegistrationDate |
|---|---|---|---|---|---|
| 5 | Ayush Gurung | Amarsingh | 9846030776 | Networks and IT Security | 1/14/2020 |
| 9 | Babita Singh | Singapore Tole | 9826105664 | Computing | 1/20/2020 |
| 7 | Dipasna Gurung | Mahendrapul | 9805577993 | Multimedia Technologies | 1/16/2020 |
| 6 | Kanxi Chhetri | Kathmandu | 9826030668 | Computing | 1/12/2020 |
| 1 | Mekhmaya Gurung | Kaukhola | 9804135004 | Computing | 1/2/2020 |
| 3 | Nisha Gurung | Newroad | 9846030554 | Computing | 1/3/2020 |
| 4 | Oshin Shrestha | Malepatan | 9856076894 | Networks and IT Security | 1/9/2020 |
| 2 | Pramila Khadka | Kathmandu | 9905609340 | Multimedia Technologies | 1/5/2020 |
| 10 | Riya Thapa | Nayabazzar | 9856070221 | Multimedia Technologies | 1/18/2020 |
| 3 | Shreesha Khadka | Rambazzar | 9814143741 | Networks and IT Security | 1/10/2020 |
| 1 | Soniya Gurung | Kaukhola | 9824159251 | Networks and IT Security | 1/1/2020 |
| 2 | Sonu Lama | Buddhachowk | 9817196537 | Multimedia Technologies | 1/6/2020 |

*Figure 11: Data Sort By Name*

When sort data by name button is clicked, the data will appear in alphabetical order.




| ID | Name | Address | Contact | CourseEnroll | RegistrationDate |
|---|---|---|---|---|---|
| 1 | Soniya Gurung | Kaukhola | 9824159251 | Networks and IT Security | 1/1/2020 |
| 3 | Shreesha Khadka | Rambazzar | 9814143741 | Networks and IT Security | 1/10/2020 |
| 6 | Kanxi Chhetri | Kathmandu | 9826030668 | Computing | 1/12/2020 |
| 5 | Ayush Gurung | Amarsingh | 9846030776 | Networks and IT Security | 1/14/2020 |
| 7 | Dipasna Gurung | Mahendrapul | 9805577993 | Multimedia Technologies | 1/16/2020 |
| 10 | Riya Thapa | Nayabazzar | 9856070221 | Multimedia Technologies | 1/18/2020 |
| 1 | Mekhmaya Gurung | Kaukhola | 9804135004 | Computing | 1/2/2020 |
| 9 | Babita Singh | Singapore Tole | 9826105664 | Computing | 1/20/2020 |
| 8 | Yashica Ranabhat | Kanchan Tole | 9824156007 | Networks and IT Security | 1/22/2020 |
| 3 | Nisha Gurung | Newroad | 9846030554 | Computing | 1/3/2020 |
| 4 | Sujana Thapa | Simalchaur | 9804189091 | Computing | 1/4/2020 |
| 2 | Pramila Khadka | Kathmandu | 9905609340 | Multimedia Technologies | 1/5/2020 |

*Figure 12: Data Sort by Registration date*

When sort data by registration date button is clicked, the data will appear according to the registration date of the students. At last application stop by clicking close button.

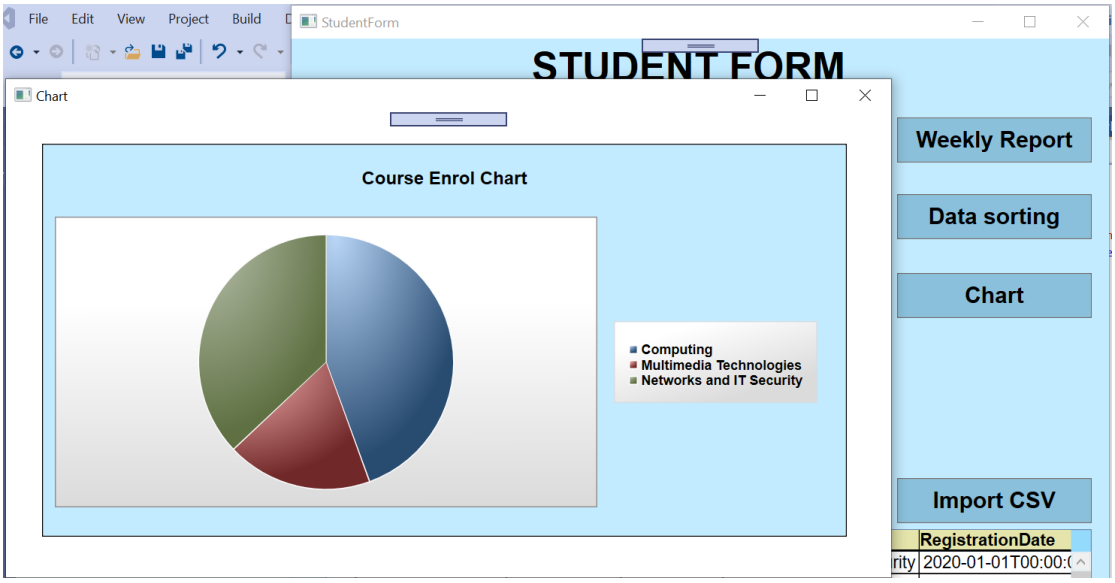



*Figure 13: Course Enrol Chart*

When we click chart button, chart including course enrolment will appear according to the given data.

