Informatics College Pokhara

Submitted By:
Student Name: Ajay Shrestha
London Met ID: 17031396
Group: L3C2
Date: 01-Jan-2020

Submitted To:
Mr. Ishwor Sapkota
Application Development

Abstract

This is an individual coursework for the module "Application Development" for Student Management System which is developed using Visual Studio Platform using C# language. The coursework is released in the week 5 and it is supposed to be submitted in the week 11.

With the great contribution of Mr. Ishwor Sapkota, the coursework was completed within the time frame.

Table of Contents

List of Figure

1. Introduction

The system designed in this project is known as Student Information System which is also designed and implemented in C# - desktop application. The main focus of this system is to keep track of the student's details, program enroll and registration date. The application allows the user to input the student personal detail including registration date so that a system can generate a weekly enrolment report of the system. The system is highly designed, developed and test under various set of conditions. In addition, there is a features to view daily or weekly table report and chart.

1.1 Current Scenario

There are abundance Schools/Colleges which is still capturing and storing student's record details locally where hard copies of files for every student is kept in office shelves, this seem to be tiresome and time consuming in case the registrar is looking of a particular student details or document. The problems facing the current manual system are data redundancy, difficult to update maintain, inconsistent data, insecurity, difficult to impose constraints on various data file and difficult to backup. Therefore, because of these drawbacks that Student Information System has been developed to address the problems catalogued above.

1.2 Proposed System

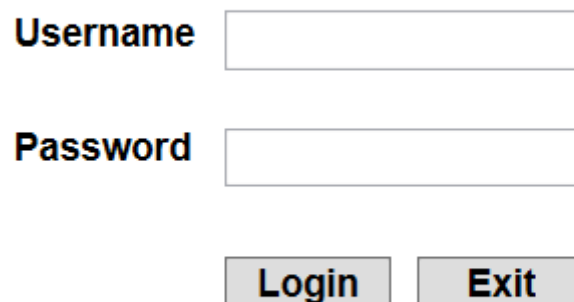The proposed system is intended to make life easy. The main purpose of the project is to build a student information system to facilitate easy access of student's records. The Student Information System will allow the registrar of different schools/colleges, login to edit, update/view student details record, weekly report, chart and vice-versa. It also enhances efficient management of student's information.

2. User Manual

There are screenshot below which will illustrate a user how to operate the system.

As the end user operates the system the initial screen will be the security screen. The username and password of the system, both is "admin". Only a valid username and password can provide access to the system. And also, if exit button is pressed then it will terminate the system.

Figure 1: Login Screen

After the successful login, the main screen of the system appears.

Student Personal Details

ID

Name

Address

Phone

Course Enroll

Registration Date   Select a date   [15]

| Save | Import | Display Report |

ID	Name	Address	Phone	Course Enroll	Registration Date	

Figure 2: Main Screen

The main screen provides three services: one is new entry of student details second one is to quick view of recent registered records and last one is to retrieve, chart, import CSV file, sort By date, sort by name etc.

Figure 3: Filling new registration

After filling all the required blanks, when save button is clicked, alert box is appeared with a message data saved successfully which means our entry is registered.

When data entry is done, if we click import button then below data grid/table will display the recent registered information.

Student Personal Details

ID	5
Name	Ishwor Sapkota
Address	Kathmandu, Nepal
Phone	9805832714
Course Enroll	Computing
Registration Date	8/2/2019

Save   Import   Display Report

ID	Name	Address	Phone	Course Enroll	Registration Date	
5	Ishwor Sapkota	Kathmandu, Nepa	9805832714	Computing	8/2/2019	

Figure 4: Import button function

Clicking the import button displays the recent registered record with all the details that were filled.

The last button "Display Report" in the main window shows the following window.
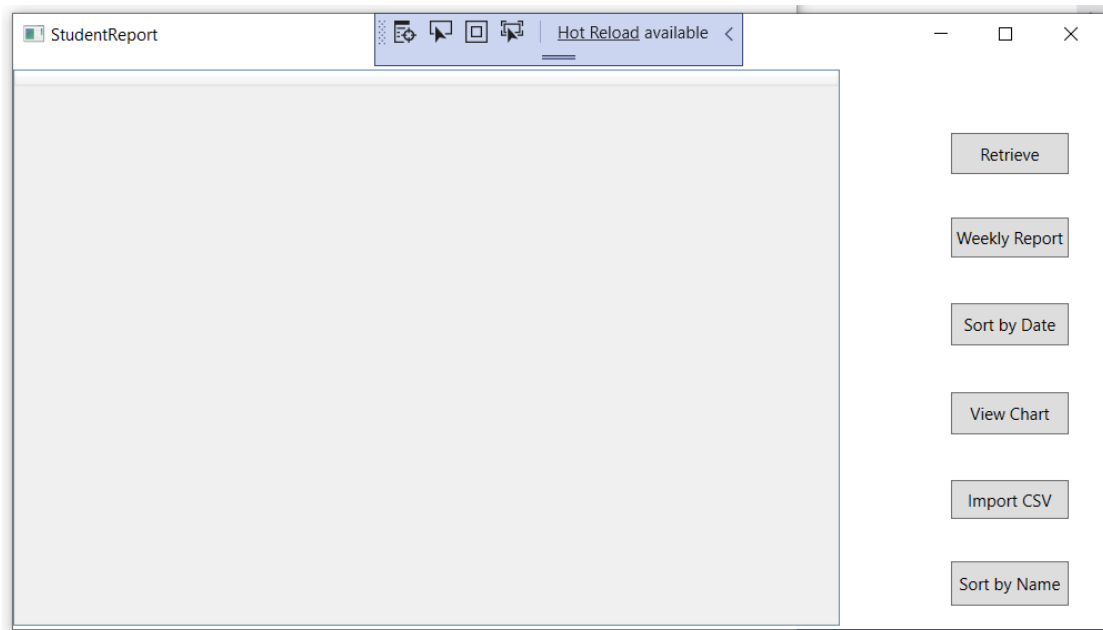

