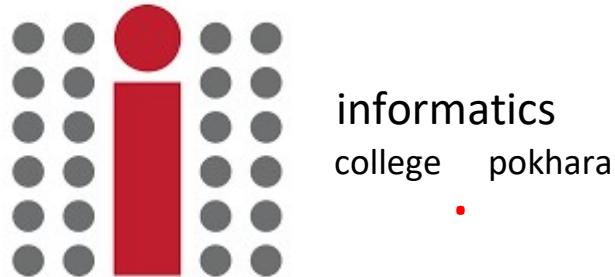


# Informatics College Pokhara



**Application Development**  
**CS6004NI**

**Course Work 1**

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

**Submitted To:** Ishwor Sapkota  
Module Leader

Component Grade and Comments	
<b>A. Implementation of Application</b>	
<b>User Interface and proper controls used for designing</b>	User Interface is complete but not separated and have proper use of controls
<b>Manual data entry or import from csv</b>	not properly saved or imported data
<b>Data Validation</b>	Only basic validation
<b>Enrollment Report &amp; weekly report in tabular format</b>	Any one of the report is missing or not complete
<b>Course wise enrollment report &amp; Chart display</b>	Very poorly designed and only contains one report format with in appropriate data
<b>Algorithm used for sorting &amp; proper sorting of data</b>	Default sorting provided by .net is used
<b>B. Documentation</b>	
<b>User Manual for running the application</b>	User Manual is below average. Is textual only.

## Marking Scheme

<b>Application architecture &amp; description of the classes ad methods sued</b>	average work with very limited explanation of the classes and methods used
<b>Flow chart, algoriathms and data sctructures used</b>	very poorly explained and no diagrammatic representation
<b>Reflective essay</b>	Average work with un clear learnings, experience or findings.

### **C. Programming Style**

<b>Clarity of code, Popper Naming convention &amp; comments</b>	very poorly written code and no comments at all
<b>System Usability</b>	very poorly developed application

<b>Overall Grade:</b>	<b>D+</b>	<b>D+</b>
-----------------------	-----------	-----------

### **Overall Comment:**

Code should be self explainable with less comments. Need some proper naming of the components and require to add comments on required area. Feature not working properly.

OK. Tried explaining the code and has completed most of the feature.



# Application Development

## CS6004NP

### Coursework 1

**Submitted By:**

Student Name: Clara Gurung

London Met ID: 17030521

Group: L3C2

Date: 10<sup>th</sup> January 2020

**Submitted To:**

Mr. Ishwor Sapkota

## **Table of Contents**

<b>1. Introduction</b> .....	1
<b>2. User Manual</b> .....	2
<b>3. Journal Articles</b> .....	9
<b>4. System Architecture</b> .....	12
4.1    Individual Diagram .....	13
4.2    Flow-Chart .....	17
<b>5. Sorting Algorithm</b> .....	18
<b>6. Conclusion</b> .....	21
<b>7. Bibliography</b> .....	22
<b>8. Appendix</b> .....	23

## Table of Figure

Figure 1:Login Form.....	2
Figure 2: Home Window.....	2
Figure 3: Validation Check .....	2
Figure 4: Validation Dialog Box .....	3
Figure 5: Student Enroll.....	3
Figure 6: Enroll Student 1.....	3
Figure 7: Successfully Enrolled .....	4
Figure 8: Clear Data.....	4
Figure 9: Import Details .....	4
Figure 10: Import CSV.....	5
Figure 11: Successfully Imported .....	5
Figure 12: Student Report .....	5
Figure 13: Reported Generated .....	6
Figure 14: Report .....	6
Figure 15: Generated by Name .....	6
Figure 16: Generated by Date .....	7
Figure 17: Total Students .....	7
Figure 18: Total .....	8
Figure 19: Pie Chart .....	8
Figure 20: Journal of Electrical and Computer Engineering .....	9
Figure 21: Student management system.....	10
Figure 22: Web Based Student Information Management System.....	11
Figure 23: Class Diagram.....	12
Figure 24: Flow-Chart.....	17

## Table of Table

Table 1:Bar Diagram .....	13
Table 2:Student Details .....	13
Table 3:Enroll Students .....	14
Table 4: Home.....	14
Table 5:Total No. Students.....	15
Table 6: Main Window.....	15
Table 7: Generate Report.....	16

## 1. Introduction

Application development is the process of creating a computer program or a set of programs to perform the different tasks that a business requires. From calculating monthly expenses to scheduling sales reports, applications help businesses automate processes and increase efficiency. Every app-building process follows the same steps: gathering requirements, designing prototypes, testing, implementation, and integration (Anon., 2020).

