



Module Code & Module Title
CS6004NP & Application Development

Assessment Weightage & Type
30% Individual Course Work

3rd Year and 1st Semester

Name: Shreesha Khadka
College ID: NP04CP4A170035
University ID: 17030746

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.

Contents

1. Introduction	1
2. User Manual.....	2
3. System Architecture	11
3.1 Class Diagram.....	12
3.2 Flow Chart.....	14
4. Sorting Algorithm.....	16
5. Reflection.....	19
6. References.....	20
7. Appendix.....	21

Table of Figure

Figure 1: Login Screen.....	2
Figure 2: Login details form	3
Figure 3: Student Form	4
Figure 4: Import CSV Form.....	4
Figure 5: Import CSV detail form.....	5
Figure 6: Data from CSV form	5
Figure 7: Empty error message	6
Figure 8: Filling the student details	6
Figure 9: Sorting Page	7
Figure 10: Retrieving data.....	7
Figure 11: Sorting by name.....	8
Figure 12: Sorting by date.....	8
Figure 13: Generate weekly report page.....	9
Figure 14: Weekly report generated.....	9
Figure 15: Course Enrol Chart	10
Figure 16: System Architecture Diagram	11
Figure 17: Class Diagram	13
Figure 18: Flowchart Diagram.....	15

1. Introduction

The designed system for our coursework is of Student Information System. This system is developed and have done testing under different circumstances and different criteria. Different functions that the student information system seek are completed by this developed system. It consists of different types of features like adding the student details, importing the csv files to the data grid, viewing weekly report of how many students are enrolled in that course, retrieving data, sorting the data by student name and its registration data, Besides that the user can also view weekly task in a chart. Other available features are clearly explained in the user manual section. This type of application are rapidly growing in today's context. In today's scenario most of the school have already implemented this type of method as it is simple to keep the record of all the students of students. This type of system should also be implemented in different sectors of government offices also.

2. User Manual

Here are some screenshots which will allow user to operate with the system.

When the user operates with the system the first screen will be the login screen where user should input the username and password.

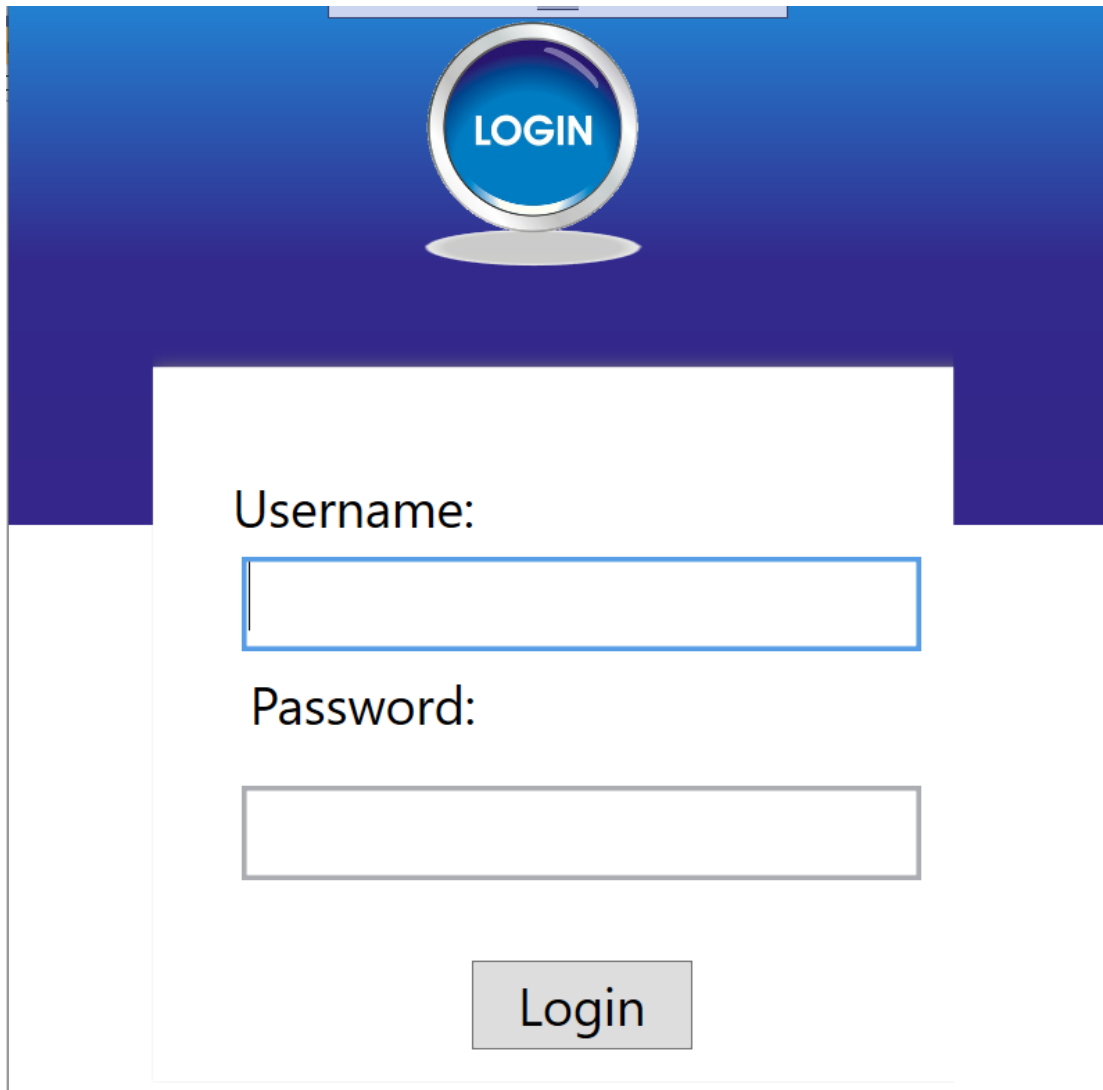
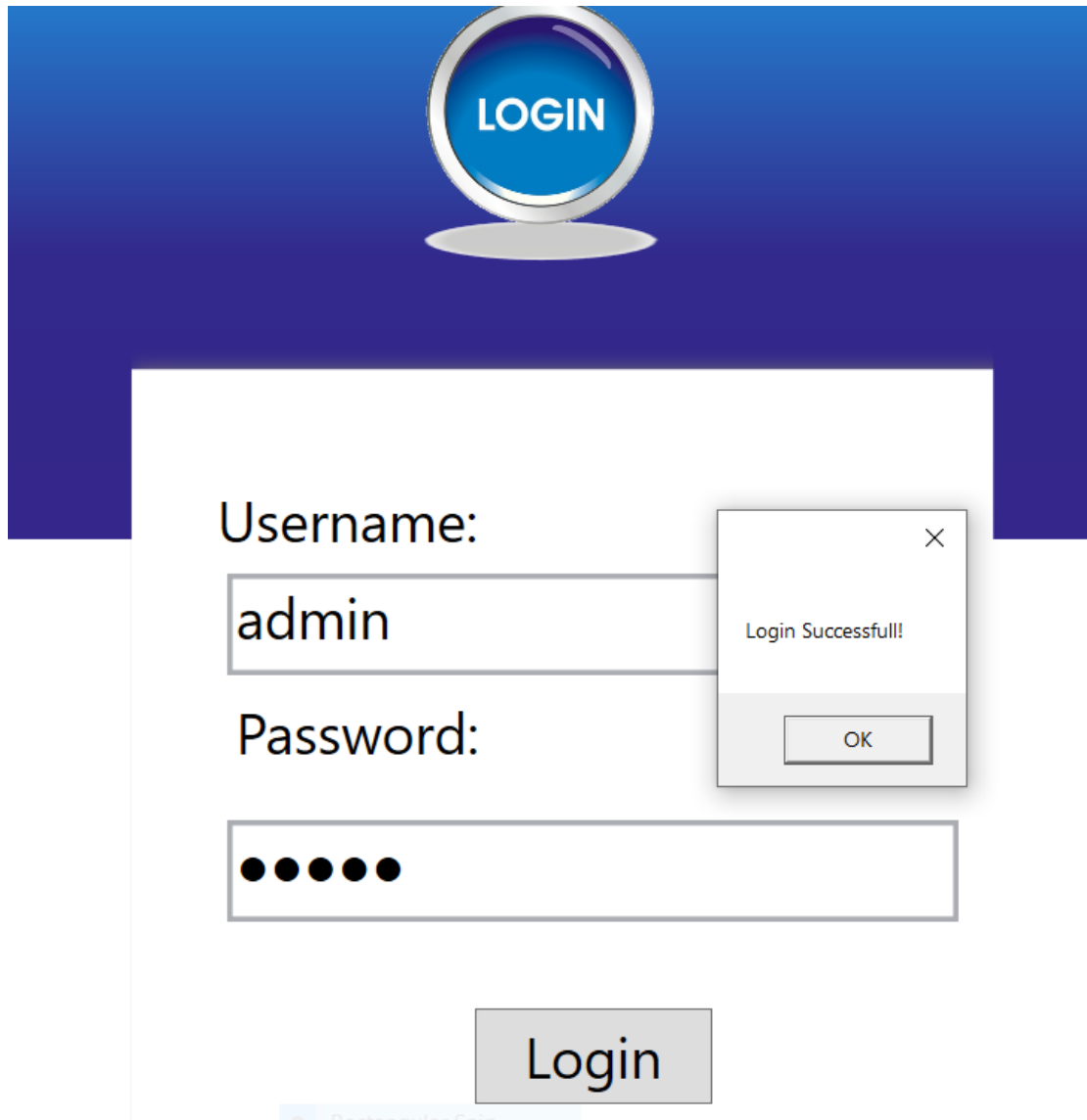
A screenshot of a login screen. The background is a blue gradient. At the top center is a circular logo with a silver border and a blue center containing the word "LOGIN" in white. Below the logo is a white rectangular box containing the login form. The form has two labels: "Username:" and "Password:". Below "Username:" is a white text input field with a blue border. Below "Password:" is a white text input field with a grey border. At the bottom of the white box is a grey button with the word "Login" in black text.