## 3. System Architecture

### Architecture Diagram

An architecture diagram is a graphical representation of a set of concepts that are part of an architecture including their principles, elements and components (dragon1, 2020).

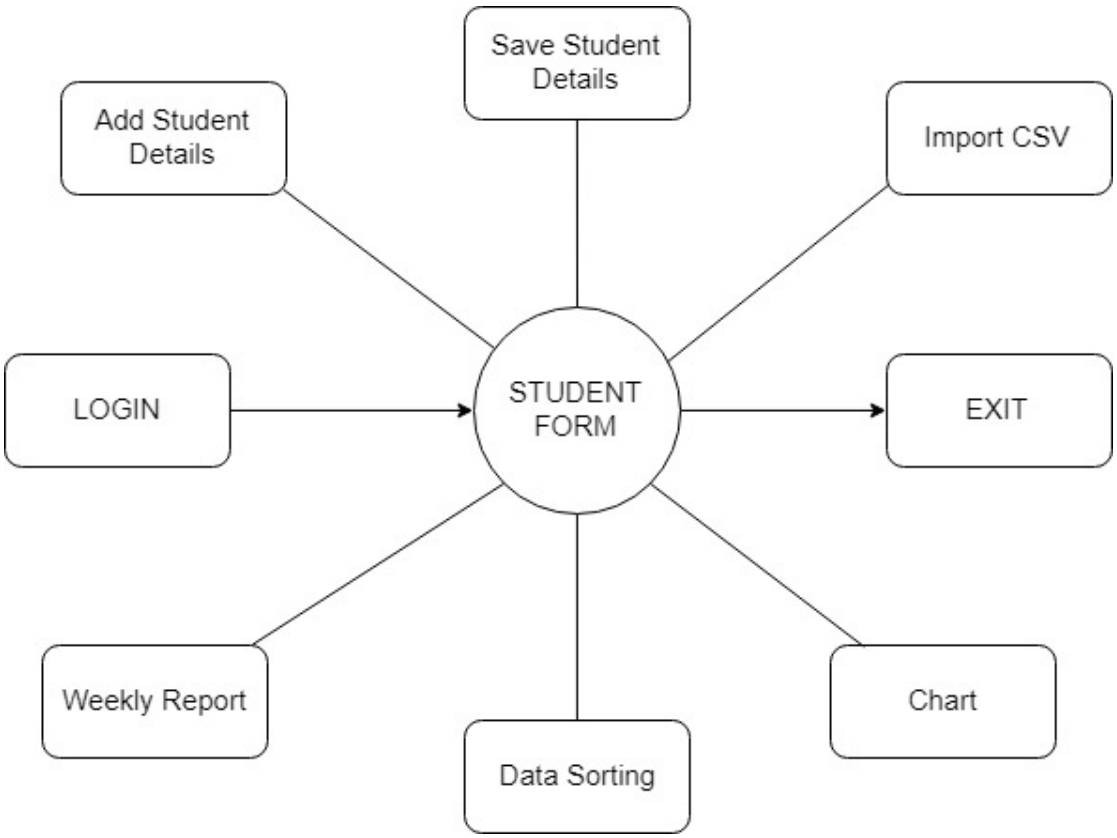



*Figure 14: Architecture Diagram*

Class Diagram

Class diagram is a static diagram. It represents the static view of an application. It is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application (tutorialspoint, 2020).

