CS6004NP & Application Development

Assessment Weightage & Type

30% Individual Coursework

3rd Year and 1st Semester

Name: Sonu Lama

College ID: 17030748

University ID: NP04CP4A170037

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

Table of Figure

1. Introduction

This is an individual coursework for the module “Application Development” for Student Management System, which is developed using Visual Studio Platform using c#. the designed system is all about the student management system which is based on a desktop application. This application has the function to allow the user input the student personal detail including registration date so that a system can generate a weekly enrolment report of the student. This system is developed in order to keep record or track of the student’s details, course enrol and registration date. This application allow the user to import an external record from a text file i.e. in .CSV format for bulk data input as well as allow user to input data manually in a text box of student id, name address, contact number, course enrol and registration date. After adding all the data in the datagrid, datasorting is done according to the students first name and another sorting according to the registration date. Course enrol details are added in the datagrid which then used for displaying the weekly tabular report showing the total number of students enrolled so far in each course offered by the institution. Data retrieve is also done for showing the student details. Appropriate datatypes

2. User Manual

Some screenshots are taken as evidence while running the program as shown below:

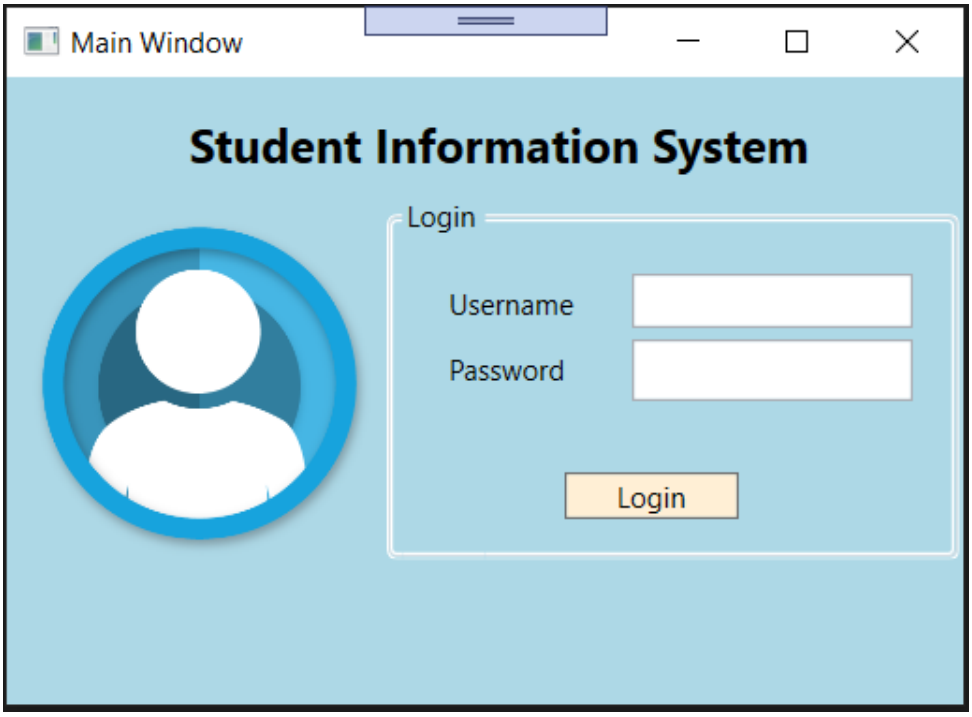


Figure 1: Login Page.

This is the very first window page where Login is needed. The username and password for this system is admin and root respectively. Another page will open after clicking login button if username and password match.

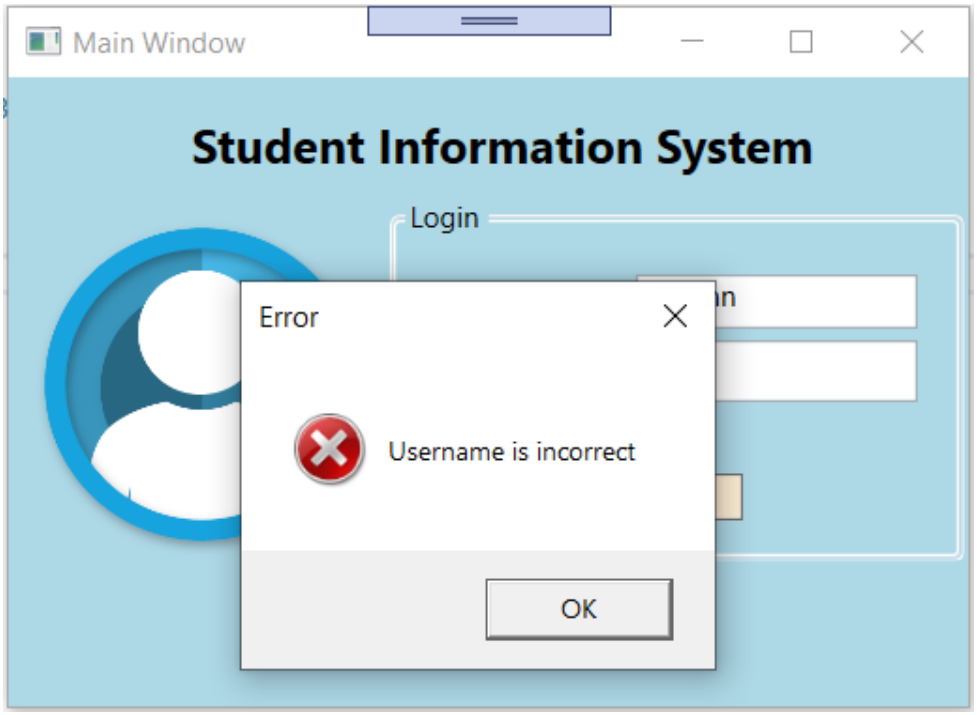


Figure 2: Incorrect Username.

Display an error message if input username is incorrect.

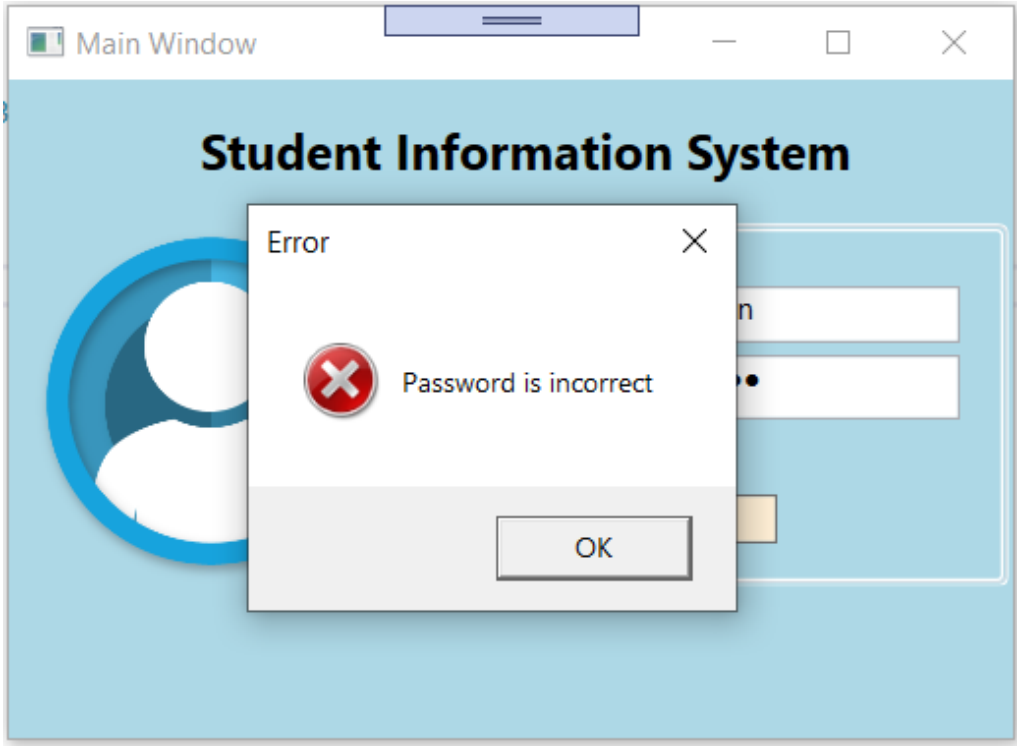


Figure 3: Incorrect Password.

Display an error message if input password is incorrect.