Figure 1: Login Screen

The username and password of this system is admin. If the user input incorrect user name and password they cannot login in the system.



The image shows a login interface. At the top, there is a blue header with a circular 'LOGIN' button. Below the header, the form is white. It contains a 'Username:' label followed by a text input field containing 'admin'. Below that is a 'Password:' label followed by a password input field with five black dots. A 'Login' button is positioned below the password field. To the right of the password field, a small dialog box is open, displaying 'Login Successfull!' and an 'OK' button. The dialog box has a close button (X) in the top right corner.

Figure 2: Login details form

After the login system there is student form and different types of button where the user can input the student details.

Student Form

Id:

Name:

Address:

Contact:

Course Enroll:

Registration Date:

ID	Name	Address	Contact	CourseEnroll	RegistrationDate
1	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019 9:09
2	Sworup Khadka	Pokhara	9866011289	Multimedia Technologies	1/5/2020 12:00:00 AM

Figure 3: Student Form

When the user clicks the import CSV button there comes a browser where the user can select csv form and open it.

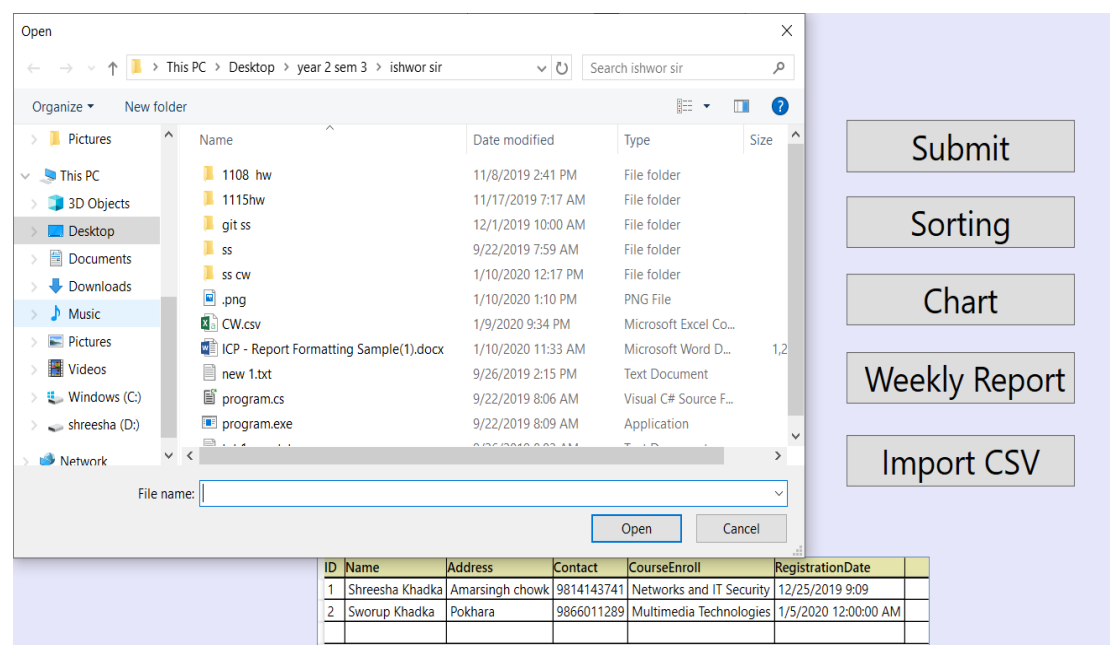


Figure 4: Import CSV Form

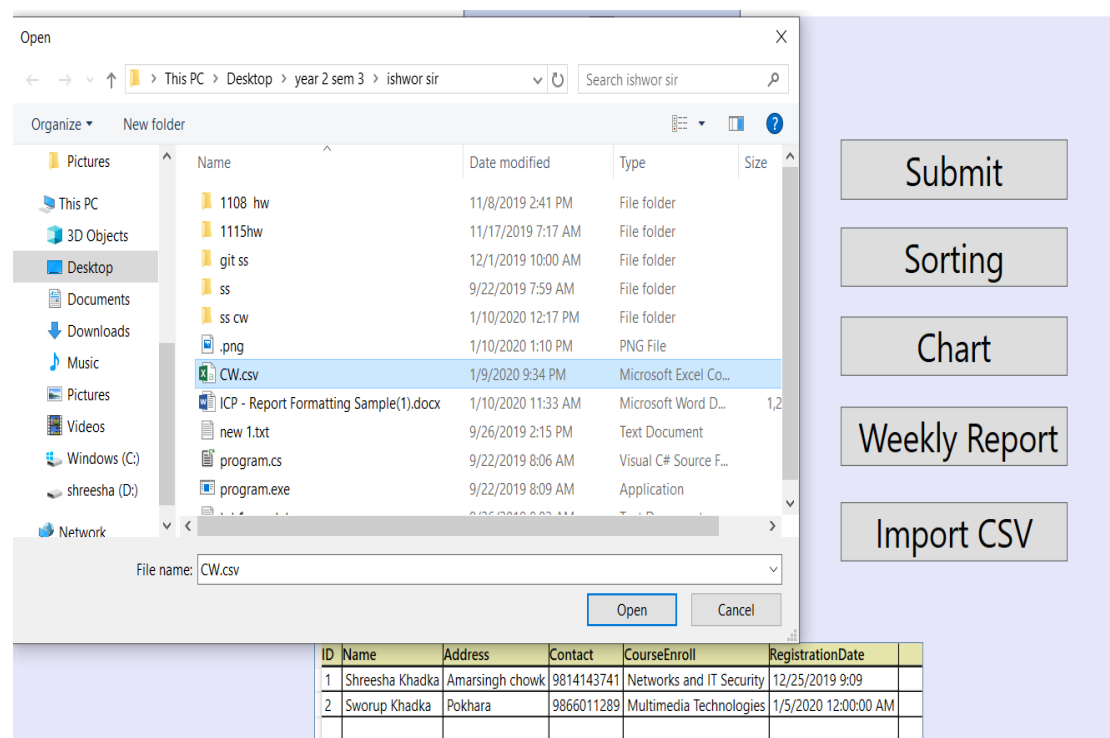


Figure 5: Import CSV detail form

After the csv file is open, the data from csv file comes in data grid.