In this Coursework of Application Development, we are supposed to make student information form. To complete the coursework, we were even asked to make the form using the WPF framework. WPF is a UI framework that creates desktop client applications. The WPF development platform supports a broad set of application development features, including an application model, resources, controls, graphics, layout, data binding, documents, and security. The framework is part of .NET, so if you have previously built applications with .NET using ASP.NET or Windows Forms, the programming experience should be familiar. WPF uses the Extensible Application Markup Language (XAML) to provide a declarative model for application programming (Microsoft, 2020). We were asked to make student detail form and, in that form, I have made windows and in-home window I have made 5 buttons named TotalNoStudents, GenerateReport, EnrollStudentDetails, BarDiagram, ImportStudentDetails. After clicking on the button, a window will be opened. After clicking on the TotalNoStudents we can see the total number of students, in GenerateReport we can generate the report and in generate report sorting is also used sort by date and sort by name, in EnrollStudentDetails all the data needs to be filled up after filling the data's then only the report can be generated, in BarDiagram a chart has been created, in ImportStudentDetails there is a button name import which helps to import the .CSV file data.

To develop this project Visual Studio 2019 is used. As we were asked to complete the coursework using the WPF framework and to complete this given task C# language is used.

## 2. User Manual

Following are the Screenshots of the Student Form.

In this login form the user have to fill the username and the password. The username and the password of the system is “admin”.



Figure 1: Login Form

This is the Home window. It contains 5 buttons and if the user clicks the button then the particular window will be opened.

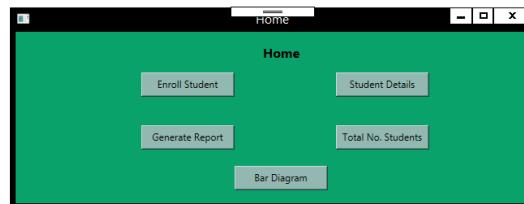


Figure 2: Home Window

User must fill the form.

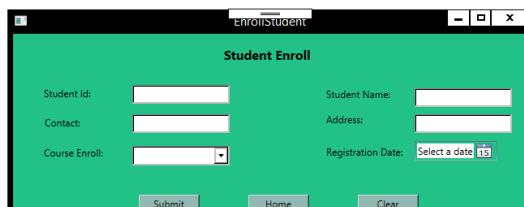


Figure 3: Validation Check

If user does not fill the form then the dialog box will appear.

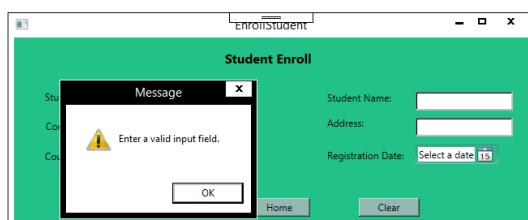


Figure 4: Validation Dialog Box

After clicking on the Enroll Student Button this window will appear where the user must fill the form.

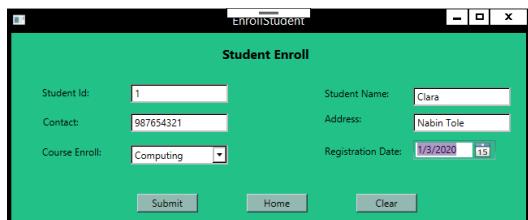


Figure 5: Student Enroll

After filling the form and the user must click on the submit button to save the details.

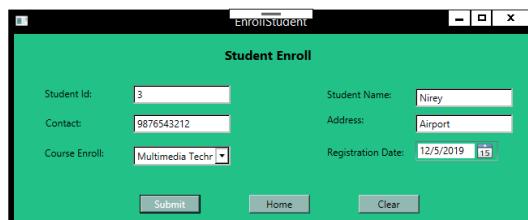


Figure 6: Enroll Student 1

After clicking the submit button a dialog box will appear as you can see in the given screenshot

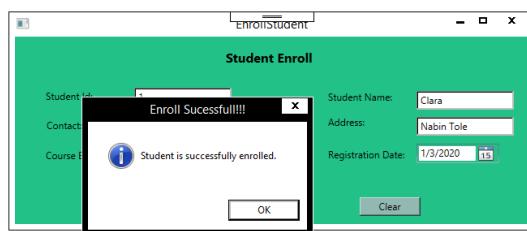


Figure 7: Successfully Enrolled

The user can also clear the data by clicking on the clear button.

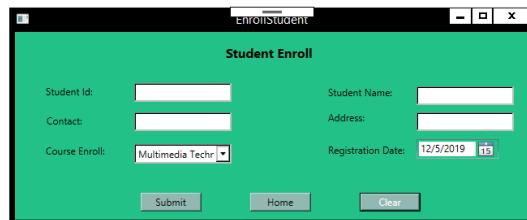


Figure 8: Clear Data

In the given picture we can that this is the Student Details window by clicking on the import button we can import the details from the CSV file.

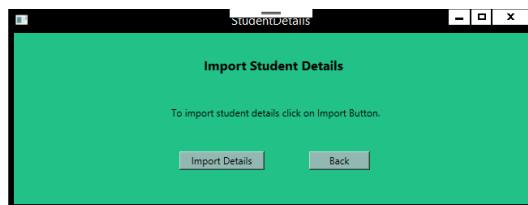


Figure 9: Import Details

After clicking the import details button then this window will appear after this window appear we should select the CVS file to import the details.