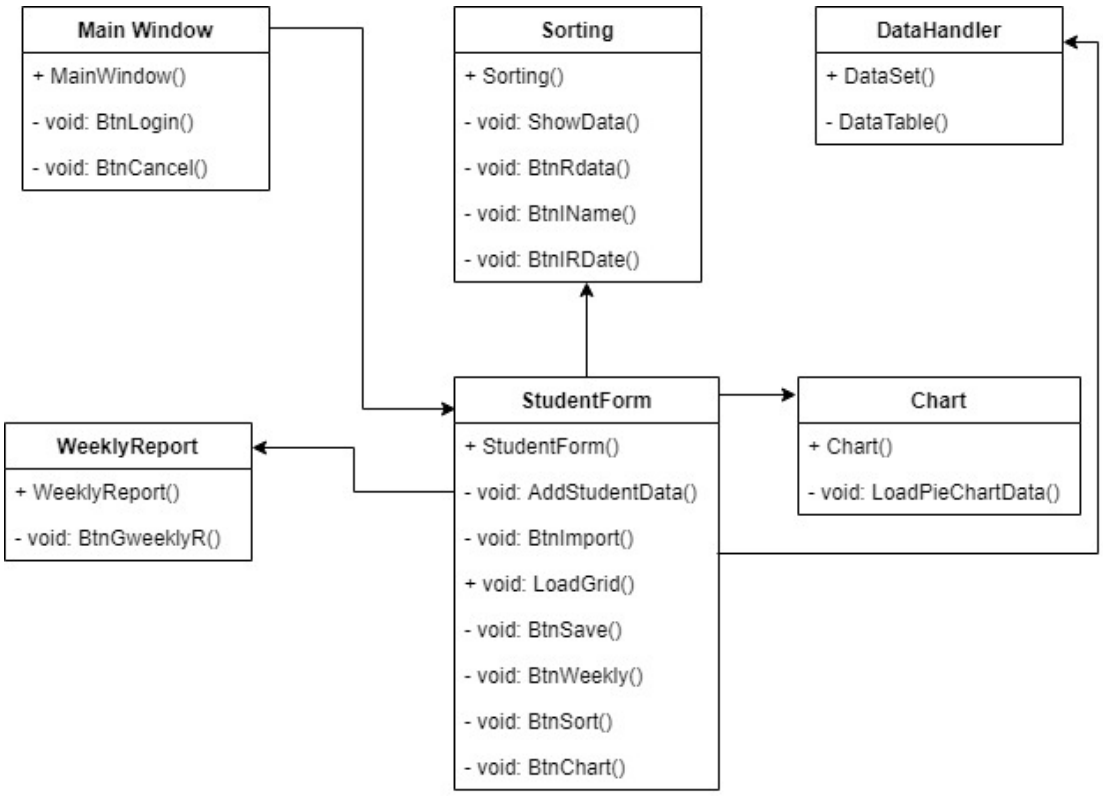



*Figure 15: Class Diagram*

Flowchart

A flowchart is a formalized graphic representation of a logic sequence, work or manufacturing process, organization chart or similar formalized structure. The purpose of a flowchart is to provide people with a common language or reference point when dealing with a project or process (techtarget, 2020).

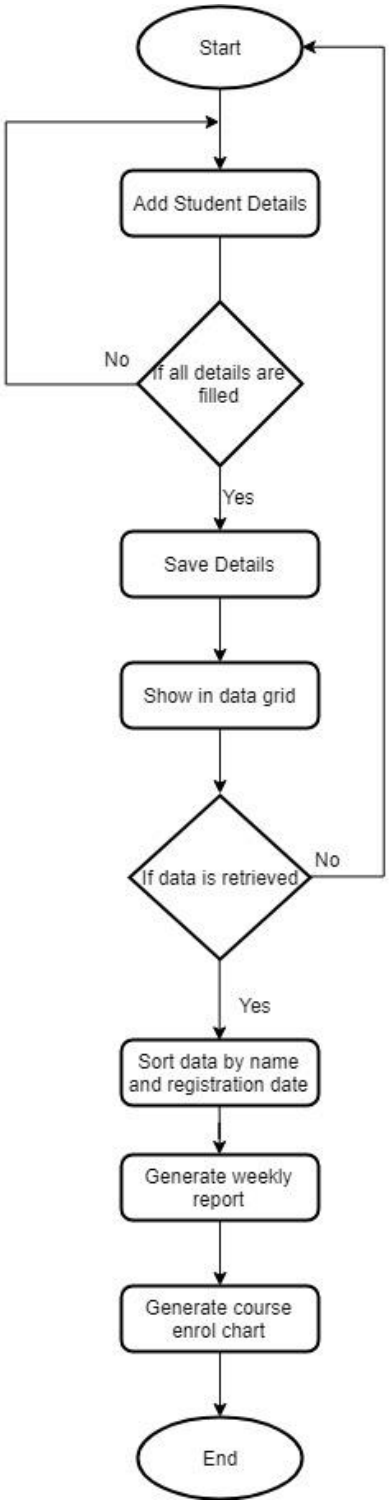



*Figure 16: Flowchart Diagram*

## 4. Algorithm

Bubble sorting algorithm is used in Student Information System application. Bubble sort is a simple algorithm which is sort a given set of elements provided in form of an array with number of elements. Bubble sort compares all the element one by one and sort them based on their values. It is known as bubble sort because with every complete iteration the largest element in the given array, bubbles up towards the last place or the highest index, just like a water bubble rise up to the water surface. Sorting takes places by stepping through all the elements one-by-one and comparing it with the adjacent element and swapping them if required (studytonight, 2020).

Let's consider an array with values {5,3,8,4,6}. Now let's show how pictorial representation of how bubble sort will sort the given array.