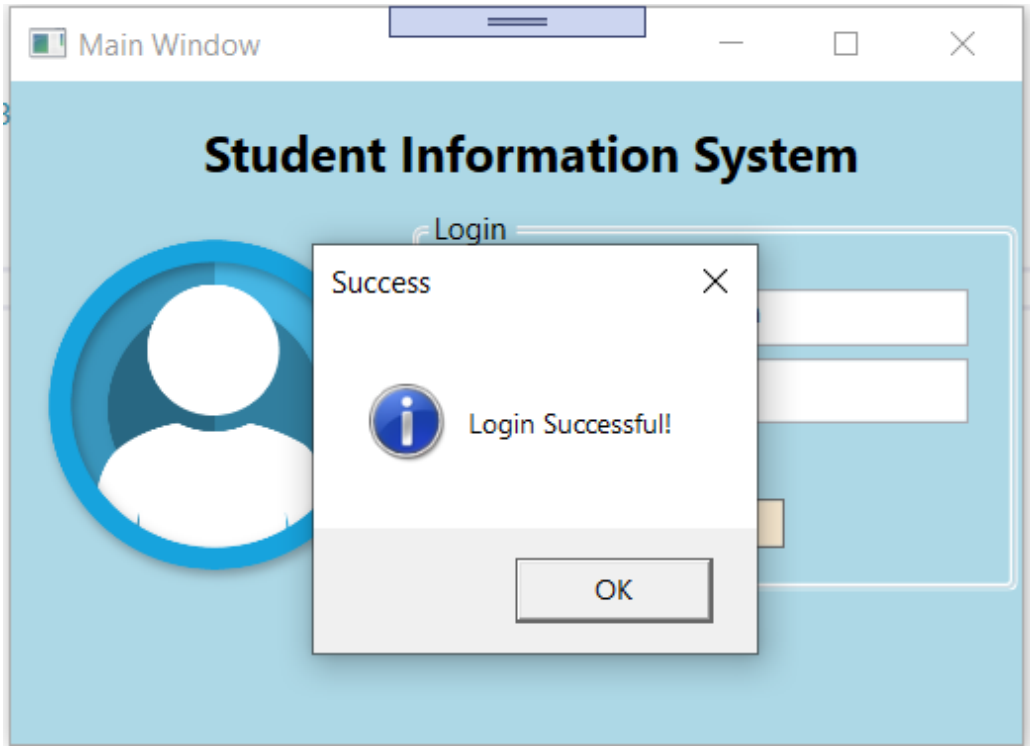


Figure 4: Login Successful.

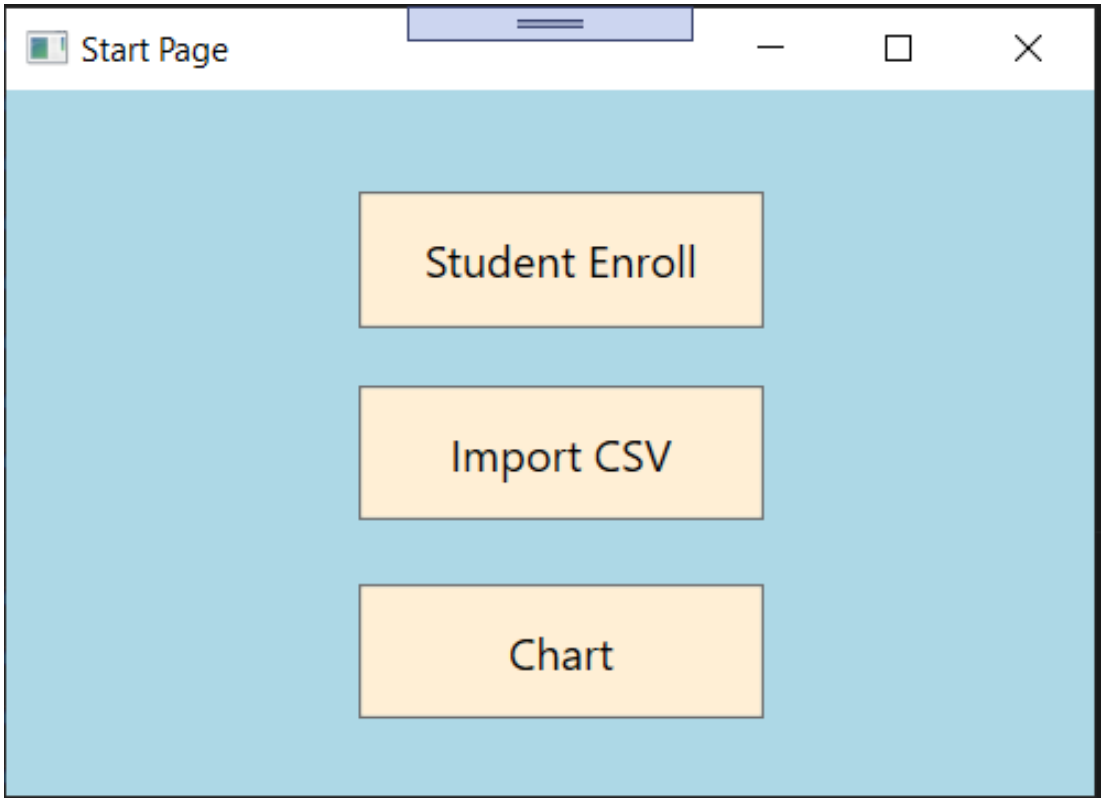


Figure 5: Choose Option.

After logging into the system, user will have the options to choose to click for. These buttons perform different tasks as per their functions.

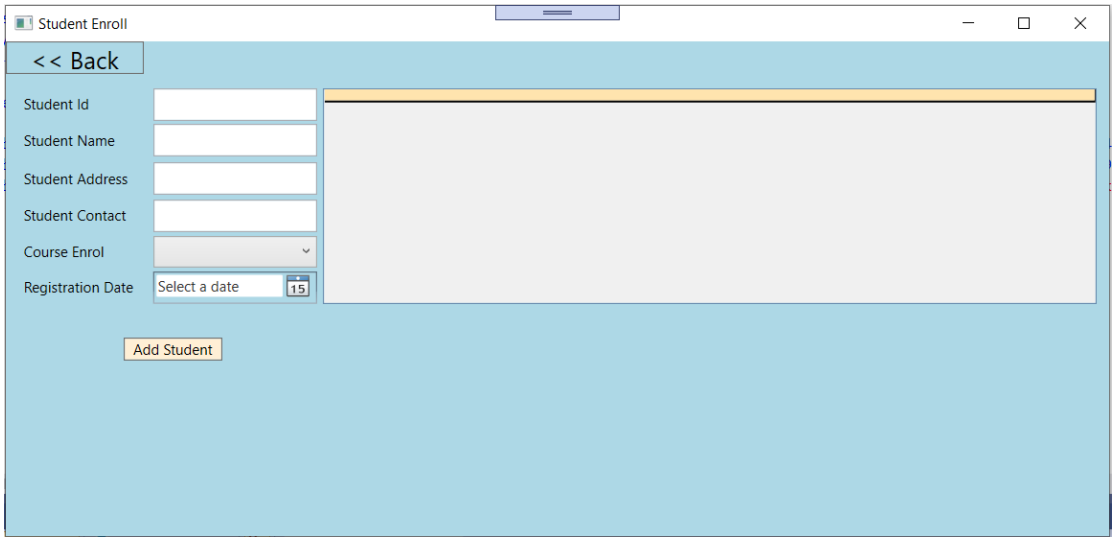


Figure 6: Student Enrol Page.

After clicking the Student Enrol button, the system let us to go to Student Enrol page where student details are added here.