Student Form

Id:

Name:

Address:

Contact:

Course Enroll:

Registration Date:

Submit

Sorting

Chart

Weekly Report

Import CSV

ID	Name	Address	Contact	CourseEnroll	RegistrationDate
1	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019 9:09
2	Sworup Khadka	Pokhara	9866011289	Multimedia Technologies	1/5/2020 12:00:00 AM
3	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019 9:09
4	Anazus Thapa	Simalchour	9804188091	Multimedia Technologies	12/26/2019 9:10
5	Oniya Gurung	Kaukhola	9824159251	Computing	1/3/2020 9:12
6	Sonu Lama	Ramghat	9817196537	Multimedia Technologies	1/3/2020 9:13
7	Niruta Devkota	Nalamukh	9814127354	Computing	1/7/2020 17:38
8	Binita Poudel	Rambazar	9846855591	Networks and IT Security	1/9/2020 11:07
9	Mahima BK	Dulegauda	9812345672	Computing	1/9/2020 11:07

Figure 6: Data from CSV form

When the user clicks the submit button without filling the details the error message displays in the screen.

Student Form

Id:

Name:

Address:

Contact:

Course Enroll:

Registration Date:

×
 Please fill the student details.

ID	Name	Address	Contact	CourseEnroll	RegistrationDate
1	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019 9:09
2	Sworup Khadka	Pokhara	9866011289	Multimedia Technologies	1/5/2020 12:00:00 AM
3	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019 9:09
4	Anazus Thapa	Simalchour	9804188091	Multimedia Technologies	12/26/2019 9:10
5	Oniya Gurung	Kaukhola	9824159251	Computing	1/3/2020 9:12
6	Sonu Lama	Ramghat	9817196537	Multimedia Technologies	1/3/2020 9:13
7	Niruta Devkota	Nalamukh	9814127354	Computing	1/7/2020 17:38
8	Binita Poudel	Rambazar	9846855591	Networks and IT Security	1/9/2020 11:07
9	Mahima RK	Dulegauda	9812345672	Computing	1/9/2020 11:07

Figure 7: Empty error message

When the user inputs all the student details the data comes in the data grid view.

Student Form

Id:

Name:

Address:

Contact:

Course Enroll:

Registration Date:

×
 Success

ID	Name	Address	Contact	CourseEnroll	RegistrationDate
1	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019 9:09
2	Sworup Khadka	Pokhara	9866011289	Multimedia Technologies	1/5/2020 12:00:00 AM
3	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019 9:09
4	Anazus Thapa	Simalchour	9804188091	Multimedia Technologies	12/26/2019 9:10
5	Oniya Gurung	Kaukhola	9824159251	Computing	1/3/2020 9:12
6	Sonu Lama	Ramghat	9817196537	Multimedia Technologies	1/3/2020 9:13
7	Niruta Devkota	Nalamukh	9814127354	Computing	1/7/2020 17:38
8	Binita Poudel	Rambazar	9846855591	Networks and IT Security	1/9/2020 11:07
9	Mahima BK	Dulegauda	9812345672	Computing	1/9/2020 11:07
10	Achin Thapa	Kathmandu	9804143741	Multimedia Technologies	1/10/2020 12:00:00 AM

Figure 8: Filling the student details

When the user clicks the sorting button from the student page the following page is displayed.

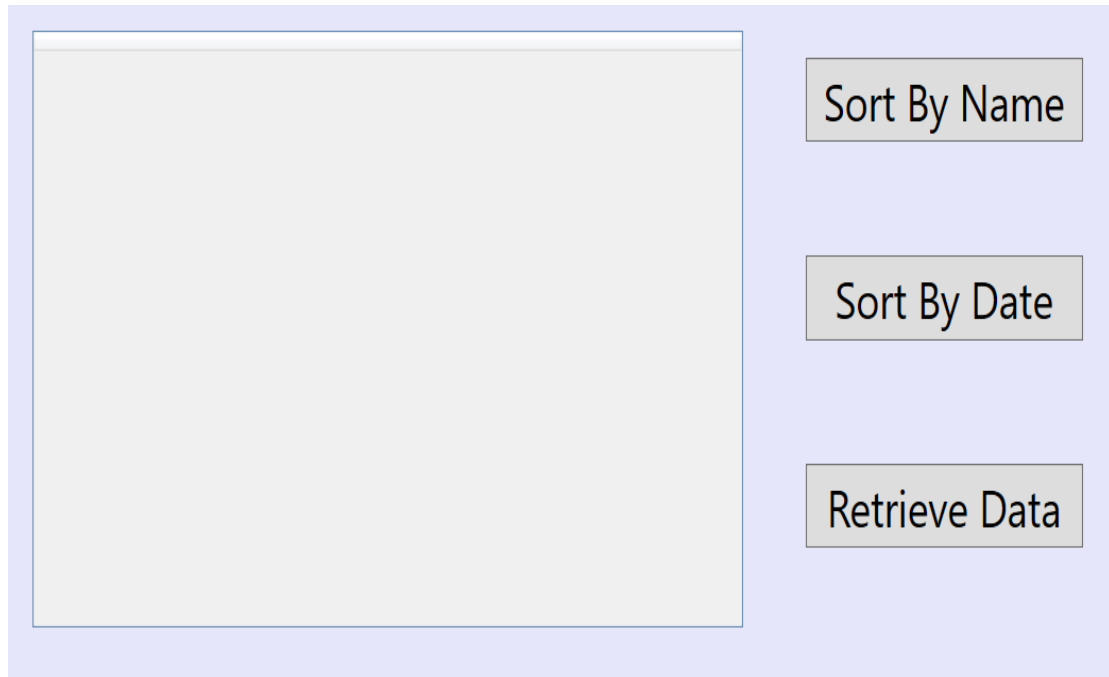


Figure 9: Sorting Page

After clicking the retrieve data button the data will be displayed in the data grid view from the xml file.

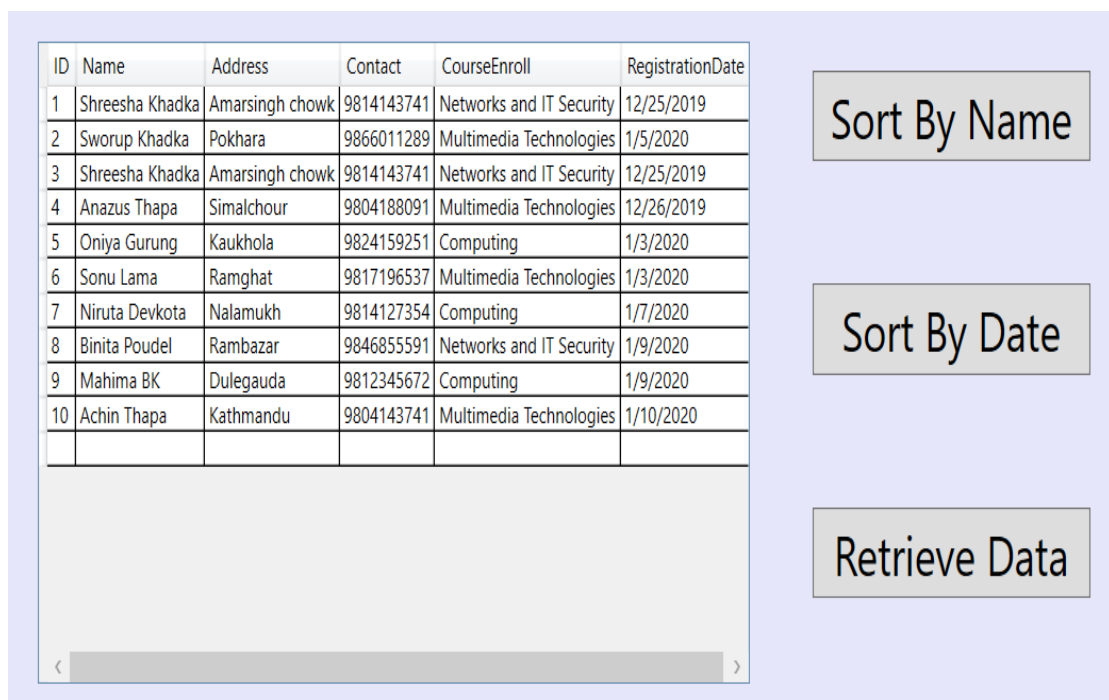


Figure 10: Retrieving data

When the user clicks the sort by name button the data from the data grid comes in arranging alphabetical form.