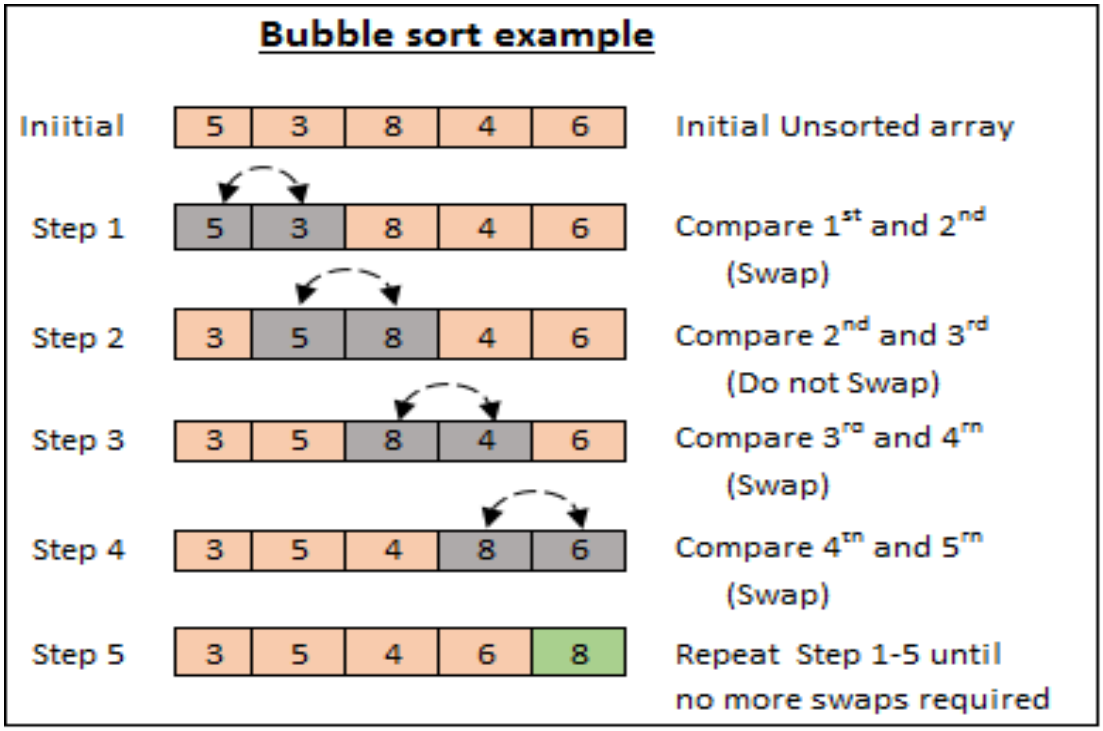



*Figure 17: Bubble Sort Algorithm (medium, 2020)*

## 5. Reflection

After completing the coursework, I became more familiar with using Visual Studio. The coursework is all about creating Student Information System application. I have done many research related to the coursework. Some difficulties ware faced in coding. At last the problem was solved. Such types of application will be helpful in different field like schools, hospitals, offices etc. with lots of effort and hard work I have completed the given coursework within given period of time.

# References

dragon1, 2020. *dragon1.com.* [Online]
Available at: https://www.dragon1.com/terms/architecture-diagram-definition
[Accessed 9 1 2020].

medium, 2020. *medium.* [Online]
Available at: https://medium.com/karuna-sehgal/an-introduction-to-bubble-sort-d85273acfcd8
[Accessed 9 1 2020].

studytonight, 2020. *studytonight.com.* [Online]
Available at: https://www.studytonight.com/data-structures/bubble-sort
[Accessed 9 1 2020].

techtarget, 2020. *techtarget.com.* [Online]
Available at: https://whatis.techtarget.com/definition/flowchart
[Accessed 9 1 2020].

tutorialspoint, 2020. *tutorialspoint.com.* [Online]
Available at: https://www.tutorialspoint.com/uml/uml_class_diagram.htm
[Accessed 9 1 2020].

# 7. Appendix

## MainWindow.xaml

```
<Window x:Class="StudentInformationSystem.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:StudentInformationSystem"
        mc:Ignorable="d"
        Title="Login Page" Height="300" Width="360"
WindowStartupLocation="CenterScreen">
    <Grid Background="#FFC3EBFF">
        <Label Content="Username" HorizontalAlignment="Left" Margin="10,80,0,0"
VerticalAlignment="Top" FontFamily="Arial" FontSize="20"/>
        <Label Content="Password" HorizontalAlignment="Left"
Margin="10,140,0,0" VerticalAlignment="Top" FontFamily="Arial" FontSize="20"/>
        <TextBox x:Name="userName" HorizontalAlignment="Left" Height="33"
Margin="117,80,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="221"
FontFamily="Arial" FontSize="20"/>
        <Button Content="LOGIN" HorizontalAlignment="Left" Margin="46,207,0,0"
VerticalAlignment="Top" Width="112" FontFamily="Arial" FontSize="25"
Click="BtnLogin" Background="#FF8BC0DB" Foreground="Black" Height="40"/>
        <PasswordBox x:Name="userPass" HorizontalAlignment="Left"
Margin="117,140,0,0" VerticalAlignment="Top" FontFamily="Arial" FontSize="20"
Width="221" Height="33"/>
        <Label Content="WELCOME" HorizontalAlignment="Left" Margin="97,10,0,0"
VerticalAlignment="Top" FontFamily="Arial" FontSize="35" FontWeight="Bold"/>
        <Button Content="CANCEL" HorizontalAlignment="Left"
Margin="195,207,0,0" VerticalAlignment="Top" Width="112" Height="40"
FontFamily="Arial" FontSize="25" Click="BtnCancel" Background="#FF8AC0DB"
Foreground="Black"/>

    </Grid>
</Window>
```

## MainWindow.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.Navigation;
using System.Windows.Shapes;

namespace StudentInformationSystem
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
```

```
        {
            InitializeComponent();
        }

        private void BtnLogin(object sender, RoutedEventArgs e)
        {
            if (userName.Text == "admin" && userPass.Password == "admin")
            {
                MessageBox.Show("Login Successful");
                StudentForm sForm = new StudentForm();
                this.Hide();
                sForm.Show();
            }
            else
            {
                MessageBox.Show("Enter correct username and password");
            }
        }

        private void BtnCancel(object sender, RoutedEventArgs e)
        {
            Close();
        }
    }
}
```

## StudentForm.xaml