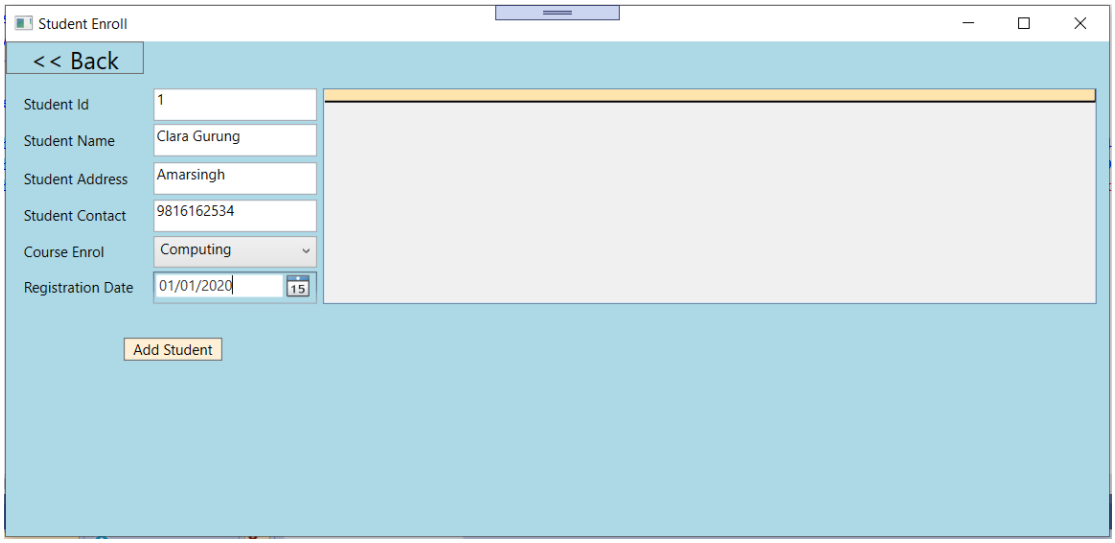


Figure 7: Student Details.

This window include the field where the details of all the students were added and save them in xml file.

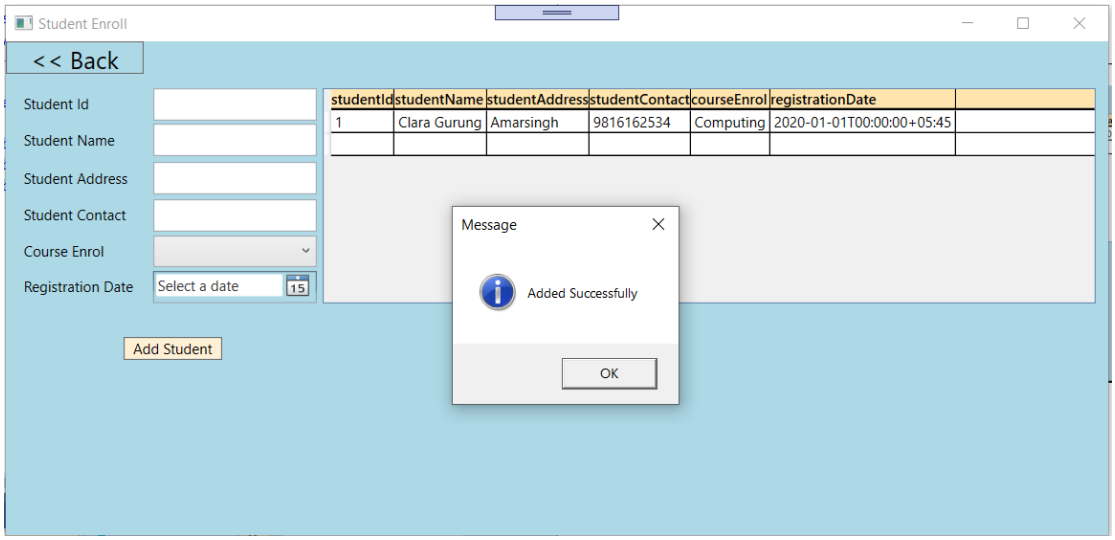


Figure 8: Successful message on adding student data.

A dialog box appears showing a success message where all the data are added in the datagrid.

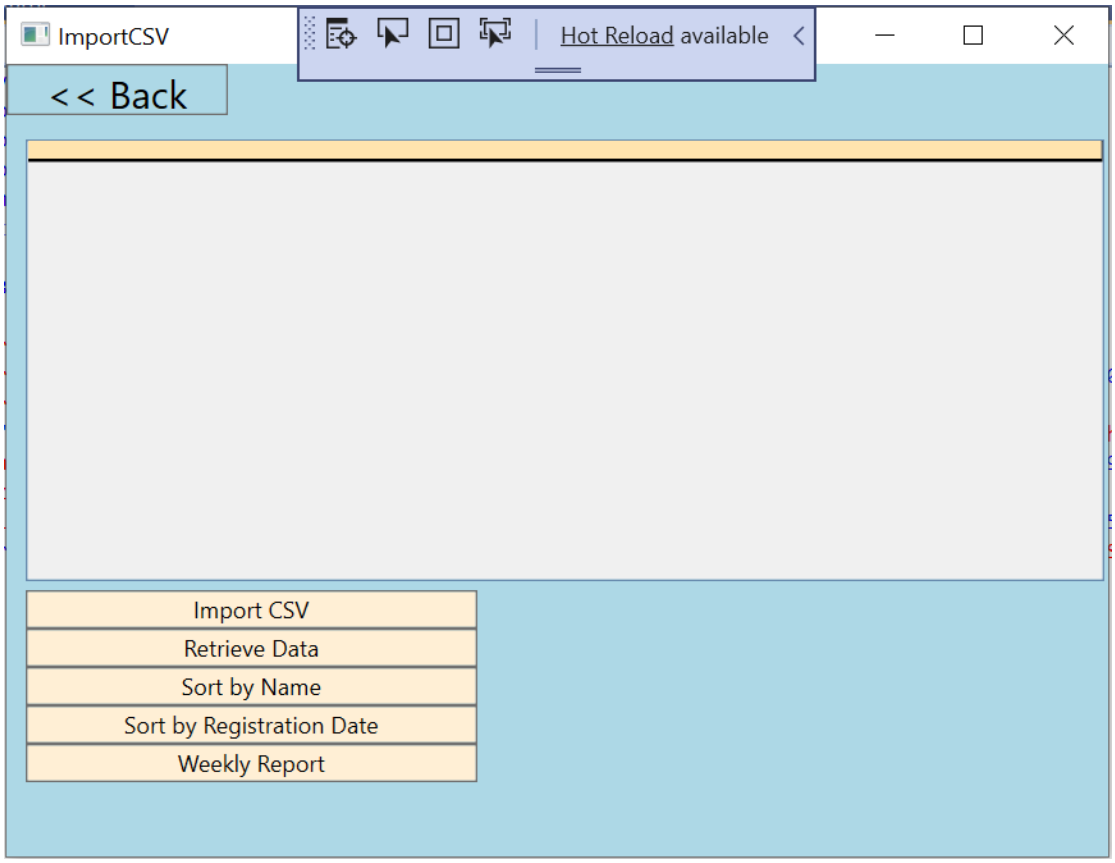


Figure 9: Import CSV page.

This is the GUI for importing, retrieving, sorting and weekly report showing the data in the datagrid.