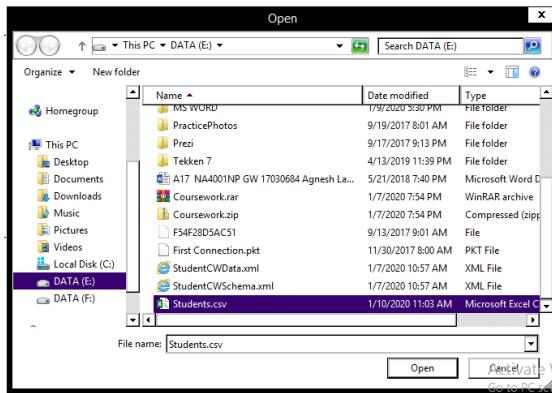


Figure 10: Import CSV

After importing it the dialog will appear which says successfully imported and recorded all student details.

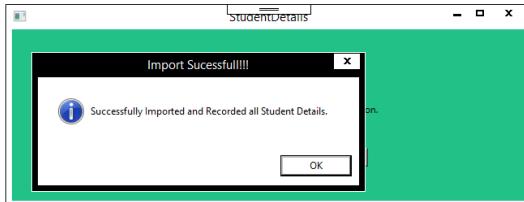


Figure 11: Successfully Imported

This is the Generate Report window where we can generate the report. To see the report user must click the Generate button.

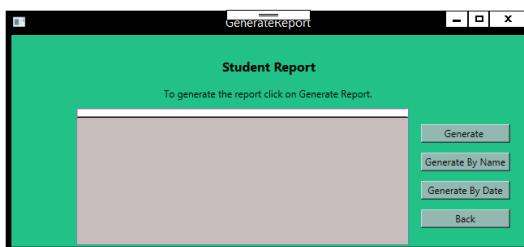


Figure 12: Student Report

After clicking on the Generate button we can see the dialog box appearing on the given screenshot.

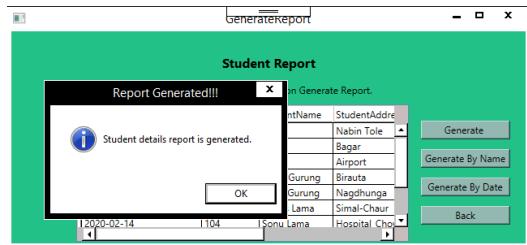


Figure 13: Reported Generated

User can see the report in this window.

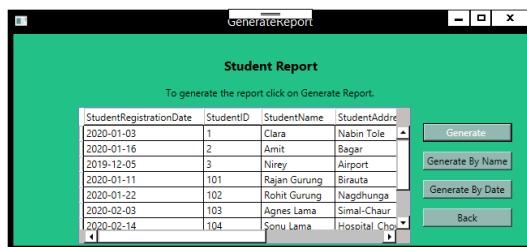


Figure 14: Report

User can even generate the report by name.

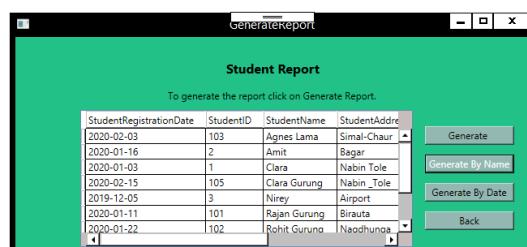


Figure 15: Generated by Name

User can also Generate the report by date by clicking on the generate by date button

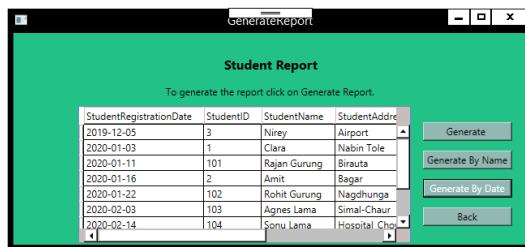


Figure 16: Generated by Date

This is the Total No of student's window where the total number of students can be seen.

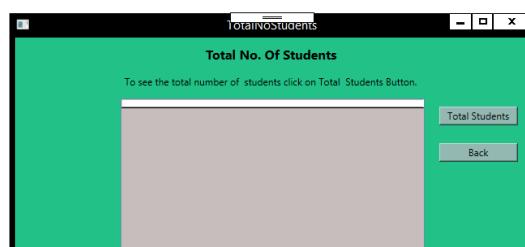


Figure 17: Total Students

To see the total number user must click on the Total Student button.

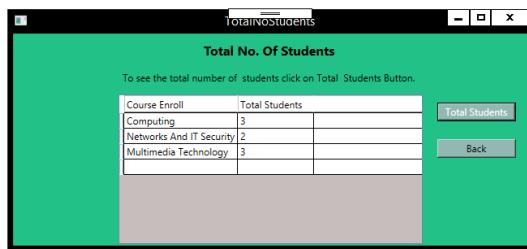


Figure 18: Total

This is the Bar Diagram window where user can see the chart.