ID	Name	Address	Contact	CourseEnroll	RegistrationDate
10	Achin Thapa	Kathmandu	9804143741	Multimedia Technologies	1/10/2020
4	Anazus Thapa	Simalchour	9804188091	Multimedia Technologies	12/26/2019
8	Binita Poudel	Rambazar	9846855591	Networks and IT Security	1/9/2020
9	Mahima BK	Dulegauda	9812345672	Computing	1/9/2020
7	Niruta Devkota	Nalamukh	9814127354	Computing	1/7/2020
5	Oniya Gurung	Kaukhola	9824159251	Computing	1/3/2020
1	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019
3	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019
6	Sonu Lama	Ramghat	9817196537	Multimedia Technologies	1/3/2020
2	Sworup Khadka	Pokhara	9866011289	Multimedia Technologies	1/5/2020

Sort By Name

Sort By Date

Retrieve Data

Figure 11: Sorting by name

When the user clicks the sort by date button the data from the data grid comes in arranging date format.

ID	Name	Address	Contact	CourseEnroll	RegistrationDate
10	Achin Thapa	Kathmandu	9804143741	Multimedia Technologies	1/10/2020
5	Oniya Gurung	Kaukhola	9824159251	Computing	1/3/2020
6	Sonu Lama	Ramghat	9817196537	Multimedia Technologies	1/3/2020
2	Sworup Khadka	Pokhara	9866011289	Multimedia Technologies	1/5/2020
7	Niruta Devkota	Nalamukh	9814127354	Computing	1/7/2020
8	Binita Poudel	Rambazar	9846855591	Networks and IT Security	1/9/2020
9	Mahima BK	Dulegauda	9812345672	Computing	1/9/2020
1	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019
3	Shreesha Khadka	Amarsingh chowk	9814143741	Networks and IT Security	12/25/2019
4	Anazus Thapa	Simalchour	9804188091	Multimedia Technologies	12/26/2019

Sort By Name

Sort By Date

Retrieve Data

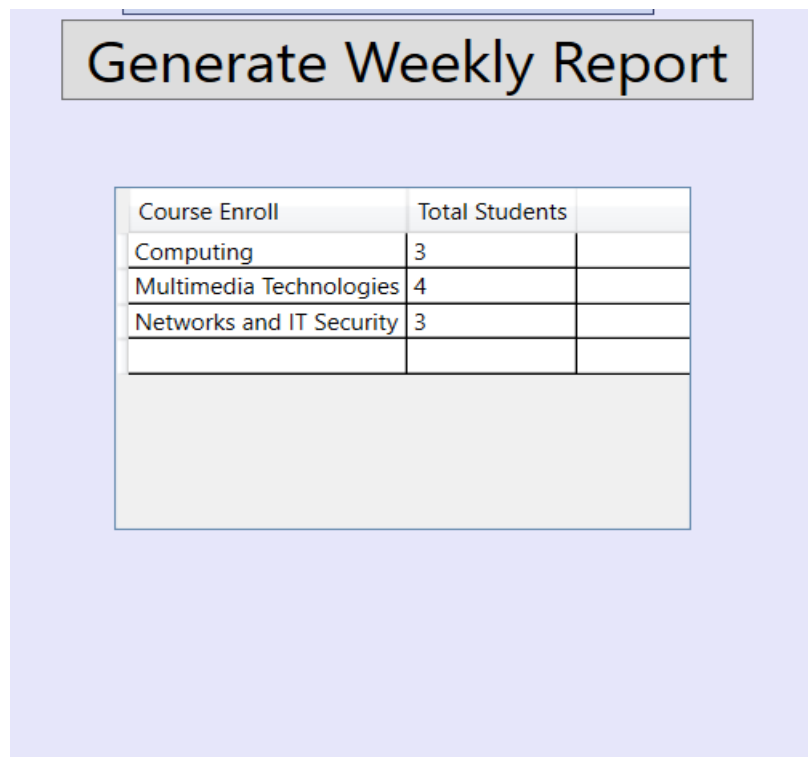
Figure 12: Sorting by date

When the user clicks the weekly report button from the main student form following generate page appears in the screen.



Figure 13: Generate weekly report page

When the user clicks generate weekly report button the data from course enrol appears. It shows that how many students are enrolled in which subject.

The image shows the same web page as Figure 13, but now the large grey box contains a table with course enrollment data. The table has three columns: "Course Enroll", "Total Students", and an empty column. The data rows are: Computing (3), Multimedia Technologies (4), and Networks and IT Security (3).

Course Enroll	Total Students	
Computing	3	
Multimedia Technologies	4	
Networks and IT Security	3	

Figure 14: Weekly report generated

When the user clicks the chart button from the student form the chart page is generated.

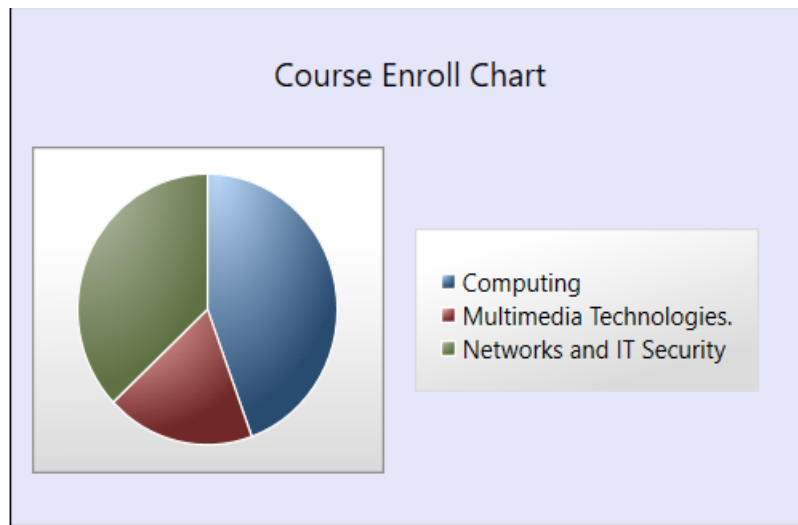


Figure 15: Course Enrol Chart

3. System Architecture

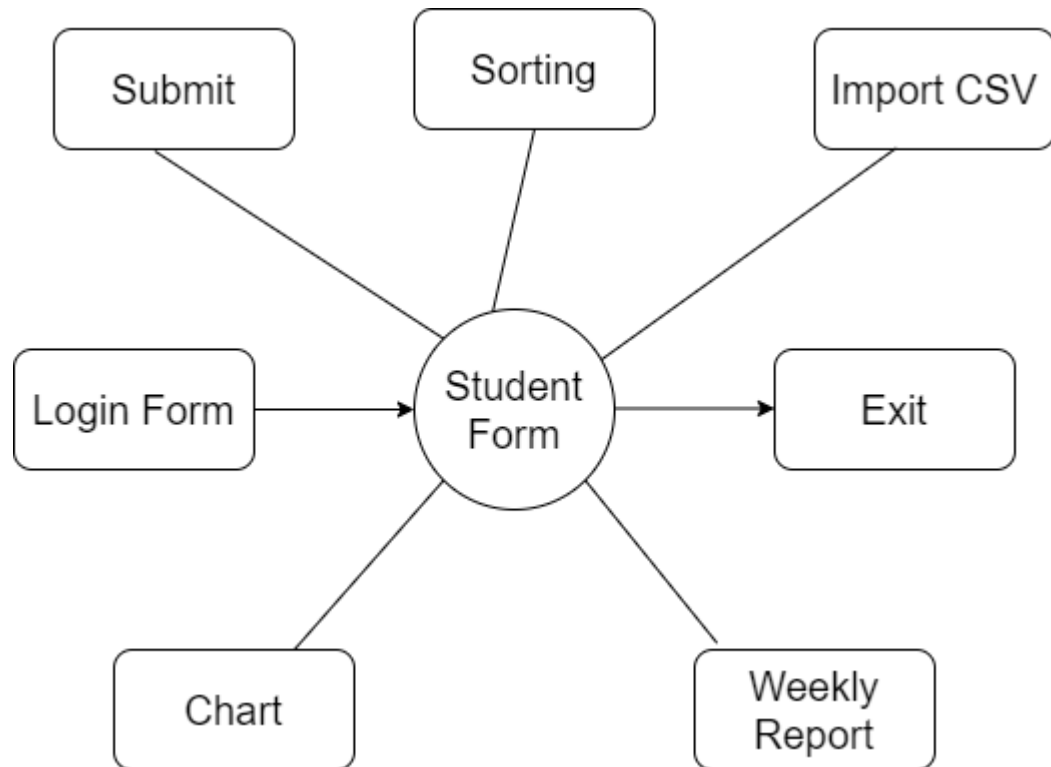


Figure 16: System Architecture Diagram

3.1 Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modelling paradigms have evolved. (Rouse, 2020)