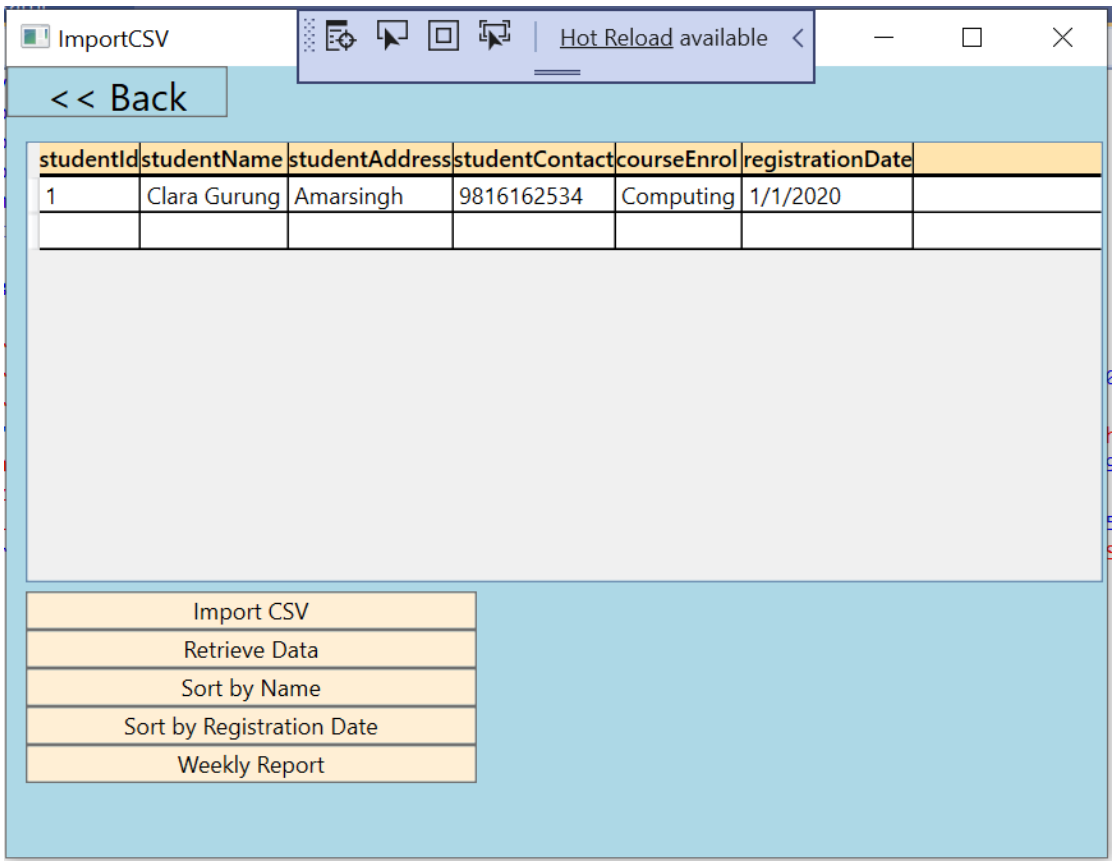


Figure 10: Data Retrieve.

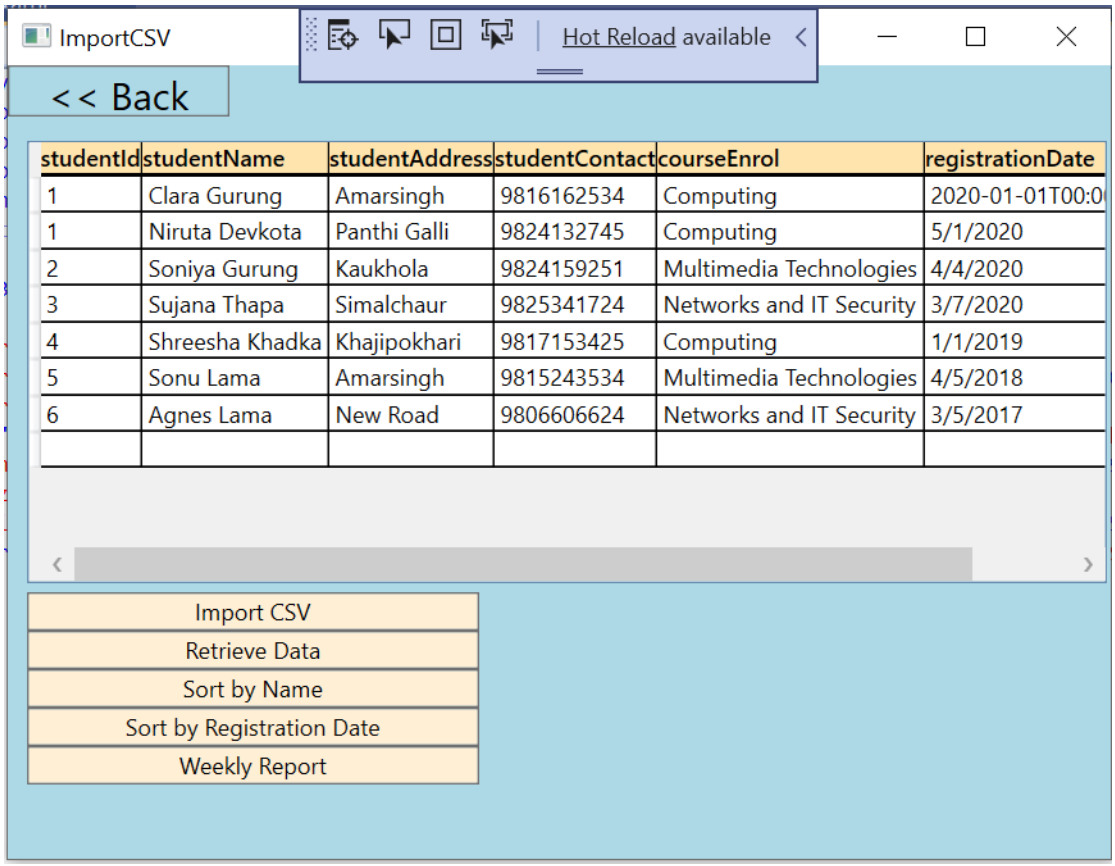


Figure 11: Importing external CSV file.

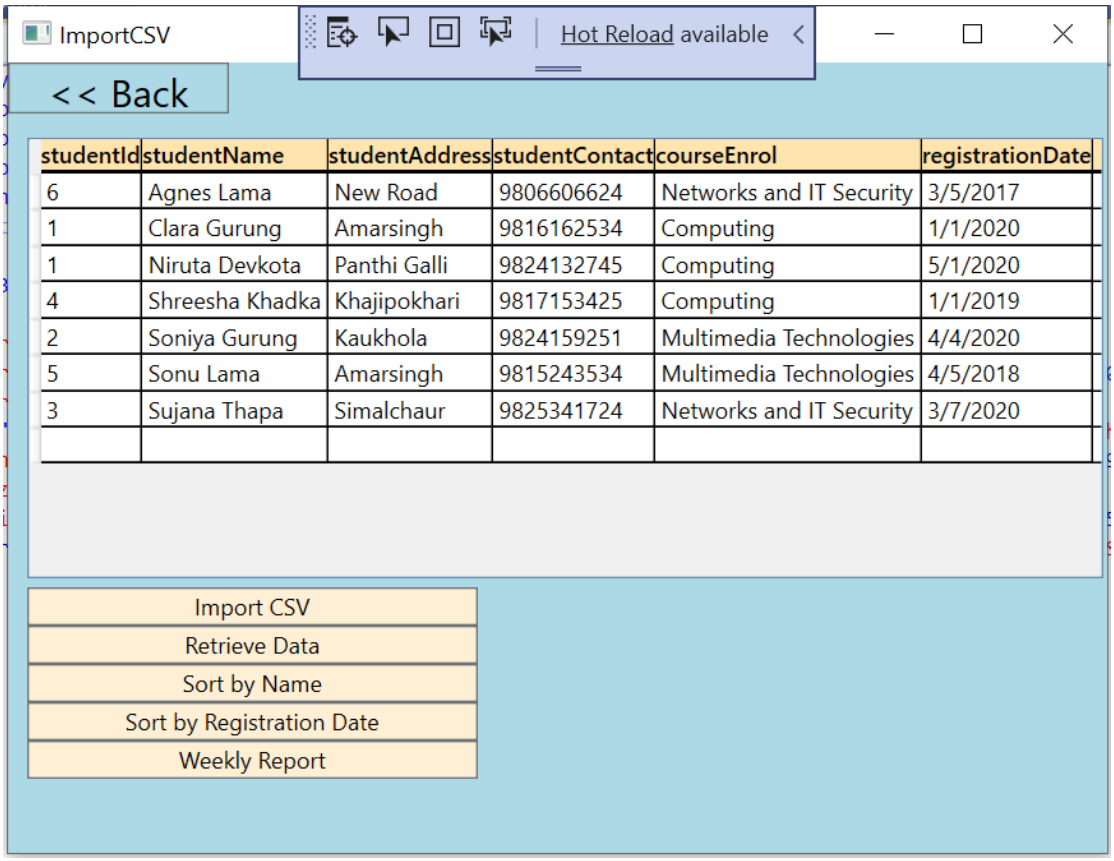


Figure 12: Data Sorting by Name.