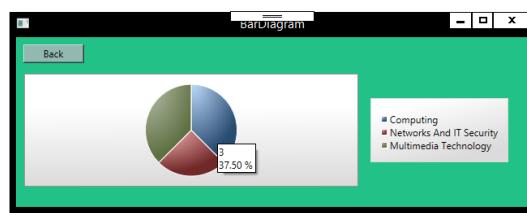


Figure 19: Pie Chart

### 3. Journal Articles

- **Journal of Electrical and Computer Engineering**

A new type of student information management system is designed to implement student information identification and management based on fingerprint identification. In order to ensure the security of data transmission, this paper proposes a data encryption method based on an improved AES algorithm. A new -box is cleverly designed, which can significantly reduce the encryption time by improving Byte Sub, Shift Row, and Mix Column in the round transformation of the traditional AES algorithm with the process of look-up table. Experimental results show that the proposed algorithm can significantly improve the encryption time compared with the traditional AES algorithm (Yang, 2017).



Figure 20: *Journal of Electrical and Computer Engineering*

- Design and implementation of college student's entrepreneurship management system based on B/S structure**

Objective: The rapid economic development of our country, enrolment expansion of colleges and universities, college graduate's employment difficult. Develop a set of college students' venture management system, can better solve the problem of university graduate's employment.

Methods: To describe the college students' innovative entrepreneurial management system development present situation, characteristics and requirements, through the relevant technology, the computer information technology applied to college students' entrepreneurial management, to implement the dynamic management of entrepreneurship students (Kai, Mar. 30, 2016).

The screenshot shows a Gale OneFile academic article page. At the top, there is a red header bar with the text 'GALE ONEFILE' and 'Informe Académico'. Below the header, the title of the article is displayed: 'Design and implementation of college students' entrepreneurship management system based on B/S structure'. The article details are as follows:

- Author:** Zhao Kai
- Date:** Mar. 30, 2016
- From:** RISTI (Revista Iberica de Sistemas e Tecnologias de Informacao) (Issue 179)
- Publisher:** AISTI (Iberian Association for Information Systems and Technologies)
- Document Type:** Article
- Length:** 3,364 words

The main content area includes an abstract section describing the objective, methods, and results of the research. On the right side, there is an 'EXPLORE' sidebar with a message about viewing a summary page from the Gale OneFile: Informe Académico database, and a 'View full article' button.

Figure 21: Student management system

- Web Based Student Information Management System**

Student Information Management System (SIMS) provides a simple interface for maintenance of student information. It can be used by educational institutes or colleges to maintain the records of students easily. The creation and management of accurate, up-to-date information regarding a students' academic career is critically important in the university as well as colleges. Student information system deals with all kind of student details, academic related reports, college details, curriculum, batch details, placement details and other resource related details too. It tracks all the details of a student from the day one to the end of the course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters, years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result and all these will be available through a secure, online interface embedded in the college's website (S.R.Bharamagoudar, 2010).