```
<Window x:Class="StudentInformationSystem.StudentForm"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:StudentInformationSystem"
        mc:Ignorable="d"
        Title="StudentForm" Height="670" Width="740"
WindowStartupLocation="CenterScreen">
    <Grid Background="#FFC3EBFF">
        <Label Content="ID" HorizontalAlignment="Left" VerticalAlignment="Top"
Margin="30,70,0,0" FontFamily="Arial" FontSize="15" FontWeight="Bold"/>
        <Label Content="Name" HorizontalAlignment="Left" Margin="30,125,0,0"
VerticalAlignment="Top" FontFamily="Arial" FontSize="15" FontWeight="Bold"/>
        <Label Content="Address" HorizontalAlignment="Left" Margin="30,180,0,0"
VerticalAlignment="Top" FontFamily="Arial" FontSize="15" FontWeight="Bold"/>
        <Label Content="Contact" HorizontalAlignment="Left" Margin="30,235,0,0"
VerticalAlignment="Top" FontFamily="Arial" FontSize="15" FontWeight="Bold"/>
        <Label Content="Course enroll" HorizontalAlignment="Left"
Margin="30,290,0,0" VerticalAlignment="Top" FontFamily="Arial" FontSize="15"
FontWeight="Bold"/>
        <Label Content="Registration date" HorizontalAlignment="Left"
Margin="30,345,0,0" VerticalAlignment="Top" FontFamily="Arial" FontSize="15"
FontWeight="Bold"/>
        <StackPanel HorizontalAlignment="Left" Height="49"
VerticalAlignment="Top" Width="733" RenderTransformOrigin="0.5,0.5">
            <Label Content="&#x9;       STUDENT FORM" FontFamily="Arial"
FontSize="35" FontWeight="Bold" Margin="0,0,9.667,0"/>
        </StackPanel>
        <TextBox x:Name="txtID" HorizontalAlignment="Left" Height="27"
Margin="177,70,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="311"
FontFamily="Arial" FontSize="15" />
```

```
        <TextBox x:Name="txtContact" HorizontalAlignment="Left" Height="27"
Margin="177,235,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="311"
FontFamily="Arial" FontSize="15"/>
        <DatePicker x:Name="rDate" HorizontalAlignment="Left"
Margin="177,345,0,0" VerticalAlignment="Top" FontFamily="Arial" FontSize="15"
Width="311" Height="27"/>
        <ComboBox x:Name="cEnroll" HorizontalAlignment="Left"
Margin="177,290,0,0" VerticalAlignment="Top" Width="311" FontFamily="Arial"
FontSize="15" Height="27">
            <ComboBoxItem Content="Networks and IT Security"/>
            <ComboBoxItem Content="Multimedia Technologies"/>
            <ComboBoxItem Content="Computing"/>
        </ComboBox>
        <TextBox x:Name="txtAddress" HorizontalAlignment="Left" Height="27"
Margin="177,180,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="311"
FontFamily="Arial" FontSize="15"/>
        <TextBox x:Name="txtName" HorizontalAlignment="Left" Height="27"
Margin="177,125,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="311"
FontFamily="Arial" FontSize="15"/>
        <Button x:Name="saveBtn" Content="Save" HorizontalAlignment="Left"
Margin="30,390,0,0" VerticalAlignment="Top" Width="173" FontFamily="Arial"
FontSize="20" Click="BtnSave" Height="40" FontWeight="Bold"
Background="#FF8AC0DD"/>
        <DataGrid x:Name="DataGridXAML"  HorizontalAlignment="Left"
Height="192" VerticalAlignment="Top" Width="680" Margin="30,435,0,0"
FontSize="15" FontFamily="Arial" Background="#FF94DAFC">
            <DataGrid.Resources>
                <Style TargetType="{x:Type DataGridColumnHeader}">
                    <Setter Property="Background" Value="#FFE4E4AB"/>
                    <Setter Property="FontWeight" Value="SemiBold"/>
                    <Setter Property="BorderThickness" Value="0 0 1 2"/>
                    <Setter Property="BorderBrush" Value="Black"/>
                </Style>
            </DataGrid.Resources>

        </DataGrid>
        <Button Content="Data sorting" HorizontalAlignment="Left"
Margin="537,138,0,0" VerticalAlignment="Top" Width="173" FontFamily="Arial"
FontSize="20" Click="BtnSort" Height="40" FontWeight="Bold"
Background="#FF8AC0DB" />
        <Button Content="Import CSV" HorizontalAlignment="Left"
Margin="537,390,0,0" VerticalAlignment="Top" Width="173" FontFamily="Arial"
FontSize="20" Click="BtnImport" Height="40" FontWeight="Bold"
Background="#FF8AC0DB"/>
        <Button x:Name="btnWeekly" Content="Weekly Report"
HorizontalAlignment="Left" Margin="537,70,0,0" VerticalAlignment="Top"
Width="173" FontFamily="Arial" FontSize="20" Click="BtnWeekly" Height="40"
FontWeight="Bold" Background="#FF8BC0DB"/>
        <Button Content="Chart" HorizontalAlignment="Left" Margin="537,208,0,0"
VerticalAlignment="Top" Width="173" FontFamily="Arial" FontSize="20"
Height="40" FontWeight="Bold" Background="#FF8AC0DB"/>