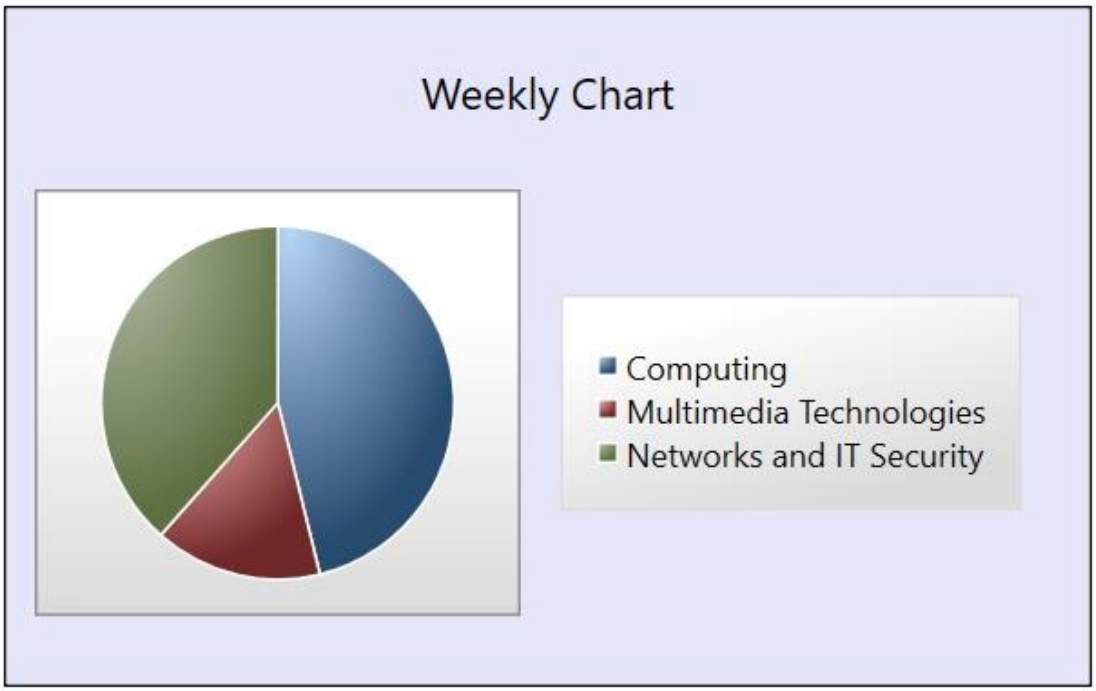


Figure 13: Weekly Chart.

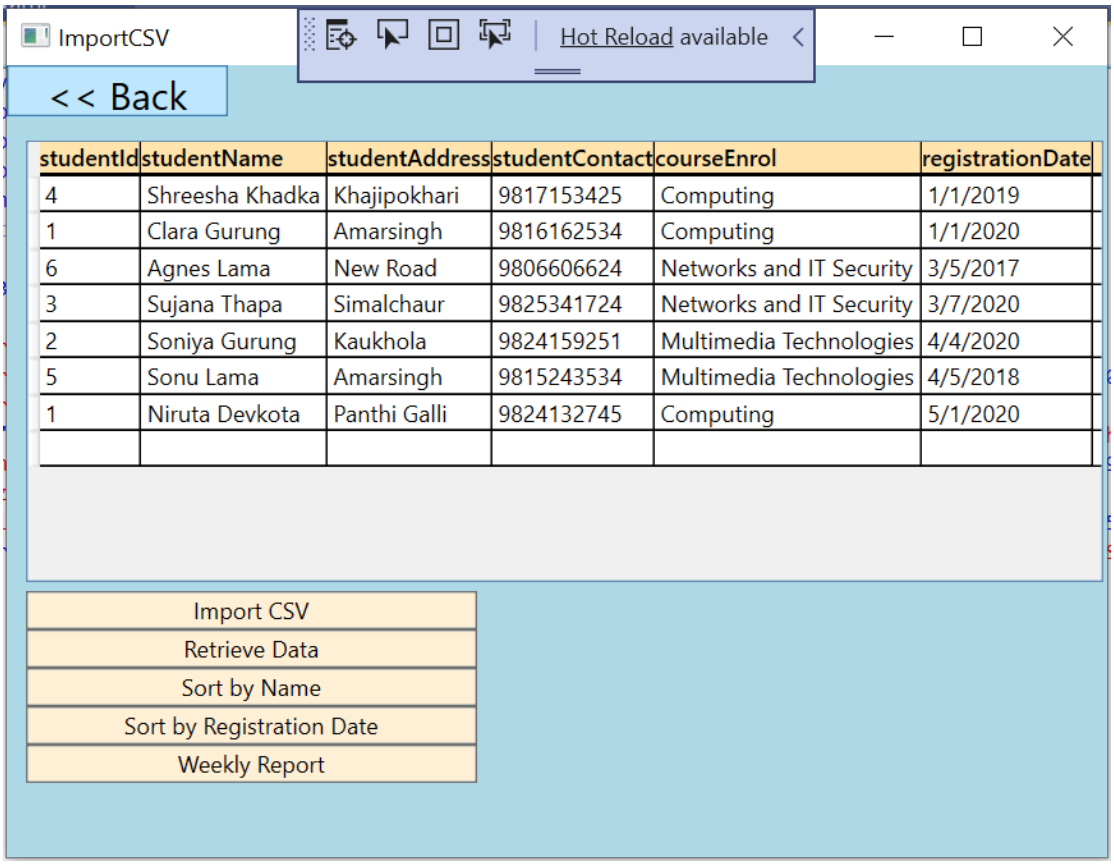


Figure 14: Data sorting by Registration Date.

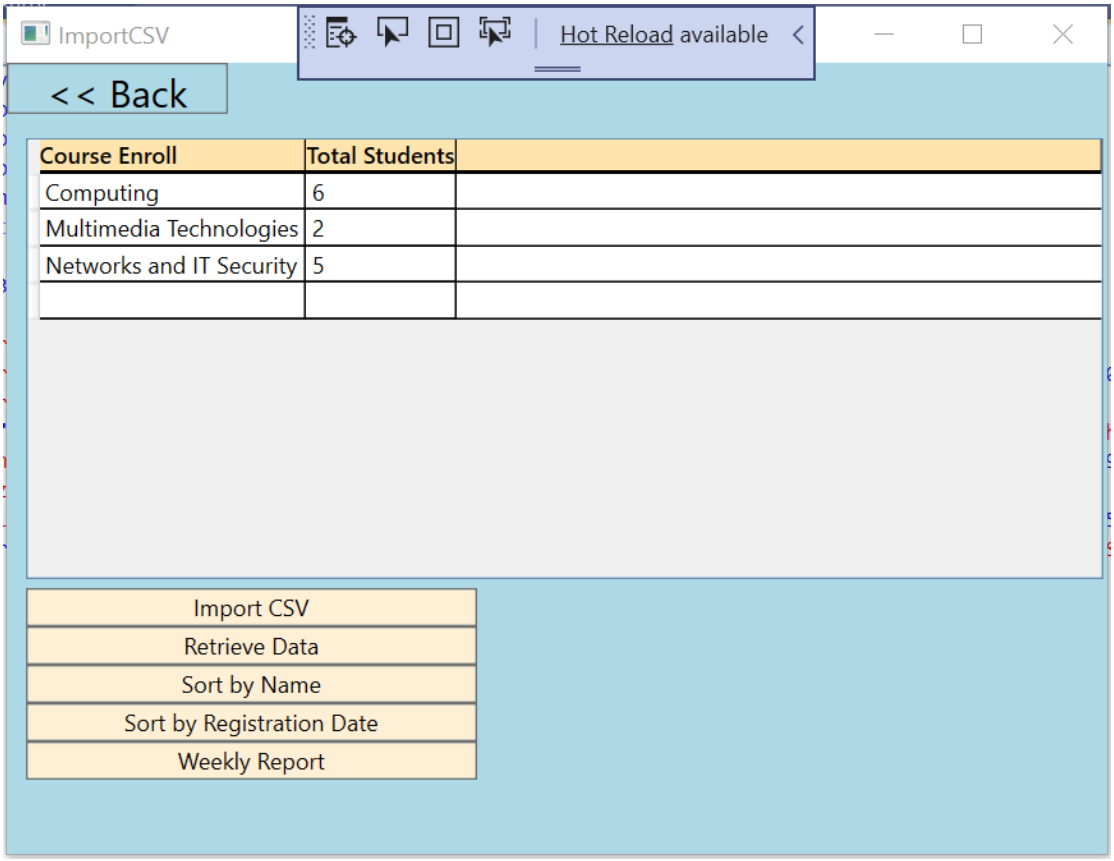


Figure 15: Weekly report table.