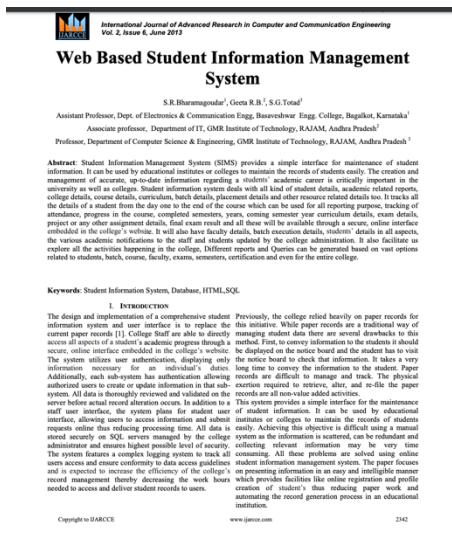


Figure 22: Web Based Student Information Management System

## 4. System Architecture

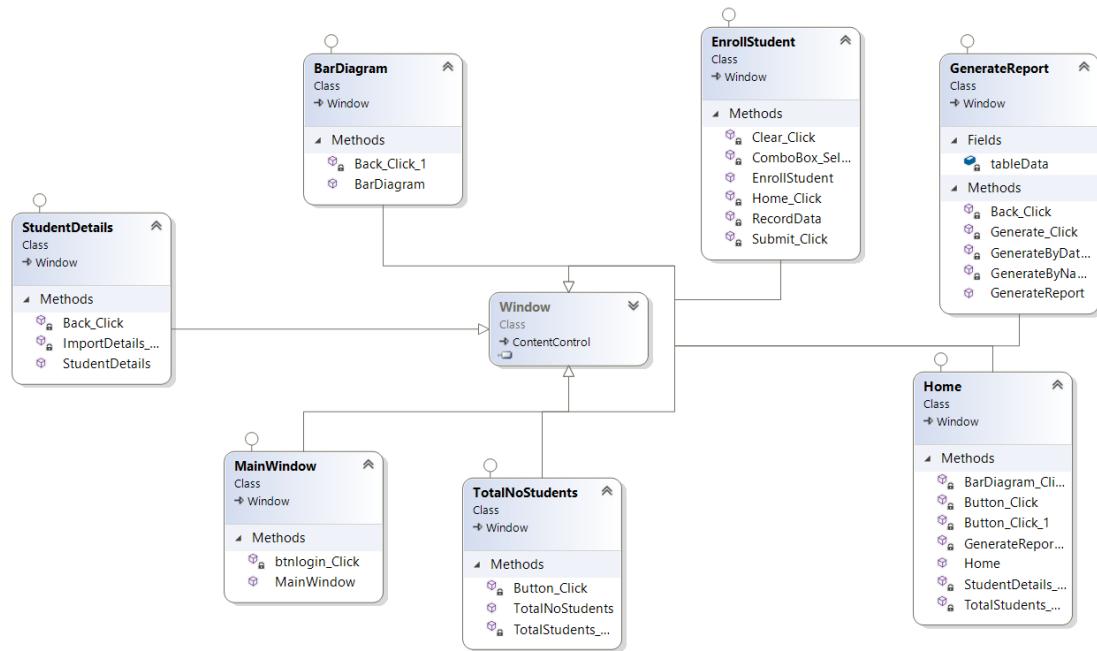


Figure 23: Class Diagram

## 4.1 Individual Diagram

- **BarDiagram**

Methods	Description	Diagram
<b>Back_Click_1</b>	Clears the data of the form.	
<b>BarDiagram</b>	Shows the chart.	<pre> classDiagram     class BarDiagram {         &lt;&lt;Class&gt;&gt;         &lt;&lt;Window&gt;&gt;         &lt;&lt;Methods&gt;&gt;         &amp;lt;&amp;gt; Back_Click_1         &amp;lt;&amp;gt; BarDiagram     }   </pre>

Table 1:Bar Diagram

- **StudentDetails**

Methods	Description	Diagram
<b>Back_Click</b>	Goes back to home window.	
<b>ImportDetails</b>	Helps to import the CSV file.	<pre> classDiagram     class StudentDetails {         &lt;&lt;Class&gt;&gt;         &lt;&lt;Window&gt;&gt;         &lt;&lt;Methods&gt;&gt;         &amp;lt;&amp;gt; Back_Click         &amp;lt;&amp;gt; ImportDetails_...         &amp;lt;&amp;gt; StudentDetails     }   </pre>

Table 2:Student Details

- **EnrollStudent**

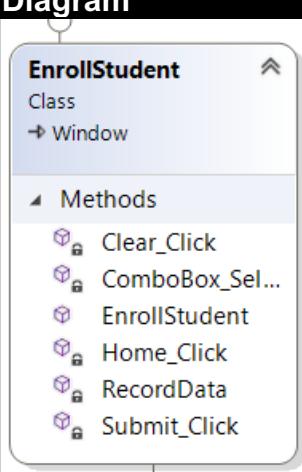
Methods	Description	Diagram
<b>Clear_Click</b>	Clears the data of the form.	
<b>EnrollStudent</b>	Helps to enrol data	
<b>Home_Click</b>	Goes back to home	
<b>Submit_Click</b>	Helps to submit data.	

Table 3:Enroll Students

- **Home**

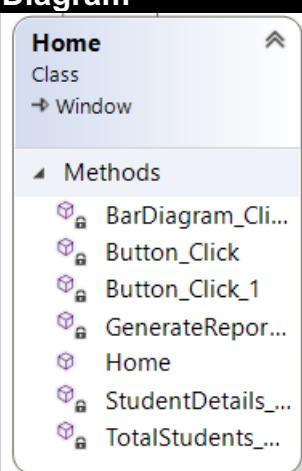
Methods	Description	Diagram
<b>GenerateReport</b>	Goes to generate the report window and helps to generate report.	
<b>BarDiagram</b>		
<b>Home</b>	Goes to home Window.	
<b>StudentDetails</b>	Shows Student Details	
<b>TotalStudents</b>	Shows total students.	

Table 4: Home

- **TotalNoStudents**

Methods	Description	Diagram
<b>TotalNoStudents</b>	Shows total number of students.	<pre> classDiagram     class TotalNoStudents {         &lt;&lt;TotalNoStudents&gt;&gt;         &lt;&lt;Class&gt;&gt;         &lt;&lt;Window&gt;&gt;         &lt;&lt;Methods&gt;&gt;         &amp;gt; Button_Click         &amp;gt; TotalNoStudents         &amp;gt; TotalStudents_...     }   </pre>

Table 5: Total No. Students

- **MainWindow**

Methods	Description	Diagram
<b>Btnlogin_Click</b>	Helps the user to login.	<pre> classDiagram     class MainWindow {         &lt;&lt;MainWindow&gt;&gt;         &lt;&lt;Class&gt;&gt;         &lt;&lt;Window&gt;&gt;         &lt;&lt;Methods&gt;&gt;         &amp;gt; btnlogin_Click         &amp;gt; MainWindow     }   </pre>