    </Grid>
</Window>
```

## StudentForm.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;
using System.Data;
using System.IO;
using Microsoft.Win32;

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

            LoadGrid();
        }

        private void AddStudentData(DataSet dataSet)
        {
            var newRow = dataSet.Tables["Student"].NewRow();
            newRow["ID"] = txtID.Text;
            newRow["Name"] = txtName.Text;
            newRow["Address"] = txtAddress.Text;
            newRow["Contact"] = txtContact.Text;
            newRow["CourseEnroll"] = cEnroll.Text;
            newRow["RegistrationDate"] = rDate.SelectedDate;
            dataSet.Tables["Student"].Rows.Add(newRow);
        }
        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 RegistrationDate { get; set; }
        }

        private void BtnImport(object sender, RoutedEventArgs e)
        {
            var dataSet = new DataSet();
            dataSet.ReadXml(@"C:\XML\student.xml");
            OpenFileDialog openFileDialog = new OpenFileDialog();
            if (openFileDialog.ShowDialog() == true)
            {
                string filePath = openFileDialog.FileName;
                //read all std from file code copy

                using (var reader = new StreamReader(filePath))
                {
```

```
                    reader.ReadLine();
                    while (!reader.EndOfStream)
                    {
                        var line = reader.ReadLine();
                        var values = line.Split(',');
                        var newRow = dataSet.Tables["Student"].NewRow();
                        newRow["ID"] = values[0];
                        newRow["Name"] = values[1];
                        newRow["Address"] = values[2];
                        newRow["Contact"] = values[3];
                        newRow["CourseEnroll"] = values[4];
                        newRow["RegistrationDate"] = values[5];
                        dataSet.Tables["Student"].Rows.Add(newRow);

                        dataSet.WriteXml(@"C:\XML\student.xml");
                    }
                }
                DataGridXAML.ItemsSource =
dataSet.Tables["Student"].DefaultView;
            }
        }

        public void LoadGrid()
        {
            var dataSet = new DataSet();
            if (File.Exists(@"C:\XML\student.xml"))
            {
                dataSet.ReadXml(@"C:\XML\student.xml");
                DataGridXAML.ItemsSource =
dataSet.Tables["Student"].DefaultView;
            }
        }

        private void BtnSave(object sender, RoutedEventArgs e)
        {
            if (txtID.Text == "" || txtName.Text == "" || txtAddress.Text == ""
|| txtContact.Text == "" ||
                txtContact.Text == "" || cEnroll.Text == "" || rDate.Text ==
"")
            {
                MessageBox.Show("Fill all students details.");
            }
            else
            {
                try
                {
                    var handler = new DataHandler();
                    var dataSet = new DataSet();

                    if (File.Exists(@"C:\XML\student.xml"))
                    {
                        dataSet.ReadXml(@"C:\XML\student.xml");

                    }
                    else
                    {
                        dataSet = handler.CreateDataSet();
                    }

                    AddStudentData(dataSet);
                    dataSet.WriteXml(@"C:\XML\student.xml");
                    LoadGrid();
```

```
                    MessageBox.Show("Successfully Added!");
                }
                catch (Exception)
                {

                }

                }
            }

        private void BtnWeekly(object sender, RoutedEventArgs e)
        {

            WeeklyReport weeklyR = new WeeklyReport();
            weeklyR.ShowDialog();
        }

        private void BtnSort(object sender, RoutedEventArgs e)
        {
            Sorting sorting = new Sorting();
            sorting.ShowDialog();
        }

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

## WeeklyReport.xaml

```
<Window x:Class="StudentInformationSystem.WeeklyReport"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:StudentInformationSystem"
        mc:Ignorable="d"
        Title="WeeklyReport" Height="250" Width="370"
WindowStartupLocation="CenterScreen">
    <Grid Background="#FFC3EBFF">
        <DataGrid x:Name="WeeklyDataGrid" HorizontalAlignment="Left"
Height="135" Margin="20,68,0,0" VerticalAlignment="Top" Width="324"
Background="#FF94DAFC" FontFamily="Arial" FontSize="15"/>
        <Button x:Name="btnGweeklyR" Content="Generate weekly report"
HorizontalAlignment="Left" Margin="69,23,0,0" VerticalAlignment="Top"
Width="234" FontFamily="Arial" FontSize="20" Click="BtnGweeklyR"
Background="#FF8AC0DB" FontWeight="Bold" Height="40"/>

    </Grid>
</Window>
```

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

        private void BtnGweeklyR(object sender, RoutedEventArgs e)
        {
            var dataset = new DataSet(); //new data set declared
            dataset.ReadXml(@"C:\XML\student.xml");  //main report read
            DataTable stdReport = dataset.Tables[0];
            int CompTotal = 0;   //initial value of course assign
            int MultiTotal = 0;
            int NetTotal = 0;

            DataTable dTable = new DataTable("DataT");
            dTable.Columns.Add("Course Enroll", typeof(String));  //two columns
created
            dTable.Columns.Add("Total Students", typeof(int));

            for (int i = 0; i < stdReport.Rows.Count; i++)
            {
                String col = stdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    CompTotal++;
                }
                else if (col == "Multimedia Technologies")
                {
                    MultiTotal++;
                }
                else if (col == "Networks and IT Security")
                {
                    NetTotal++;
                }
            }

            dTable.Rows.Add("Computing", CompTotal); // final assign
            dTable.Rows.Add("Multimedia Technologies", MultiTotal);
```

```
            dTable.Rows.Add("Networks and IT Security", NetTotal);

            WeeklyDataGrid.ItemsSource = dTable.DefaultView; // weekly data
grid
        }
    }
}
```