UML Class Diagram gives an overview of a software system by displaying classes, attributes, operations, and their relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments. Class Diagram helps construct the code for the software application development. (guru99, 2020)

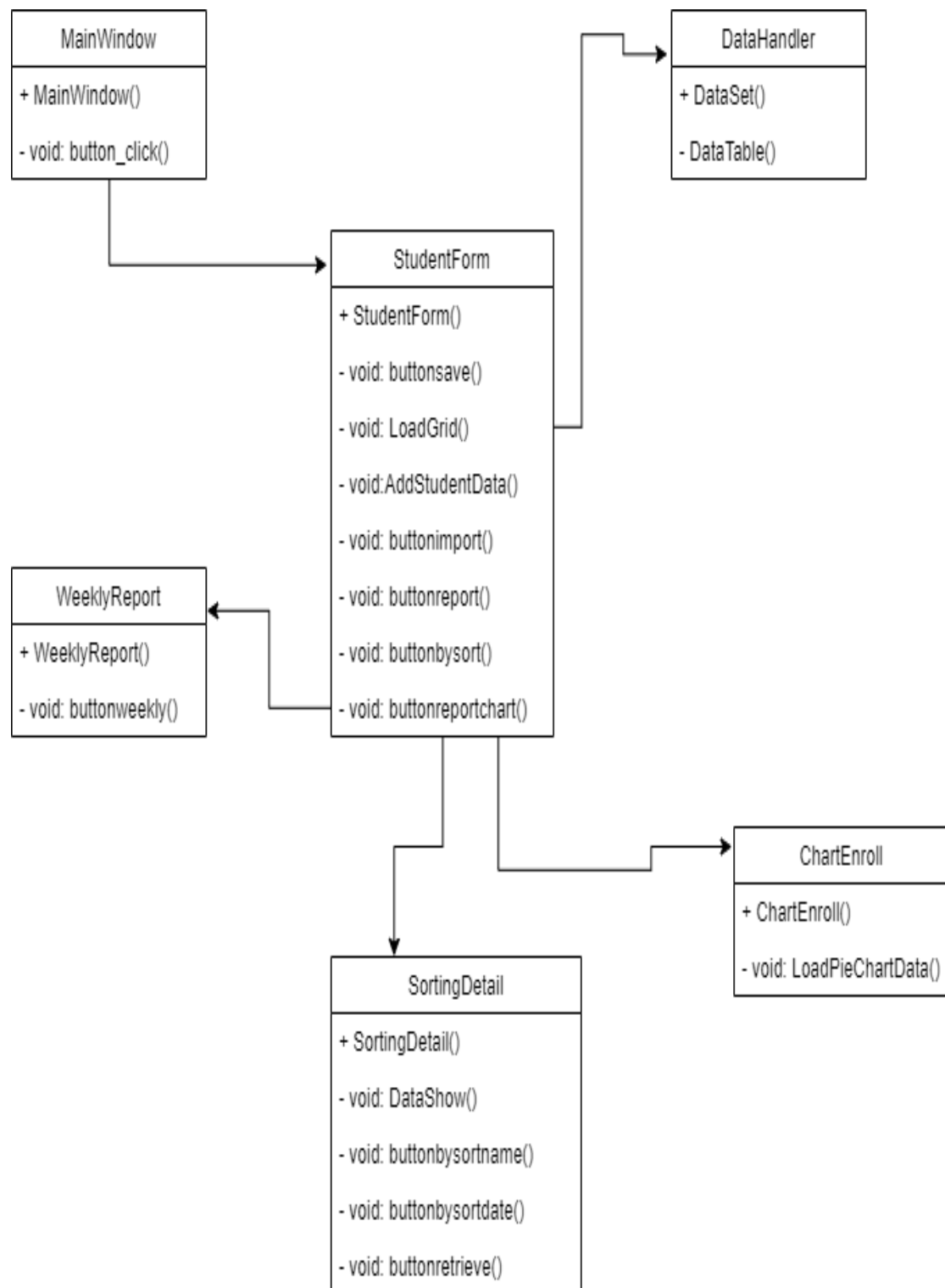


Figure 17: Class Diagram

3.2 Flow Chart

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 flow chart is to provide people with a common language or reference point when dealing with a project or process.

Flowcharts use simple geometric symbols and arrows to define relationships. In programming, for instance, the beginning or end of a program is represented by an oval. A process is represented by a rectangle, a decision is represented by a diamond and an I/O process is represented by a parallelogram. The Internet is represented by a cloud. (Rouse, 2020)

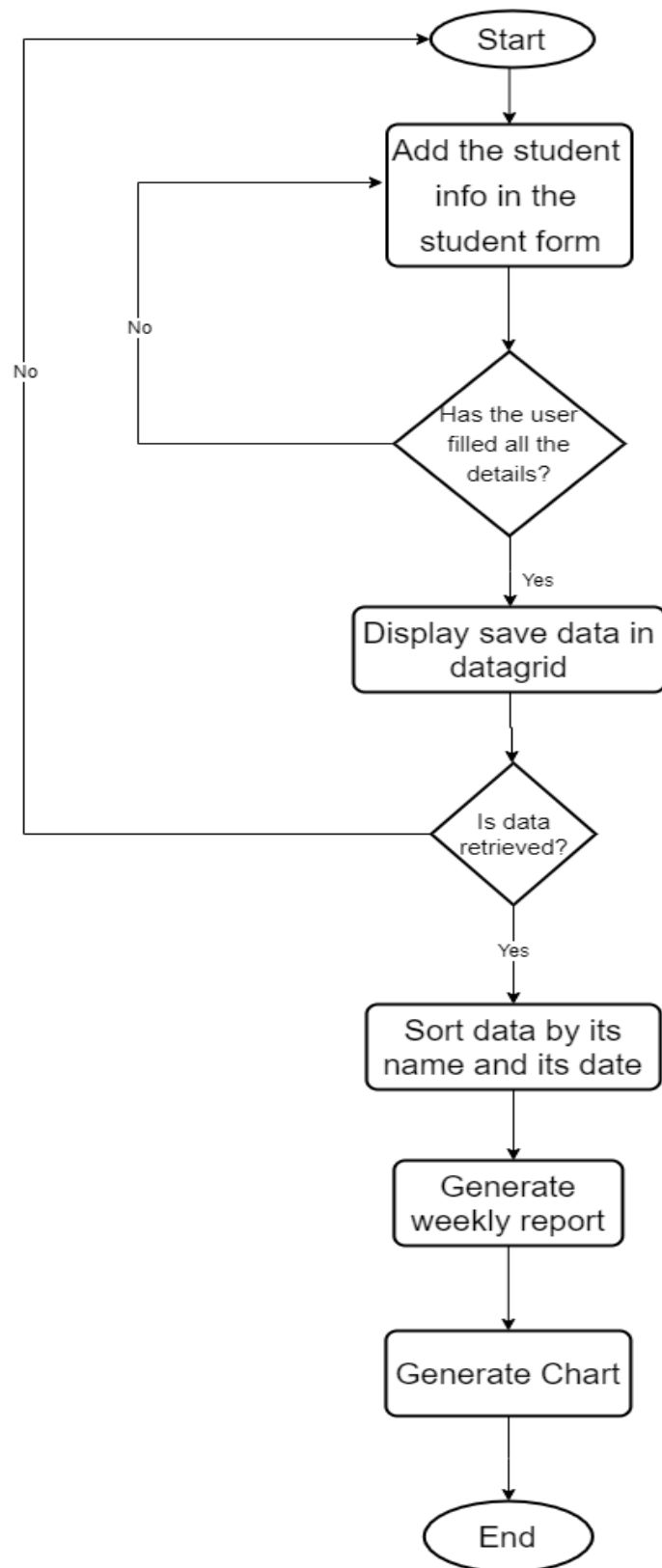


Figure 18: Flowchart Diagram

4. Sorting Algorithm

The algorithm used in our coursework is of bubble sort algorithm.