Table 6: Main Window

- **GenerateReport**

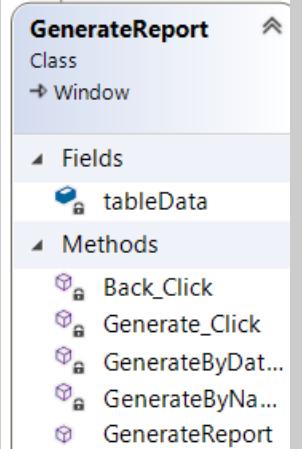
Methods	Description	Diagram
<b>Back_Click</b>	Goes back to home window.	
<b>GenerateByDate</b>	Helps to generate the report according to date.	
<b>GenerateByName</b>	Helps to generate the report by name.	
<b>GenerateReport</b>	Helps to generate all the report.	 <pre> classDiagram     class GenerateReport {         &lt;&lt;Window&gt;&gt;         tableData         &lt;&lt;Methods&gt;&gt;         &lt;&lt;Back_Click&gt;&gt;         &lt;&lt;Generate_Click&gt;&gt;         &lt;&lt;GenerateByDate&gt;&gt;         &lt;&lt;GenerateByName&gt;&gt;         &lt;&lt;GenerateReport&gt;&gt;     }   </pre>

Table 7: Generate Report

## 4.2 Flow-Chart

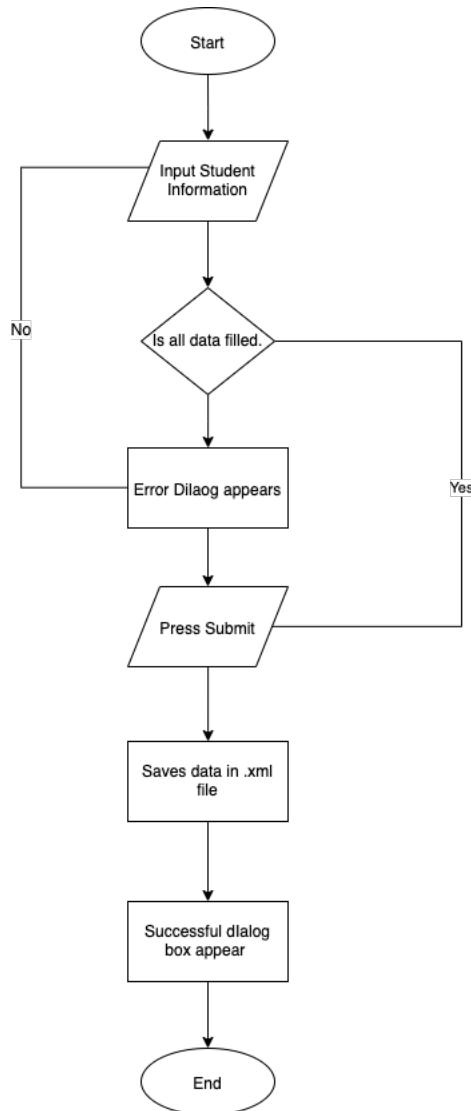


Figure 24: Flow-Chart

## 5. Sorting Algorithm

A sorting algorithm is a method for reorganizing a large number of items into a specific order, such as alphabetical, highest-to-lowest value or shortest-to-longest distance. Sorting algorithms take lists of items as input data, perform specific operations on those lists and deliver ordered arrays as output. To develop this form bubble sort algorithm is used.

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.

- **Steps of Bubble Sort Algorithm:**

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



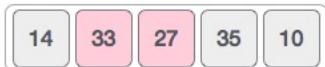
Bubble sort starts with very first two elements, comparing them to check which one is greater.



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 –



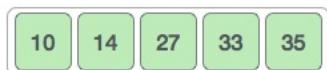
To be precise, we are now showing how an array should look like after each iteration. After the second iteration, it should look like this –



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



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



Now we should look into some practical aspects of bubble sort (Point, 2020).

## 6. Conclusion

This is the first coursework of Application Development. In this coursework I have used Visual Studio 2019 to complete the project. I was quite familiar with the Visual Studio and C# language. But I was not familiar with the WPF framework I had never used it before so it was quite difficult for me to do at first but after doing many researches I was able to complete the given task. WPF was totally new for me. But now I became familiar with it too.

## 7. Bibliography

Anon., 2020. *ZOHO Creator*. [Online]

Available at: <https://www.zoho.com/creator/application-development/>  
[Accessed 1 1 2020].

Kai, Z., Mar. 30, 2016. Design and implementation of college students' entrepreneurship management system based on B/S structure. *Design and implementation of college students' entrepreneurship management system based on B/S structure*.

Microsoft, 2020. *Microsoft*. [Online]

Available at: <https://docs.microsoft.com/en-us/visualstudio/designers/getting-started-with-wpf?view=vs-2019>  
[Accessed 1 1 2020].

Point, T., 2020. *Tutorials Point*. [Online]

Available at:

[https://www.tutorialspoint.com/data\\_structures\\_algorithms/bubble\\_sort\\_algorithm.htm](https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm)

[Accessed 2 1 2020].

S.R.Bharamagoudar, 2010. Web Based Student Information Management System. *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2( Issue 6, June 2013), p. 7.

Yang, P., 2017. Journal of Electrical and Computer Engineering. *Journal of Electrical and Computer Engineering*, Volume 2017(2020), p. 6 pages.