3. System Architecture

3.1 Architecture Diagram

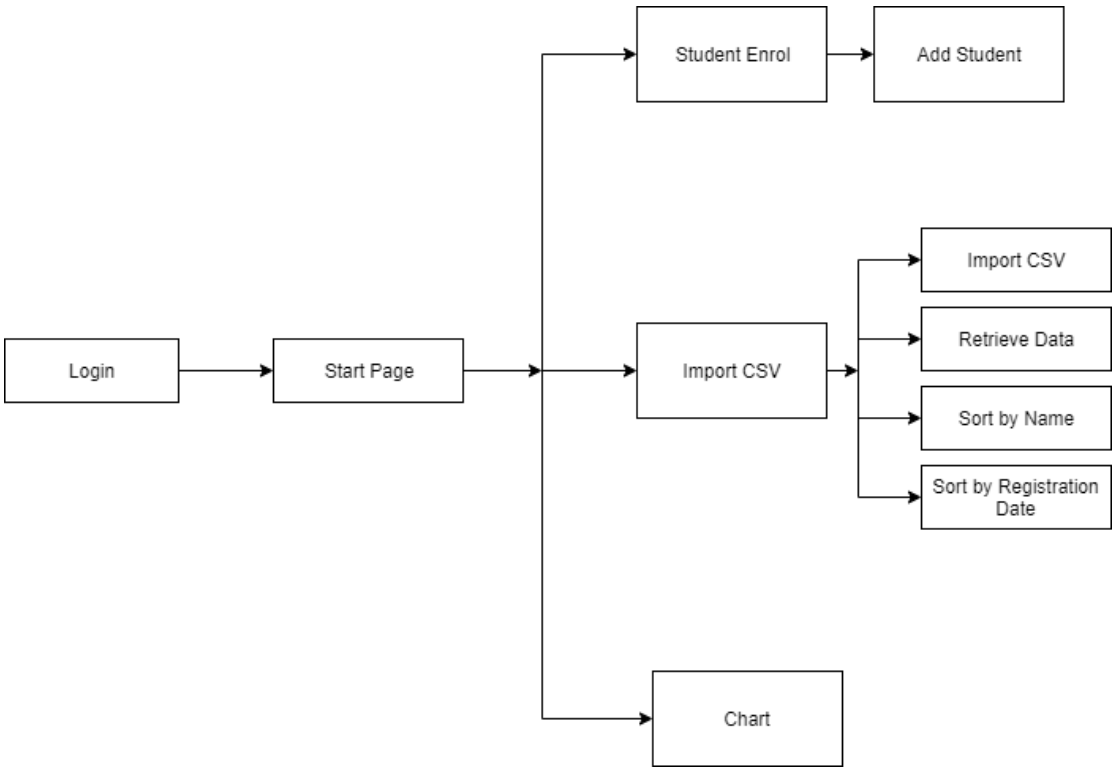


Figure 16: Architecture Diagram.

3.2 Flowchart

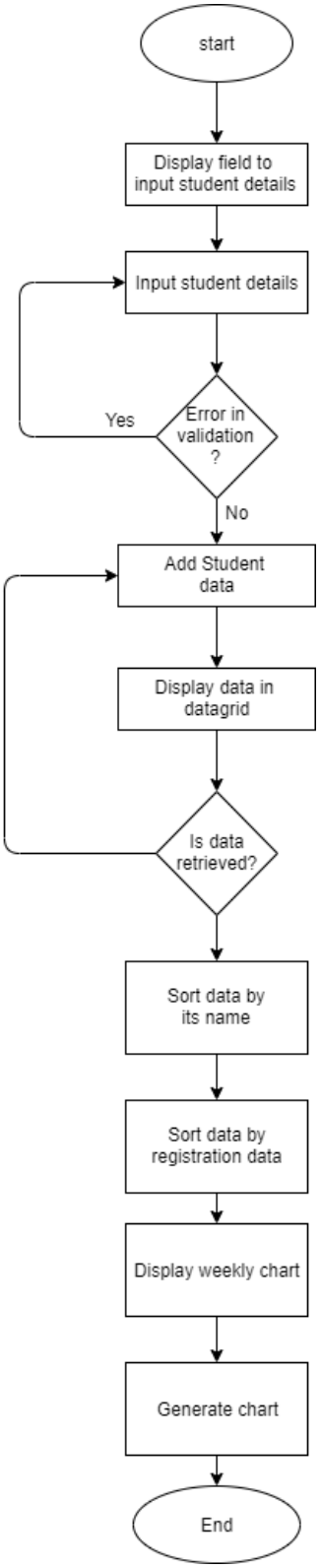


Figure 17: Flowchart

3.3 Class Diagram

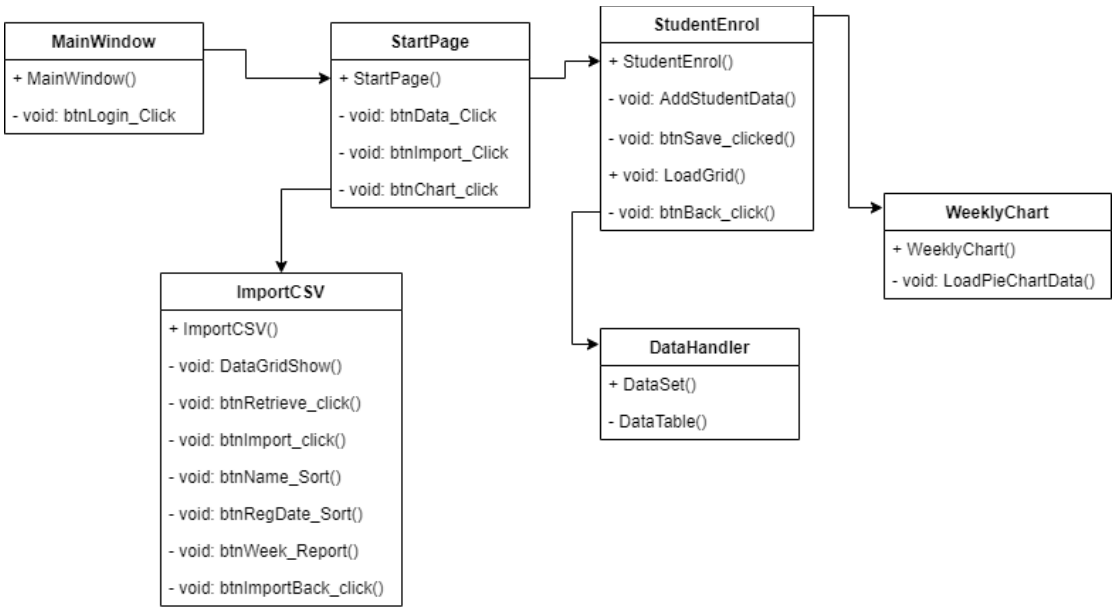


Figure 18: Class Diagram.

4. Algorithm

4.1 Bubble sort

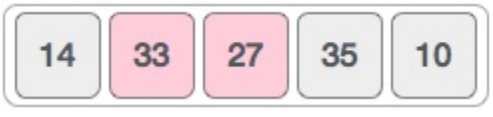
Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of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

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.

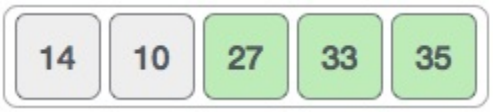
Then we move to the next two values, 35 and 10.

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

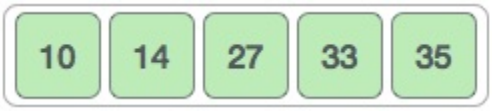
We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this −

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.



Now we should look into some practical aspects of bubble sort.

(tutorialspoint, 2019)

5. Reflection

Creating the framework in Microsoft Visual Studios 2019 keeping C# as essential programming dialect isn't modern involvement for me. But creating in C# environment is modern for me. Creating a record keeping framework for student information system is truly a extreme assignment though. Serialization and deserialization are another modern thing while creating the framework. In spite of the fact that, making unused classes and strategies makes a difference to pace the improvement assignment. Bringing in and sending out of CSV record is additionally a unused assignment and it truly offer assistance me in picking up information of record dealing with. Making a lesson chart inside the visual studio makes a difference me in documentation stage. With the developing of innovation, the visual studio and its community makes a difference newbie designer like us to pace our improvement speed.

References

tutorialspoint, 2019. *www.tutorialspoint.com.* [Online] Available at: https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm
[Accessed 5 01 2019].