Bubble Sort is a simple algorithm which is used to sort a given set of n elements provided in form of an array with n number of elements. Bubble Sort compares all the element one by one and sort them based on their values. If the given array has to be sorted in ascending order, then bubble sort will start by comparing the first element of the array with the second element, if the first element is greater than the second element, it will swap both the elements, and then move on to compare the second and the third element, and so on. If we have total n elements, then we need to repeat this process for $n-1$ times. 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 rises up to the water surface. Bubble Sorting takes place by stepping through all the elements one-by-one and comparing it with the adjacent element and swapping them if required. (studytonight, 2020)

Bubble Sort algorithm occurs in various steps. (tutorialspoint, 2020)

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



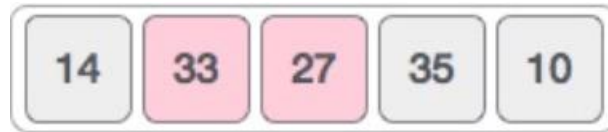
2. Bubble sort starts with very first two elements, comparing them to check which one is greater.



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



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



5. The new array should look like this.



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



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



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



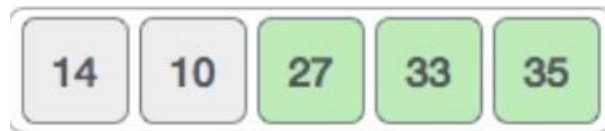
9. We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this.



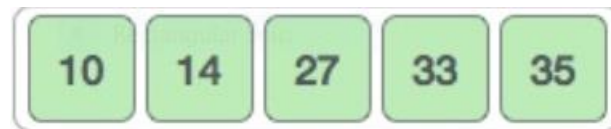
10. To be precise, we are now showing how an array should look like after each iteration. After the second iteration, it should look like this.



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



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



5. Reflection

Building up a system for a student information system isn't a simple undertaking. To clarify these communication, Class Diagram dive deep into the classes and their objects and methods. This system required a long time to be build using Visual Studio 2019 using C# programming language. The framework for this system has one student form screen where we can input the details of the student. After all the planning and analysis work, the implementation of these analogies was to prepare a good user interface. After the completion of this task, I learned to make a system that keeps the record of the student from their personal record like name, address, phone number and the course they are enrolled in the time of their registration. At first when the coursework was given to us I found it a bit difficult to do but I took help from my friend and concerning lecturer. While completing this coursework I faced many problems despite that I gained a lot of knowledge about the task, which was given for me and completed the task in a given period of time.

6. References

guru99, 2020. *www.guru99.com*. [Online]

Available at: <https://www.guru99.com/uml-class-diagram.html>

[Accessed 09 01 2020].

Rouse, M., 2020. *Searchapparchitecture.techtarget.com*. [Online]

Available at: <https://searchapparchitecture.techtarget.com/definition/class-diagram>

[Accessed 09 01 2020].

Rouse, M., 2020. *techtarget.com*. [Online]

Available at: <https://whatis.techtarget.com/definition/flowchart>

[Accessed 09 01 2020].

studytonight, 2020. *www.studytonight.com*. [Online]

Available at: <https://www.studytonight.com/data-structures/bubble-sort>

[Accessed 04 01 2020].

tutorialspoint, 2020. *www.tutorialspoint.com*. [Online]

Available at:

https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm

[Accessed 08 01 2020].

7. Appendix

MainWindow.xaml

```
<Window x:Class="Coursework.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:Coursework"
  mc:Ignorable="d"
  ResizeMode="NoResize"
  Title="Login Form" Height="668.643" Width="648.235">
  <Grid>
    <Rectangle Height="300" VerticalAlignment="Top">
      <Rectangle.Fill>
        <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
          <GradientStop Color="#FF2281D1"/>
          <GradientStop Color="#FF34268A" Offset="1"/>
          <GradientStop Color="#FF33288B" Offset="0.5"/>
        </LinearGradientBrush>
      </Rectangle.Fill>
    </Rectangle>
    <Image Source="pictures/log.png" Margin="234,10,262.8,0"
  VerticalAlignment="Top" />
    <Rectangle Width="470" Margin="83,0,88.6,10.2" Fill="White"
  VerticalAlignment="Bottom" Height="412">
      <Rectangle.Effect>
        <DropShadowEffect Color="#FFB4B4B4" Direction="0"
  BlurRadius="15" RenderingBias="Quality" ShadowDepth="1"/>
      </Rectangle.Effect>
    </Rectangle>
    <Label Content="Username:" HorizontalAlignment="Left"
  Margin="124,264,0,0" VerticalAlignment="Top" FontSize="30"/>
    <TextBox x:Name="username" HorizontalAlignment="Left"
  Margin="134,318,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
  Width="392" Height="55" BorderThickness="3" FontSize="30"/>
    <Label Content="Password:" HorizontalAlignment="Left"
  Margin="134,378,0,0" VerticalAlignment="Top" FontSize="30"/>
    <PasswordBox x:Name="userpassword" HorizontalAlignment="Left"
  Margin="134,450,0,0" VerticalAlignment="Top" Width="392" Height="55"
  BorderThickness="3" FontSize="30"/>
    <Button x:Name="Button" Content="Login" HorizontalAlignment="Left"
  Margin="267,551,0,0" VerticalAlignment="Top" Width="127" FontSize="30"
  Height="51" Click="button_click"/>

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

        private void button_click(object sender, RoutedEventArgs e)
        {
            if(userpassword.Password != "" && username.Text!= "" )
            {
                if(userpassword.Password == "admin" && username.Text ==
"admin")
                {
                    MessageBox.Show(" Login Successfull!");

                    StudentForm studentform = new StudentForm();
                    this.Hide();
                    studentform.ShowDialog();
                }
            }
        }
    }
}