## 8. Appendix

### ➤ MainWindow (Login Page)

```
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 btnlogin_Click(object sender, RoutedEventArgs e)
        {
```

```
{  
    if (Username.Text.Trim() == "" && Password.Password.Trim()  
== "")  
    {  
        MessageBox.Show("Invalid Username or Password");  
    }  
    else  
    {  
        if      (Username.Text.Trim()      ==      "admin"      &&  
Password.Password.Trim() == "admin")  
        {  
            Home homeWindow = new Home();  
            homeWindow.Show();  
            this.Close();  
  
        }  
        else  
        {  
            MessageBox.Show("Invalid Username or Password");  
        }  
    }  
}  
}
```

➤ **Home**

```
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 Coursework

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

```
        private void Button_Click(object sender, RoutedEventArgs e)  
        {  
            EnrollStudent enrollStudent = new EnrollStudent();  
            enrollStudent.Show();  
            this.Close();  
        }
```

```
}

private void Button_Click_1(object sender, RoutedEventArgs e)
{
    StudentDetails student = new StudentDetails();
    student.Show();
    this.Close();
}

private void GenerateReport_Click(object sender,
RoutedEventArgs e)
{
    GenerateReport generate = new GenerateReport();
    generate.Show();
    this.Close();
}

private void TotalStudents_Click(object sender, RoutedEventArgs e)
{
    TotalNoStudents totalNoStudents = new TotalNoStudents();
    totalNoStudents.Show();
    this.Close();
}

private void BarDiagram_Click(object sender, RoutedEventArgs e)
{
    BarDiagram barDiagram = new BarDiagram();
    barDiagram.Show();
    this.Close();
}
```

```
private void StudentDetails_Click(object sender, RoutedEventArgs  
e)  
{  
    StudentDetails student = new StudentDetails();  
    student.Show();  
    this.Close();  
  
}  
}  
}
```

➤ **ImportStudentDetails**

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

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

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

```
try
{
    var dataSet = new DataSet();
    dataSet.ReadXml(@"StudentDetails.xml");
    Microsoft.Win32.OpenFileDialog openFileDlg = new
    Microsoft.Win32.OpenFileDialog();

    if (openFileDlg.ShowDialog() == true)
    {

        string filePath = openFileDlg.FileName;
        //read all std from file code copy
        using (var read = new StreamReader(filePath))
        {
            read.ReadLine();
            while (!read.EndOfStream)
            {
                var line = read.ReadLine();
                var values = line.Split(',');
                var newRow = dataSet.Tables["Information"].NewRow();
                newRow["StudentRegistrationDate"] = values[0];
                newRow["StudentID"] = values[1];
                newRow["StudentName"] = values[2];
                newRow["StudentAddress"] = values[3];
                newRow["StudentContact"] = values[4];
                newRow["StudentCourse"] = values[5];
                dataSet.Tables["Information"].Rows.Add(newRow);

                dataSet.WriteXml(@"StudentDetails.xml");
            }
        }
    }
}
```

```
        MessageBox.Show("Successfully Imported and Recorded  
all Student Details.", "Import Sucessfull!!!", MessageBoxButton.OK,  
MessageBoxImage.Asterisk);  
    }  
}  
catch (Exception ex)  
{  
    MessageBox.Show(ex.Message);  
}  
  
}  
  
private void Back_Click(object sender, RoutedEventArgs e)  
{  
    Home home = new Home();  
    home.Show();  
    Close();  
}  
}  
}
```

➤ **Information**

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

```
namespace Coursework
```

```
{  
    public class Information  
    {  
        private string txtData1;  
        private string txtData2;  
        private string txtData3;  
        private string txtData4;  
        private string txtData5;  
        private string txtData6;
```

```
        public string StudentRegistrationDate
```

```
{  
    get { return txtData1; }  
    set { txtData1 = value; }  
}
```

```
        public string StudentID
```

```
{  
    get { return txtData2; }  
    set { txtData2 = value; }  
}
```

```
        public string StudentName
```

```
{
```

```
        get { return txtData3; }
        set { txtData3 = value; }

    }

    public string StudentAddress
    {
        get { return txtData4; }
        set { txtData4 = value; }
    }

    public string StudentContact
    {
        get { return txtData5; }
        set { txtData5 = value; }
    }

    public string StudentCourse
    {
        get { return txtData6; }
        set { txtData6 = value; }
    }
}
```

➤ **GenerateReport**

```
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;
```

```
namespace Coursework
{
    /// <summary>
    /// Interaction logic for GenerateReport.xaml
    /// </summary>
```

```
public partial class GenerateReport : Window
{
    public GenerateReport()
    {
        InitializeComponent();
    }

    private void Back_Click(object sender, RoutedEventArgs e)
    {
        Home home = new Home();
```

```
        home.Show();
        this.Close();
    }

    DataTable tableData;
    private void GenerateByDate_Click(object sender,
RoutedEventArgs e)
{
    try
    {
        if (System.IO.File.Exists(@"StudentDetails.xml"))
        {

            var dataSet = new DataSet();
            dataSet.ReadXml(@"StudentDetails.xml");

            //DataTable StudentDT = new DataTable("dt");
            tableData = dataSet.Tables[0];
            tableData.DefaultView.Sort = "StudentRegistrationDate
ASC";
            ReportDataGrid.Items.Refresh();
            ReportDataGrid.ItemsSource = tableData.DefaultView;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Generate_Click(object sender, RoutedEventArgs e)
{
    try
```