## Sorting.xaml

```
<Window x:Class="StudentInformationSystem.Sorting"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:StudentInformationSystem"
        mc:Ignorable="d"
        Title="Sorting" Height="400" Width="800"
WindowStartupLocation="CenterScreen">
    <Grid Background="#FFC3EBFF">
        <Button Content="Sort Data by Name" HorizontalAlignment="Left"
Margin="197,36,0,0" VerticalAlignment="Top" Width="212" FontFamily="Arial"
FontSize="20" Click="BtnIName" FontWeight="Bold" Height="40"
Background="#FF8AC0DB"/>
        <Button Content="Retrieve Data" HorizontalAlignment="Left"
Margin="24,36,0,0" VerticalAlignment="Top" Width="133" FontFamily="Arial"
FontSize="20" Click="BtnRdata" FontWeight="Bold" Height="40"
Background="#FF8AC0DB"/>
        <Button Content="Sort Data By Registration Date"
HorizontalAlignment="Left" Margin="448,36,0,0" VerticalAlignment="Top"
Width="323" FontFamily="Arial" FontSize="20" Click="BtnIRDate"
FontWeight="Bold" Height="40" Background="#FF8AC0DB"/>
        <DataGrid x:Name="DataGridSorting" HorizontalAlignment="Left"
Height="275" Margin="24,81,0,0" VerticalAlignment="Top" Width="747"
FontFamily="Arial" FontSize="15" Background="#FF94DAFC"/>

    </Grid>
</Window>
```

## Sorting.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 StudentInformationSystem
{
    /// <summary>
    /// Interaction logic for Sorting.xaml
```

```
    /// </summary>
    public partial class Sorting : Window
    {
        DataTable dataT;
        public Sorting()
        {
            InitializeComponent();
        }

        private void ShowData()
        {
            string dataXMLFile = @"C:\XML\student.xml";
            System.Data.DataSet dataSet = new DataSet();
            dataSet.ReadXml(dataXMLFile);

            dataT = new DataTable("dt");
            dataT.Columns.Add("ID", typeof(string));
            dataT.Columns.Add("Name", typeof(string));
            dataT.Columns.Add("Address", typeof(string));
            dataT.Columns.Add("Contact", typeof(string));
            dataT.Columns.Add("CourseEnroll", typeof(string));
            dataT.Columns.Add("RegistrationDate", typeof(string));

            for (int i = 0; i < dataSet.Tables[0].Rows.Count; i++)
            {
                string s = dataSet.Tables[0].Rows[i][5].ToString();
                DateTime dTime = DateTime.Parse(s);
                dataT.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(),
                    dTime.ToShortDateString());
            }
            DataView dataView = new DataView(dataT);
            DataGridSorting.ItemsSource = dataView;
        }

        private void BtnRdata(object sender, RoutedEventArgs e)
        {
            ShowData();
        }

        private void BtnIName(object sender, RoutedEventArgs e)
        {
            DataView dataView = new DataView(dataT);
            dataView.Sort = "Name";
            DataGridSorting.ItemsSource = dataView;
        }

        private void BtnIRDate(object sender, RoutedEventArgs e)
        {
            DataView dataView = new DataView(dataT);
            dataView.Sort = "RegistrationDate";
            DataGridSorting.ItemsSource = dataView;
        }
    }
}
```

## DataHandler.cs

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

namespace StudentInformationSystem
{
    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("CourseEnroll", typeof(string));
            dt.Columns.Add("RegistrationDate", typeof(DateTime));
            return dt;
        }
    }
}
```

## Chart.xaml

```
<Window x:Class="StudentInformationSystem.Chart"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:DV="clr-
namespace:System.Windows.Controls.DataVisualization;assembly=System.Windows.Con
trols.DataVisualization.Toolkit"
        xmlns:DVC="clr-
namespace:System.Windows.Controls.DataVisualization.Charting;assembly=System.Wi
ndows.Controls.DataVisualization.Toolkit"

        xmlns:local="clr-namespace:StudentInformationSystem"

        Title="Chart" Height="450" Width="800">
    <Grid>
        <DVC:Chart Margin="32,28,39.333,35.667" Title="Course Enrol Chart"
Background="#FFC3EBFF" FontWeight="Bold" FontFamily="Arial">
            <DVC:PieSeries   x:Name="chart" IndependentValueBinding="{Binding
Path=Key}"  DependentValueBinding="{Binding Path=Value}">
            </DVC:PieSeries>
        </DVC:Chart>

    </Grid>
</Window>
```

## Chart.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 StudentInformationSystem
{
    /// <summary>
    /// Interaction logic for Chart.xaml
    /// </summary>
    public partial class Chart : Window
    {
        public Chart()
        {
            InitializeComponent();
            LoadPieChartData();
        }
        private void LoadPieChartData()
        {
            var dataset = new DataSet(); //new data set declared
            dataset.ReadXml(@"C:\XML\student.xml");  //main report read
            DataTable stdReport = dataset.Tables[0];
            int CompTotal = 0;   //initial value of course assign
            int MultiTotal = 0;
            int NetTotal = 0;

            DataTable dTable = new DataTable("DataT");
            dTable.Columns.Add("Course Enroll", typeof(String));  //two columns
created
            dTable.Columns.Add("Total Students", typeof(int));

            for (int i = 0; i < stdReport.Rows.Count; i++)
            {
                String col = stdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    CompTotal++;
                }
                else if (col == "Multimedia Technologies")
                {
                    MultiTotal++;
                }
                else if (col == "Networks and IT Security")
                {
                    NetTotal++;
                }
            }
```