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="Main Window" Height="305.728" Width="423.457"
Background="LightBlue" WindowStartupLocation="CenterScreen">
    <Grid>
        <Label Content="Username" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="184,84,0,0"/>
        <TextBox x:Name="txtUser" HorizontalAlignment="Left" Height="23"
TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="267,84,0,0"/>
        <Label Content="Password" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="184,112,0,0" Width="64"/>
        <PasswordBox x:Name="txtPass" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="120" Margin="267,112,0,0" Height="26"/>
        <Button x:Name="btnLogin" Content="Login" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="75" Margin="238,169,0,0"
Click="btnLogin_Click" Background="PapayaWhip"/>
        <Image x:Name="imgLogo" HorizontalAlignment="Left" Height="146"
Margin="10,60,0,0" VerticalAlignment="Top" Width="147" Stretch="Fill"
Source="https://icosst.kics.edu.pk/2018/wp-
content/uploads/2018/07/icons1.png"/>
        <GroupBox Header="Login" HorizontalAlignment="Left" Height="156"
Margin="162,50,0,0" VerticalAlignment="Top" Width="245"/>
```

```
        <Label Content="Student Information System" HorizontalAlignment="Left"
Margin="73,10,0,0"          VerticalAlignment="Top"          FontSize="20"
FontWeight="Bold"/>

    </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_Click(object sender, RoutedEventArgs e)
        {
            string user, pass;
            user = txtUser.Text;
            pass = txtPass.Password;
            if(user != "admin")
            {
                MessageBox.Show("Username is incorrect", "Error",
MessageBoxButton.OK, MessageBoxImage.Error);
                txtUser.Clear();
            }
            else if(pass != "root")
            {
                MessageBox.Show("Password is incorrect", "Error",
MessageBoxButton.OK, MessageBoxImage.Error);
                txtPass.Clear();
            }

            else
            {
                MessageBox.Show("Login Successful!", "Success",
MessageBoxButton.OK, MessageBoxImage.Information);
                this.Hide();
                StartPage start = new StartPage();
                start.ShowDialog();
            }
        }
    }
}
```

StartPage.xaml

```
<Window x:Class="StudentInformationSystem.StartPage"
        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="Start Page" Height="293.983" Width="409.648"
Background="LightBlue" WindowStartupLocation="CenterScreen">
    <Grid>
        <Button x:Name="btnData" Content="Student Enroll"
HorizontalAlignment="Left" VerticalAlignment="Top" Width="147"
Margin="128,37,0,0" Height="49" FontSize="16" Background="PapayaWhip"
Click="btnData_Click"/>
        <Button x:Name="btnImport" Content="Import CSV"
HorizontalAlignment="Left" VerticalAlignment="Top" Width="147"
Margin="128,107,0,0" Height="49" Background="PapayaWhip" FontSize="16"
Click="btnImport_Click"/>
        <Button x:Name="btnChart" Content="Chart" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="147" Margin="128,179,0,0" Height="49"
Background="PapayaWhip" FontSize="16" Click="btnChart_click"/>

    </Grid>
</Window>
```

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

        private void btnData_Click(object sender, RoutedEventArgs e)
        {
            this.Hide();
            StudentEnrol startPage = new StudentEnrol();
            startPage.ShowDialog();
        }

        private void btnImport_Click(object sender, RoutedEventArgs e)
        {
            this.Hide();
            ImportCSV csv = new ImportCSV();
            csv.ShowDialog();
        }
```

```
        private void btnChart_click(object sender, RoutedEventArgs e)
        {
            this.Hide();
            WeeklyChart chart = new WeeklyChart();
            chart.ShowDialog();
        }
    }

}
```

StudentEnrol.xaml

```
<Window x:Class="StudentInformationSystem.StudentEnrol"
        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="Student Enroll" Height="450" Width="935.6"
Background="LightBlue" WindowStartupLocation="CenterScreen">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition Width="0*"/>
        </Grid.ColumnDefinitions>

        <Label Content="Student Id" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="10,39,0,0" Height="27" Width="113"/>
        <TextBox x:Name="txtID" HorizontalAlignment="Left" Height="27"
TextWrapping="Wrap" VerticalAlignment="Top" Width="137"
Margin="123,39,0,0"/>
```

```
<Label Content="Student Name" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="10,69,0,0" Height="27" Width="108"/>
<TextBox x:Name="txtName" HorizontalAlignment="Left" Height="27"
TextWrapping="Wrap" VerticalAlignment="Top" Width="137"
Margin="123,69,0,0"/>

<Label Content="Student Address" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="10,101,0,0" Height="27" Width="108"/>
<TextBox x:Name="txtAddress" HorizontalAlignment="Left" Height="27"
TextWrapping="Wrap" VerticalAlignment="Top" Width="137"
Margin="123,101,0,0"/>

<Label Content="Student Contact" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="10,132,0,0" Height="27" Width="108"/>
<TextBox x:Name="txtContact" HorizontalAlignment="Left" Height="27"
TextWrapping="Wrap" VerticalAlignment="Top" Width="137"
Margin="123,132,0,0"/>

<Label Content="Course Enrol" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="10,162,0,0" Height="27" Width="108"
RenderTransformOrigin="0.512,1.072"/>
<ComboBox x:Name="cbCourse" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="137" Margin="123,162,0,0" Height="27">
    <ComboBoxItem Content="Computing"/>
    <ComboBoxItem Content="Multimedia Technologies"/>
    <ComboBoxItem Content="Networks and IT Security"/>
</ComboBox>

<Label Content="Registration Date" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="10,192,0,0" Height="27" Width="108"/>
<DatePicker x:Name="date" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="123,192,0,0" Width="137" Height="27"/>
```

```
        <Button Content="Add Student" HorizontalAlignment="Left"
Background="PapayaWhip" Margin="98,247,0,0" VerticalAlignment="Top"
Width="83" Click="btnSave_clicked" Height="19"/>

        <DataGrid x:Name="DataGridXAML" Height="180"
VerticalAlignment="Top"  Margin="265,39,10.4,0">

            <DataGrid.Resources>
                <Style TargetType="{x:Type DataGridColumnHeader}">
                    <Setter Property="Background" Value="#FFE4AE"/>
                    <Setter Property="FontWeight" Value="SemiBold"/>
                    <Setter Property="BorderThickness" Value="0 0 1 2"/>
                    <Setter Property="BorderBrush" Value="Black"/>
                </Style>

            </DataGrid.Resources>



        </DataGrid>
        <Button Content="&lt;&lt; Back" HorizontalAlignment="Left" Width="115"
Margin="0,0,0,386" FontSize="20" Background="LightBlue"
Click="btnBack_click"/>

    </Grid>
</Window>
```

StudentEnrol.xaml.cs

```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
```

```
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 StudentEnrol.xaml
    /// </summary>
    public partial class StudentEnrol : Window
    {

        public StudentEnrol()
        {
            InitializeComponent();

            LoadGrid();

        }
        private void AddStudentData(DataSet dataSet)
        {
            var newRow = dataSet.Tables["Student"].NewRow();
            newRow["studentId"] = txtID.Text;
            newRow["studentName"] = txtName.Text;
            newRow["studentAddress"] = txtAddress.Text;
            newRow["studentContact"] = txtContact.Text;
```

```
            newRow["courseEnrol"] = cbCourse.Text;
            newRow["registrationDate"] = date.SelectedDate;
            dataSet.Tables["Student"].Rows.Add(newRow);

            /*Student dataStudent = new Student();
            dataStudent.studentId = txtID.Text;
            dataStudent.studentName = txtName.Text;
            dataStudent.studentAddress = txtAddress.Text;
            dataStudent.studentContact = txtContact.Text;
            dataStudent.courseEnrol = cbCourse.Text;
            dataStudent.registrationDate = date.Text;

            DataGridXAML.Items.Add(dataStudent);*/
        }

        private void btnSave_clicked(object sender, RoutedEventArgs e)
        {
            if (txtID.Text == "" || txtName.Text == "" || txtAddress.Text == "" ||
txtContact.Text == "" || cbCourse.Text == "" || date.Text == "")
            {
                MessageBox.Show("Enter a valid input field.","Message");
            }
            else
            {
                try
                {
                    var handler = new DataHandler();
                    var dataSet = new DataSet();

                    if (File.Exists(@"C:\DataHandler\StudentData.xml"))
                    {
                        dataSet.ReadXml(@"C:\DataHandler\StudentData.xml");
```

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

                }
                AddStudentData(dataSet);
                dataSet.WriteXml(@"C:\DataHandler\StudentData.xml");
                LoadGrid();
                txtID.Text = "";
                txtName.Text = "";
                txtAddress.Text = "";
                txtContact.Text = "";
                cbCourse.Text = "";
                date.Text = "";
                MessageBox.Show("Added Successfully", "Message",
MessageBoxButton.OK, MessageBoxImage.Information);
            }
            catch (Exception)
            {