```

StudentForm.xaml

```

<Window x:Class="Coursework.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:Coursework"
    mc:Ignorable="d"
    Title="StudentForm" Height="826.208" Width="1158.232"
    Background="Lavender">
    <Grid>
        <Label Content="Student Form" HorizontalAlignment="Left"
Margin="191,10,0,0" VerticalAlignment="Top" FontSize="40"/>
        <Label Content="Id:" HorizontalAlignment="Left" Margin="62,89,0,0"
VerticalAlignment="Top" FontSize="30" RenderTransformOrigin="0.44,1.004"/>
        <TextBox x:Name="studentid" HorizontalAlignment="Left"
Margin="379,101,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="319" Height="37" BorderThickness="3" FontSize="20"/>
        <Label Content="Name:" HorizontalAlignment="Left" Margin="62,144,0,0"
VerticalAlignment="Top" FontSize="30"/>
        <TextBox x:Name="studentname" HorizontalAlignment="Left"
Margin="379,157,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="319" Height="37" BorderThickness="3" FontSize="20"/>
        <Label Content="Address:" HorizontalAlignment="Left"
Margin="62,221,0,0" VerticalAlignment="Top" FontSize="30"/>
        <TextBox x:Name="studentaddress" HorizontalAlignment="Left"
Margin="379,221,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="319" Height="37" BorderThickness="3" FontSize="20"/>
        <Label Content="Contact:" HorizontalAlignment="Left"
Margin="62,279,0,0" VerticalAlignment="Top" FontSize="30"/>
        <TextBox x:Name="studentcontact" HorizontalAlignment="Left"
Margin="379,278,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="319" Height="37" BorderThickness="3" FontSize="20" />
        <Label Content="Course Enroll:" HorizontalAlignment="Left"
Margin="62,334,0,0" VerticalAlignment="Top" FontSize="30"/>
        <ComboBox x:Name="studentcourse" HorizontalAlignment="Left"
Margin="379,334,0,0" VerticalAlignment="Top" Width="319" Height="37"
BorderThickness="3" FontSize="20" >
            <ComboBoxItem Content="Networks and IT Security"/>
            <ComboBoxItem Content="Multimedia Technologies"/>
            <ComboBoxItem Content="Computing"/>
        </ComboBox>
        <Label Content="Registration Date:" HorizontalAlignment="Left"
Margin="62,401,0,0" VerticalAlignment="Top" FontSize="30"/>
        <DatePicker x:Name="studentdate" HorizontalAlignment="Left"
Margin="379,401,0,0" VerticalAlignment="Top" Width="319" Height="37"
BorderThickness="3" FontSize="20"/>
        <Button x:Name="btnsave" Content="Submit" HorizontalAlignment="Left"
Margin="801,93,0,0" VerticalAlignment="Top" Width="217" FontSize="30"
Height="45" Click="buttonsave"/>
        <Button x:Name="btnimport" Content="Import CSV"
HorizontalAlignment="Left" Margin="801,366,0,0" VerticalAlignment="Top"
Width="217" FontSize="30" Height="45" Click="buttonimport"/>
        <Button x:Name="btnreport" Content="Weekly Report"
HorizontalAlignment="Left" Margin="801,294,0,0" VerticalAlignment="Top"
Width="217" FontSize="30" Height="45" Click="buttonreport"/>
        <Button x:Name="btnsort" Content="Sorting" HorizontalAlignment="Left"
Margin="801,159,0,0" VerticalAlignment="Top" Width="217" FontSize="30"
Height="45" Click="buttonbysort"/>
    </Grid>

```

```

        <Button x:Name="btnreportchart" Content="Chart"
HorizontalAlignment="Left" Margin="801,226,0,0" VerticalAlignment="Top"
Width="217" FontSize="30" Height="45" Click="buttonreportchart"/>

        <DataGrid x:Name="DataGridView1"
            HorizontalAlignment="Left" Height="298" Margin="296,472,0,0"
VerticalAlignment="Top" Width="584">
            <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>

    </Grid>
</Window>

```

StudentForm.xaml.cs

```

<Window x:Class="Coursework.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:Coursework"
    mc:Ignorable="d"
    Title="StudentForm" Height="826.208" Width="1158.232"
    Background="Lavender">
    <Grid>
        <Label Content="Student Form" HorizontalAlignment="Left"
Margin="191,10,0,0" VerticalAlignment="Top" FontSize="40"/>
        <Label Content="Id:" HorizontalAlignment="Left" Margin="62,89,0,0"
VerticalAlignment="Top" FontSize="30" RenderTransformOrigin="0.44,1.004"/>
        <TextBox x:Name="studentid" HorizontalAlignment="Left"
Margin="379,101,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="319" Height="37" BorderThickness="3" FontSize="20"/>
        <Label Content="Name:" HorizontalAlignment="Left" Margin="62,144,0,0"
VerticalAlignment="Top" FontSize="30"/>
        <TextBox x:Name="studentname" HorizontalAlignment="Left"
Margin="379,157,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="319" Height="37" BorderThickness="3" FontSize="20"/>
        <Label Content="Address:" HorizontalAlignment="Left"
Margin="62,221,0,0" VerticalAlignment="Top" FontSize="30"/>
        <TextBox x:Name="studentaddress" HorizontalAlignment="Left"
Margin="379,221,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="319" Height="37" BorderThickness="3" FontSize="20"/>
        <Label Content="Contact:" HorizontalAlignment="Left"
Margin="62,279,0,0" VerticalAlignment="Top" FontSize="30"/>
        <TextBox x:Name="studentcontact" HorizontalAlignment="Left"
Margin="379,278,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="319" Height="37" BorderThickness="3" FontSize="20" />
        <Label Content="Course Enroll:" HorizontalAlignment="Left"
Margin="62,334,0,0" VerticalAlignment="Top" FontSize="30"/>
        <ComboBox x:Name="studentcourse" HorizontalAlignment="Left"
Margin="379,334,0,0" VerticalAlignment="Top" Width="319" Height="37"
BorderThickness="3" FontSize="20" >
            <ComboBoxItem Content="Networks and IT Security"/>

```

```

        <ComboBoxItem Content="Multimedia Technologies"/>
        <ComboBoxItem Content="Computing"/>
    </ComboBox>
    <Label Content="Registration Date:" HorizontalAlignment="Left"
Margin="62,401,0,0" VerticalAlignment="Top" FontSize="30"/>
    <DatePicker x:Name="studentdate" HorizontalAlignment="Left"
Margin="379,401,0,0" VerticalAlignment="Top" Width="319" Height="37"
BorderThickness="3" FontSize="20"/>
    <Button x:Name="btnsave" Content="Submit" HorizontalAlignment="Left"
Margin="801,93,0,0" VerticalAlignment="Top" Width="217" FontSize="30"
Height="45" Click="buttonsave"/>
    <Button x:Name="btnimport" Content="Import CSV"
HorizontalAlignment="Left" Margin="801,366,0,0" VerticalAlignment="Top"
Width="217" FontSize="30" Height="45" Click="buttonimport"/>
    <Button x:Name="btnreport" Content=" Weekly Report"
HorizontalAlignment="Left" Margin="801,294,0,0" VerticalAlignment="Top"
Width="217" FontSize="30" Height="45" Click="buttonreport"/>
    <Button x:Name="btnsort" Content="Sorting" HorizontalAlignment="Left"
Margin="801,159,0,0" VerticalAlignment="Top" Width="217" FontSize="30"
Height="45" Click="buttonbysort"/>
    <Button x:Name="btnreportchart" Content="Chart"
HorizontalAlignment="Left" Margin="801,226,0,0" VerticalAlignment="Top"
Width="217" FontSize="30" Height="45" Click="buttonreportchart"/>

    <DataGrid x:Name ="DataGridView1"
        HorizontalAlignment="Left" Height="298" Margin="296,472,0,0"
VerticalAlignment="Top" Width="584">
        <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>

</Grid>
</Window>

```

WeeklyReport.xaml

```

<Window x:Class="Coursework.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:Coursework"
    mc:Ignorable="d"
    Title="WeeklyReport" Height="463.847" Width="465.025"
    Background="Lavender">
    <Grid>
        <DataGrid x:Name="DataGridView3" HorizontalAlignment="Left"
Height="184" Margin="89,100,0,0" VerticalAlignment="Top" Width="288" />
        <Button x:Name="weekly" Content="Generate Weekly Report"
HorizontalAlignment="Left" Margin="62,10,0,0" VerticalAlignment="Top"
Width="346" FontSize="30" Click="buttonweekly"/>

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

        private void buttonweekly(object sender, RoutedEventArgs e)
        {
            var dataset = new DataSet();
            dataset.ReadXml(@"D:\student.xml");
            DataTable stdReport = dataset.Tables[0];
            int total_Com = 0;
            int total_Mul = 0;
            int total_Net = 0;

            DataTable dt = new DataTable("tbl");
            dt.Columns.Add("Course Enroll", typeof(String));
            dt.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")
                {
                    total_Com++; // incrementing values of each course based
on user input
                }
                else if (col == "Multimedia Technologies")
                {
                    total_Mul++;
                }
                else if (col == "Networks and IT Security")
                {
                    total_Net++;
                }
            }
        }
    }
}

```

```

        dt.Rows.Add("Computing", total_Com);           // final assign
        dt.Rows.Add("Multimedia Technologies", total_Mul);
        dt.Rows.Add("Networks and IT Security", total_Net);

        DataGridView3.ItemsSource = dt.DefaultView; // is the name of data
grid
    }
}
}
}