```
            dTable.Rows.Add("Computing", CompTotal); // final assign
            dTable.Rows.Add("Multimedia Technologies", MultiTotal);
            dTable.Rows.Add("Networks and IT Security", NetTotal);

((System.Windows.Controls.DataVisualization.Charting.PieSeries)chart).ItemsSour
ce =
                new KeyValuePair<string, int>[]{
        new KeyValuePair<string,int>("Computing", 12),
        new KeyValuePair<string,int>("Multimedia Technologies", 5),
        new KeyValuePair<string,int>("Networks and IT Security", 10) };

        }
    }
}
```

student.xml

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Student>
    <ID>1</ID>
    <Name>Soniya Gurung</Name>
    <Address>Kaukhola</Address>
    <Contact>9824159251</Contact>
    <CourseEnroll>Networks and IT Security</CourseEnroll>
    <RegistrationDate>2020-01-01T00:00:00+05:45</RegistrationDate>
  </Student>
  <Student>
    <ID>2</ID>
    <Name>Pramila Khadka</Name>
    <Address>Kathmandu</Address>
    <Contact>9905609340</Contact>
    <CourseEnroll>Multimedia Technologies</CourseEnroll>
    <RegistrationDate>1/5/2020 12:00:00 AM</RegistrationDate>
  </Student>
  <Student>
    <ID>3</ID>
    <Name>Nisha Gurung</Name>
    <Address>Newroad</Address>
```

```
  <Contact>9846030554</Contact>
  <CourseEnroll>Computing</CourseEnroll>
  <RegistrationDate>1/3/2020 12:00:00 AM</RegistrationDate>
 </Student>
 <Student>
  <ID>4</ID>
  <Name>Oshin Shrestha</Name>
  <Address>Malepatan</Address>
  <Contact>9856076894</Contact>
  <CourseEnroll>Networks and IT Security</CourseEnroll>
  <RegistrationDate>1/9/2020 12:00:00 AM</RegistrationDate>
 </Student>
 <Student>
  <ID>1</ID>
  <Name>Mekhmaya Gurung</Name>
  <Address>Kaukhola</Address>
  <Contact>9804135004</Contact>
  <CourseEnroll>Computing</CourseEnroll>
  <RegistrationDate>1/2/2020</RegistrationDate>
 </Student>
 <Student>
  <ID>2</ID>
  <Name>Sonu Lama</Name>
  <Address>Buddhachowk</Address>
  <Contact>9817196537</Contact>
  <CourseEnroll>Multimedia Technologies</CourseEnroll>
  <RegistrationDate>1/6/2020</RegistrationDate>
 </Student>
 <Student>
  <ID>3</ID>
```

```
  <Name>Shreesha Khadka</Name>
  <Address>Rambazzar</Address>
  <Contact>9814143741</Contact>
  <CourseEnroll>Networks and IT Security</CourseEnroll>
  <RegistrationDate>1/10/2020</RegistrationDate>
</Student>
<Student>
  <ID>4</ID>
  <Name>Sujana Thapa</Name>
  <Address>Simalchaur</Address>
  <Contact>9804189091</Contact>
  <CourseEnroll>Computing</CourseEnroll>
  <RegistrationDate>1/4/2020</RegistrationDate>
</Student>
<Student>
  <ID>5</ID>
  <Name>Ayush Gurung</Name>
  <Address>Amarsingh</Address>
  <Contact>9846030776</Contact>
  <CourseEnroll>Networks and IT Security</CourseEnroll>
  <RegistrationDate>1/14/2020</RegistrationDate>
</Student>
<Student>
  <ID>6</ID>
  <Name>Kanxi Chhetri</Name>
  <Address>Kathmandu</Address>
  <Contact>9826030668</Contact>
  <CourseEnroll>Computing</CourseEnroll>
  <RegistrationDate>1/12/2020</RegistrationDate>
</Student>
```

```
<Student>
  <ID>7</ID>
  <Name>Dipasna Gurung</Name>
  <Address>Mahendrapul</Address>
  <Contact>9805577993</Contact>
  <CourseEnroll>Multimedia Technologies</CourseEnroll>
  <RegistrationDate>1/16/2020</RegistrationDate>
</Student>
<Student>
  <ID>8</ID>
  <Name>Yashica Ranabhat</Name>
  <Address>Kanchan Tole</Address>
  <Contact>9824156007</Contact>
  <CourseEnroll>Networks and IT Security</CourseEnroll>
  <RegistrationDate>1/22/2020</RegistrationDate>
</Student>
<Student>
  <ID>9</ID>
  <Name>Babita Singh</Name>
  <Address>Singapore Tole</Address>
  <Contact>9826105664</Contact>
  <CourseEnroll>Computing</CourseEnroll>
  <RegistrationDate>1/20/2020</RegistrationDate>
</Student>
<Student>
  <ID>10</ID>
  <Name>Riya Thapa</Name>
  <Address>Nayabazzar</Address>
  <Contact>9856070221</Contact>
  <CourseEnroll>Multimedia Technologies</CourseEnroll>
```

```
    <RegistrationDate>1/18/2020</RegistrationDate>
  </Student>
</NewDataSet>
```