            }
        }



    }
    public void LoadGrid()
    {
        var dataSet = new DataSet();
        if (File.Exists(@"C:\DataHandler\StudentData.xml"))
        {
            dataSet.ReadXml(@"C:\DataHandler\StudentData.xml");
```

```
            DataGridXAML.ItemsSource =
dataSet.Tables["Student"].DefaultView;
        }
    }

    private void btnBack_click(object sender, RoutedEventArgs e)
    {
        this.Hide();
        StartPage back = new StartPage();
        back.ShowDialog();
    }
  }
}
```

ImportCSV.xaml

```
<Window x:Class="StudentInformationSystem.ImportCSV"
    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="ImportCSV" Height="450" Width="588.8" Background="LightBlue"
WindowStartupLocation="CenterScreen">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition/>
      <ColumnDefinition Width="0*"/>
    </Grid.ColumnDefinitions>
```

```
        <StackPanel HorizontalAlignment="Left" Height="146"
VerticalAlignment="Top" Width="235" Margin="10,274,0,0">
            <Button Content="Retrieve Data" Click="btnRetrieve_click"
Background="PapayaWhip"/>
            <Button Content="Import CSV" Click="btnImport_click"
Background="PapayaWhip"/>
            <Button Content="Sort by Name" Click="btnName_Sort"
Background="PapayaWhip"/>
            <Button Content="Sort by Registration Date" Click="btnRegDate_Sort"
Background="PapayaWhip"/>
            <Button Content="Weekly Report" Click="btnWeek_Report"
Background="PapayaWhip"/>
        </StackPanel>
        <DataGrid x:Name="DataGridXAML2" HorizontalAlignment="Left"
Height="230" VerticalAlignment="Top" Width="562" Margin="10,39,0,0">

            <DataGrid.Resources>
                <Style TargetType="{x:Type DataGridColumnHeader}">
                    <Setter Property="Background" Value="#FFE4AE"/>
                    <Setter Property="FontWeight" Value="SemiBold"/>
                    <Setter Property="BorderThickness" Value="0 0 1 2"/>
                    <Setter Property="BorderBrush" Value="Black"/>
                </Style>

            </DataGrid.Resources>

        </DataGrid>
        <Button Content="&lt;&lt; Back" HorizontalAlignment="Left" Width="115"
Margin="0,0,0,386" FontSize="20" Background="LightBlue"
Click="btnImportBack_click"/>

    </Grid>
</Window>
```

ImportCSV.xaml.cs

```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
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 ImportCSV.xaml
    /// </summary>
    public partial class ImportCSV : Window
    {
        DataTable dataTable;
        public ImportCSV()
        {
            InitializeComponent();
        }

        private void DataGridShow()
```

```
{
    string dataXMLFile = @"C:\DataHandler\StudentData.xml";
    System.Data.DataSet dataset = new DataSet();
    dataset.ReadXml(dataXMLFile);

    dataTable = new DataTable("dt");
    dataTable.Columns.Add("studentId", typeof(String));
    dataTable.Columns.Add("studentName", typeof(String));
    dataTable.Columns.Add("studentAddress", typeof(String));
    dataTable.Columns.Add("studentContact", typeof(String));
    dataTable.Columns.Add("courseEnrol", typeof(String));
    dataTable.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);
        dataTable.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(dataTable);
    DataGridXAML2.ItemsSource = dataView;
}

private void btnRetrieve_click(object sender, RoutedEventArgs e)
{
    DataGridShow();
}
private void btnImport_click(object sender, RoutedEventArgs e)
```

```
        {
            var dataSet = new DataSet();
            dataSet.ReadXml(@"C:\DataHandler\StudentData.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["studentId"] = values[0];
                        newRow["studentName"] = values[1];
                        newRow["studentAddress"] = values[2];
                        newRow["studentContact"] = values[3];
                        newRow["courseEnrol"] = values[4];
                        newRow["registrationDate"] = values[5];
                        dataSet.Tables["Student"].Rows.Add(newRow);

                        dataSet.WriteXml(@"C:\DataHandler\StudentData.xml");
                    }
                }
                DataGridXAML2.ItemsSource                                    =
dataSet.Tables["Student"].DefaultView;
            }
        }
```

```
        private void btnName_Sort(object sender, RoutedEventArgs e)
        {
            DataView dataView = new DataView(dataTable);
            dataView.Sort = "studentName";
            DataGridXAML2.ItemsSource = dataView;
        }

        private void btnRegDate_Sort(object sender, RoutedEventArgs e)
        {
            DataView dataView = new DataView(dataTable);
            dataView.Sort = "registrationDate";
            DataGridXAML2.ItemsSource = dataView;
        }



        private void btnWeek_Report(object sender, RoutedEventArgs e)
        {
            // declaring new data set
            var dataset = new DataSet();

            // reading main report
            dataset.ReadXml(@"C:\DataHandler\StudentData.xml");
            DataTable stdReport = dataset.Tables[0];
            // assigning initial values of Course to
            int computing = 0;
            int multimedia = 0;
            int networking = 0;

            DataTable dt = new DataTable("dt");
            dt.Columns.Add("Course Enroll", typeof(String));     // creating two
columns
            dt.Columns.Add("Total Students", typeof(int));
```

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


            String col = stdReport.Rows[i]["courseEnrol"].ToString();
            if (col == "Computing")
            {
                computing++;    // incrementing values of each course based on
user input
            }
            else if (col == "Multimedia Technologies")
            {
                multimedia++;
            }
            else if (col == "Networks and IT Security")
            {
                networking++;
            }
        }

        dt.Rows.Add("Computing", computing);          // final assign
        dt.Rows.Add("Multimedia Technologies", multimedia);
        dt.Rows.Add("Networks and IT Security", networking);


        DataGridXAML2.ItemsSource = dt.DefaultView; // is the name of data
grid

    }



    private void btnImportBack_click(object sender, RoutedEventArgs e)
    {
```

```
            this.Hide();
            StartPage goBack = new StartPage();
            goBack.ShowDialog();
        }
    }
}
```

WeeklyChart.xaml

```
<Window x:Class="StudentInformationSystem.WeeklyChart"
        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.Controls.DataVisualization.Toolkit"
        xmlns:DVC="clr-
namespace:System.Windows.Controls.DataVisualization.Charting;assembly=
System.Windows.Controls.DataVisualization.Toolkit"
        xmlns:local="clr-namespace:StudentInformationSystem"


        Title="WeeklyChart" Height="450" Width="800">
    <Grid>
        <DVC:Chart Margin="0" Title="Weekly Chart" Width="400" Height="250"
Background="Lavender">
            <DVC:PieSeries                                    x:Name="weeklychart"
IndependentValueBinding="{Binding                                  Path=Key}"
DependentValueBinding="{Binding Path=Value}">
            </DVC:PieSeries>
        </DVC:Chart>
```

```
    </Grid>
</Window>
```

WeeklyChart.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 WeeklyChart.xaml
/// </summary>
public partial class WeeklyChart : Window
{
    public WeeklyChart()
    {
        InitializeComponent();
        LoadPieChartData();
    }
    private void LoadPieChartData()
    {
        // declaring new data set
        var dataset = new DataSet();

        // reading main report
        dataset.ReadXml(@"C:\DataHandler\StudentData.xml");
        DataTable stdReport = dataset.Tables[0];
        // assigning initial values of Course to
        int computing = 0;
        int multimedia = 0;
        int networking = 0;

        DataTable dt = new DataTable("dt");
        dt.Columns.Add("Course Enroll", typeof(String));  // creating two
columns
```

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


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



                String col = stdReport.Rows[i]["courseEnrol"].ToString();
                if (col == "Computing")
                {
                    computing++;    // incrementing values of each course based on
user input
                }
                else if (col == "Multimedia Technologies")
                {
                    multimedia++;
                }
                else if (col == "Networks and IT Security")
                {
                    networking++;
                }
            }


            dt.Rows.Add("Computing", computing);          // final assign
            dt.Rows.Add("Multimedia Technologies", multimedia);
            dt.Rows.Add("Networks and IT Security", networking);
```

```
((System.Windows.Controls.DataVisualization.Charting.PieSeries)weeklychart
).ItemsSource =
                new KeyValuePair<string, int>[]{
        new KeyValuePair<string,int>("Computing", computing),
        new KeyValuePair<string,int>("Multimedia Technologies", multimedia),
        new KeyValuePair<string,int>("Networks and IT Security", networking) };


        }
    }
}
```