```

SortingDetail.xaml

```

<Window x:Class="Coursework.SortingDetail"
        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:Coursework"
        mc:Ignorable="d"
        Title="SortingDetail" Height="450" Width="858.916"
        Background="Lavender">
    <Grid>
        <DataGrid x:Name="DataGridView2" HorizontalAlignment="Left"
            Height="353" Margin="24,27,0,0" VerticalAlignment="Top" Width="535"/>
        <Button Content="Sort By Name" HorizontalAlignment="Left"
            Margin="606,43,0,0" VerticalAlignment="Top" Width="209" FontSize="30"
            Height="50" Click="buttonbysortname"/>
        <Button Content="Sort By Date" HorizontalAlignment="Left"
            Margin="606,160,0,0" VerticalAlignment="Top" Width="209" FontSize="30"
            Height="50" Click="buttonbysortdate"/>
        <Button Content="Retrieve Data" HorizontalAlignment="Left"
            Margin="606,283,0,0" VerticalAlignment="Top" Width="209" FontSize="30"
            Height="50" Click="buttonretrieve"/>
    </Grid>
</Window>

```

SortingDetail.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 Coursework
{
    /// <summary>
    /// Interaction logic for SortingDetail.xaml
    /// </summary>

```

```
public partial class SortingDetail : Window
{
    DataTable buffer;

    public SortingDetail()
    {
        InitializeComponent();
    }
    private void DataShow()
    {
        string dataXMLFile = @"D:\student.xml";
        System.Data.DataSet dataSet = new DataSet();
        dataSet.ReadXml(dataXMLFile);

        buffer = new DataTable("dt");
        buffer.Columns.Add("ID", typeof(string));
        buffer.Columns.Add("Name", typeof(string));
        buffer.Columns.Add("Address", typeof(string));
        buffer.Columns.Add("Contact", typeof(string));
        buffer.Columns.Add("CourseEnroll", typeof(string));
        buffer.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);
            buffer.Rows.Add(
                dataSet.Tables[0].Rows[i][0].ToString(),
                dataSet.Tables[0].Rows[i][1].ToString(),
                dataSet.Tables[0].Rows[i][2].ToString(),
                dataSet.Tables[0].Rows[i][3].ToString(),
                dataSet.Tables[0].Rows[i][4].ToString(),
                dtime.ToShortDateString());
        }
        DataView dataView = new DataView(buffer);
        DataGridView2.ItemsSource = dataView;
    }

    private void buttonbysortname(object sender, RoutedEventArgs e)
    {
        DataView dataView = new DataView(buffer);
        dataView.Sort = "Name";
        DataGridView2.ItemsSource = dataView;
    }

    private void buttonbysortdate(object sender, RoutedEventArgs e)
    {
        DataView dataView = new DataView(buffer);
        dataView.Sort = "RegistrationDate ";
        DataGridView2.ItemsSource = dataView;
    }

    private void buttonretrieve(object sender, RoutedEventArgs e)
    {
        DataShow();
    }
}
```

```
    }
}
```

ChartEnroll.xaml

```
<Window x:Class="Coursework.ChartEnroll"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:DVC="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"
        Title="ChartEnroll" Height="450" Width="800">
    <Grid>
        <DVC:Chart Margin="0" Title="Course Enroll Chart" Width="400"
Height="250" Background="Lavender">
            <DVC:PieSeries x:Name="course" IndependentValueBinding="{Binding
Path=Key}" DependentValueBinding="{Binding Path=Value}">
            </DVC:PieSeries>
        </DVC:Chart>

    </Grid>
</Window>
```

ChartEnroll.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 Coursework
{
    /// <summary>
    /// Interaction logic for ChartEnroll.xaml
    /// </summary>
    public partial class ChartEnroll : Window
    {
        public ChartEnroll()
        {
            InitializeComponent();
            LoadPieChartData();
        }
        private void LoadPieChartData()
        {
            var dataset = new DataSet();
            dataset.ReadXml(@"D:\student.xml");
            DataTable stdReport = dataset.Tables[0];
            int total_Com = 0;
        }
    }
}
```



```

int total_Mul = 0;
int total_Net = 0;

DataTable dt = new DataTable("tbl");
dt.Columns.Add("Course Enroll", typeof(String));
dt.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")
    {
        total_Com++; // incrementing values of each course based
on user input
    }
    else if (col == "Multimedia Technologies")
    {
        total_Mul++;
    }
    else if (col == "Networks and IT Security")
    {
        total_Net++;
    }
}

dt.Rows.Add("Computing", total_Com); // final assign
dt.Rows.Add("Multimedia Technologies", total_Mul);
dt.Rows.Add("Networks and IT Security", total_Net);

((System.Windows.Controls.DataVisualization.Charting.PieSeries)course).ItemsSource =
    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) };
}
}

```

DataHandler.cs

```

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

namespace Coursework
{
    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;
    }
}

```

Student.xml

```

<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Student>
    <ID>1</ID>
    <Name>Shreesha Khadka</Name>
    <Address>Amarsingh chowk</Address>
    <Contact>9814143741</Contact>
    <CourseEnroll>Networks and IT Security</CourseEnroll>
    <RegistrationDate>12/25/2019 9:09</RegistrationDate>
  </Student>
  <Student>
    <ID>2</ID>
    <Name>Sworup Khadka</Name>
    <Address>Pokhara</Address>
    <Contact>9866011289</Contact>
    <CourseEnroll>Multimedia Technologies</CourseEnroll>
    <RegistrationDate>1/5/2020 12:00:00 AM</RegistrationDate>
  </Student>
  <Student>
    <ID>3</ID>
    <Name>Shreesha Khadka</Name>

```

```
<Address>Amarsingh chowk</Address>
<Contact>9814143741</Contact>
<CourseEnroll>Networks and IT Security</CourseEnroll>
<RegistrationDate>12/25/2019 9:09</RegistrationDate>
</Student>
<Student>
  <ID>4</ID>
  <Name>Anazus Thapa </Name>
  <Address>Simalchour</Address>
  <Contact>9804188091</Contact>
  <CourseEnroll>Multimedia Technologies</CourseEnroll>
  <RegistrationDate>12/26/2019 9:10</RegistrationDate>
</Student>
<Student>
  <ID>5</ID>
  <Name>Oniya Gurung</Name>
  <Address>Kaukhola</Address>
  <Contact>9824159251</Contact>
  <CourseEnroll>Computing</CourseEnroll>
  <RegistrationDate>1/3/2020 9:12</RegistrationDate>
</Student>
<Student>
  <ID>6</ID>
  <Name>Sonu Lama</Name>
  <Address>Ramghat</Address>
  <Contact>9817196537</Contact>
  <CourseEnroll>Multimedia Technologies</CourseEnroll>
  <RegistrationDate>1/3/2020 9:13</RegistrationDate>
</Student>
<Student>
```

```
<ID>7</ID>
<Name>Niruta Devkota</Name>
<Address>Nalamukh</Address>
<Contact>9814127354</Contact>
<CourseEnroll>Computing</CourseEnroll>
<RegistrationDate>1/7/2020 17:38</RegistrationDate>
</Student>
<Student>
  <ID>8</ID>
  <Name>Binita Poudel</Name>
  <Address>Rambazar</Address>
  <Contact>9846855591</Contact>
  <CourseEnroll>Networks and IT Security</CourseEnroll>
  <RegistrationDate>1/9/2020 11:07</RegistrationDate>
</Student>
<Student>
  <ID>9</ID>
  <Name>Mahima BK</Name>
  <Address>Dulegauda</Address>
  <Contact>9812345672</Contact>
  <CourseEnroll>Computing</CourseEnroll>
  <RegistrationDate>1/9/2020 11:07</RegistrationDate>
</Student>
<Student>
  <ID>10</ID>
  <Name>Achin Thapa</Name>
  <Address>Kathmandu</Address>
  <Contact>9804143741</Contact>
  <CourseEnroll>Multimedia Technologies</CourseEnroll>
  <RegistrationDate>1/10/2020 12:00:00 AM</RegistrationDate>
```

</Student>

</NewDataSet>