```
{  
    DataSet ds = new DataSet();  
    ds.ReadXml("StudentDetails.xml");  
    DataView dv = new DataView();  
    dv = ds.Tables[0].DefaultView;  
    this.ReportDataGrid.ItemsSource = dv;  
  
    MessageBox.Show("Student details report is generated.",  
    "Report Generated!!!", MessageBoxButtons.OK,  
    MessageBoxIcon.Asterisk);  
}  
catch (Exception ex)  
{  
    MessageBox.Show(ex.Message);  
}  
}  
  
private void GenerateByName_Click_1(object sender,  
RoutedEventArgs e)  
{  
    try  
    {  
        if (System.IO.File.Exists(@"StudentDetails.xml"))  
        {  
  
            var dataSet = new DataSet();  
            dataSet.ReadXml(@"StudentDetails.xml");  
  
            //DataTable StudentDT = new DataTable("dt");  
            tableData = dataSet.Tables[0];  
            tableData.DefaultView.Sort = "StudentName ASC";  
            ReportDataGrid.Items.Refresh();  
            ReportDataGrid.ItemsSource = tableData.DefaultView;  
        }  
    }
```

```
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
}  
}
```

➤ **EnrollStudent**

```
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.Xml.Serialization;
using System.IO;
```

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

```
static void RecordData(Information obj, string filename)
{
    XmlSerializer           xmlSerializer      = new
    XmlSerializer(typeof(List<Information>));
```

```

List<Information> list = null;
try
{
    using (Stream s = File.OpenRead(filename))
    {
        list = xmlSerializer.Deserialize(s) as List<Information>;
    }
}
catch
{
    list = new List<Information>();
}
list.Add(obj);
using (Stream s = File.OpenWrite(filename))
{
    xmlSerializer.Serialize(s, list);
}
}

```

```

private void Submit_Click(object sender, RoutedEventArgs e)
{
    if (StudentId.Text == "" || StudentName.Text == "" || Address.Text
== "" || Contact.Text == "" || CourseEnroll.Text == "" || Date.Text == "")
    {
        MessageBox.Show("Enter a valid input field.", "Message",
MessageBoxButton.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        try
        {
            Information information = new Information();
            information.StudentID = StudentId.Text;

```

```
        information.StudentRegistrationDate =  
        Date.SelectedDate.Value.ToString("yyyy-MM-dd");  
        information.StudentName = StudentName.Text;  
        information.StudentAddress = Address.Text;  
        information.StudentContact = Contact.Text;  
        information.StudentCourse = CourseEnroll.Text;  
        RecordData(information, "StudentDetails.xml");  
  
        MessageBox.Show("Student is successfully enrolled.", "Enroll  
Successfull!!!", MessageBoxButton.OK, MessageBoxImage.Asterisk);  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
}  
  
}  
  
private void Home_Click(object sender, RoutedEventArgs e)  
{  
    Home home = new Home();  
    home.Show();  
    this.Close();  
}  
  
private void Clear_Click(object sender, RoutedEventArgs e)  
{  
    StudentId.Text = "";  
    StudentName.Text = "";  
    Contact.Text = "";
```

```
        Address.Text = "";  
    }  
  
    private void ComboBox_SelectionChanged(object sender,  
SelectionChangedEventArgs e)  
    {  
  
    }  
}
```

➤ **TotalNoStudents**

```
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.Xml.Serialization;
```

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

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            Home home = new Home();
```

```
        home.Show();
        this.Close();
    }

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

    dataSet.ReadXml(@"E:\\Coursework\\bin\\Debug\\StudentDetails.xml")
;

    DataTable studentDataReport = dataSet.Tables[0];

    int total_Computing = 0;
    int total_Network = 0;
    int total_Multimedia = 0;

    DataTable dataTable = new DataTable("table");
    dataTable.Columns.Add("Course Enroll", typeof(String));
    dataTable.Columns.Add("Total Students", typeof(int));

    for (int i = 0; i < studentDataReport.Rows.Count; i++)
    {
        String col =
studentDataReport.Rows[i]["StudentCourse"].ToString();
        if (col == "Computing")
        {
            total_Computing++;
        }
        else if (col.Equals("Networks And IT Security"))
        {
            total_Network++;
        }
    }
}
```

```
        }

        else if (col.Equals ("Multimedia Technology"))

        {

            total_Multimedia++;

        }

    }

    dataTable.Rows.Add("Computing", total_Computing);

    dataTable.Rows.Add("Networks      And      IT      Security",

total_Network);

    dataTable.Rows.Add("Multimedia           Technology",

total_Multimedia);

    Console.WriteLine(dataTable.Rows.Count);

    StudentDataGrid.DataContext = dataTable.DefaultView;

}

}

}
```

➤ **XmlDetails**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml.Serialization;
using System.IO;

namespace Coursework
{
    public class XmlDetails
    {
        public static void RecordData(object obj, string filename)
        {
            XmlSerializer ser = new XmlSerializer(obj.GetType());
            FileStream stream = new FileStream(filename,
FileMode.Append, FileAccess.Write);
            ser.Serialize(stream, obj);
            stream.Close();
        }
    }
}
```