Figure 5: Display Report Window

This window shows the features button like Retrieve, Weekly Report, Sort by Date, View Chart, Import CSV and Sort by Name with data grid layout.
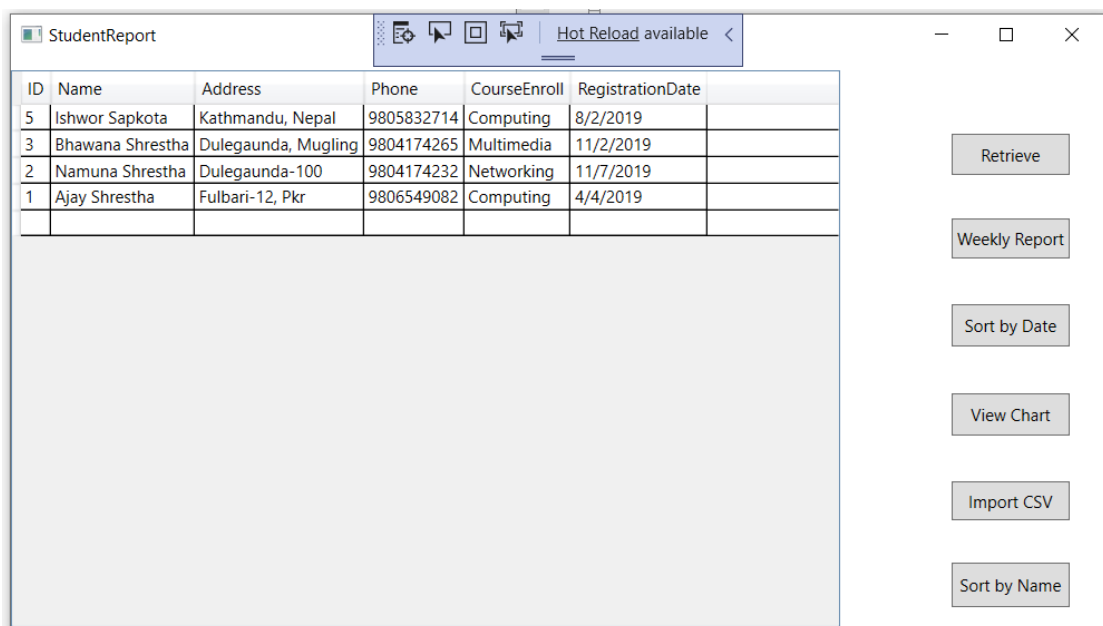
Figure 6: Retrieve function

When you press the Retrieve button, it displays the overall records inserted or registered from all time.
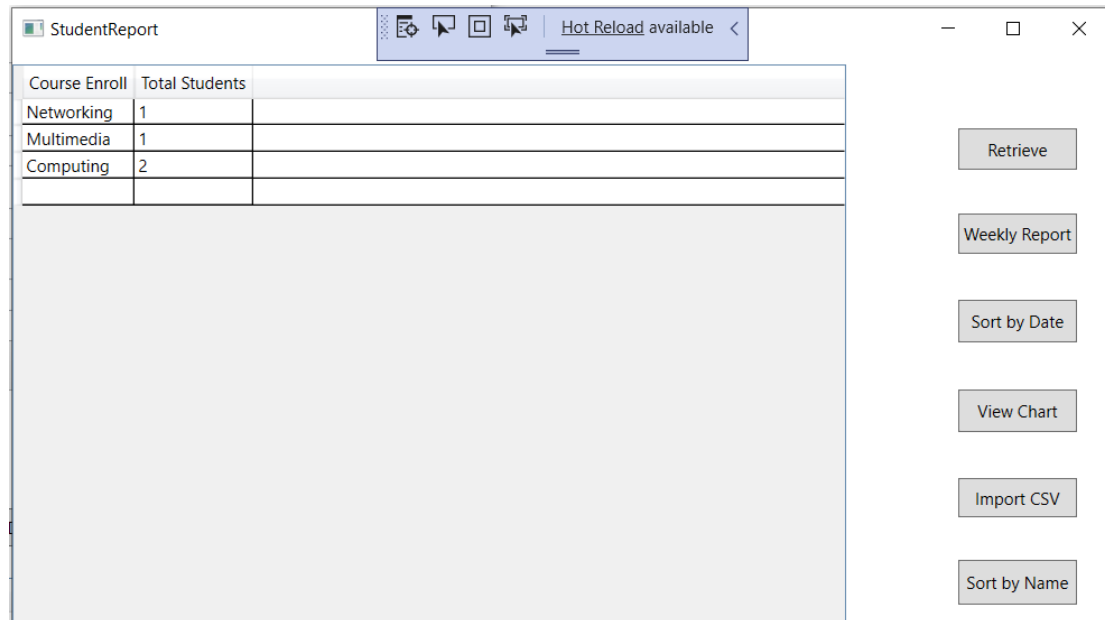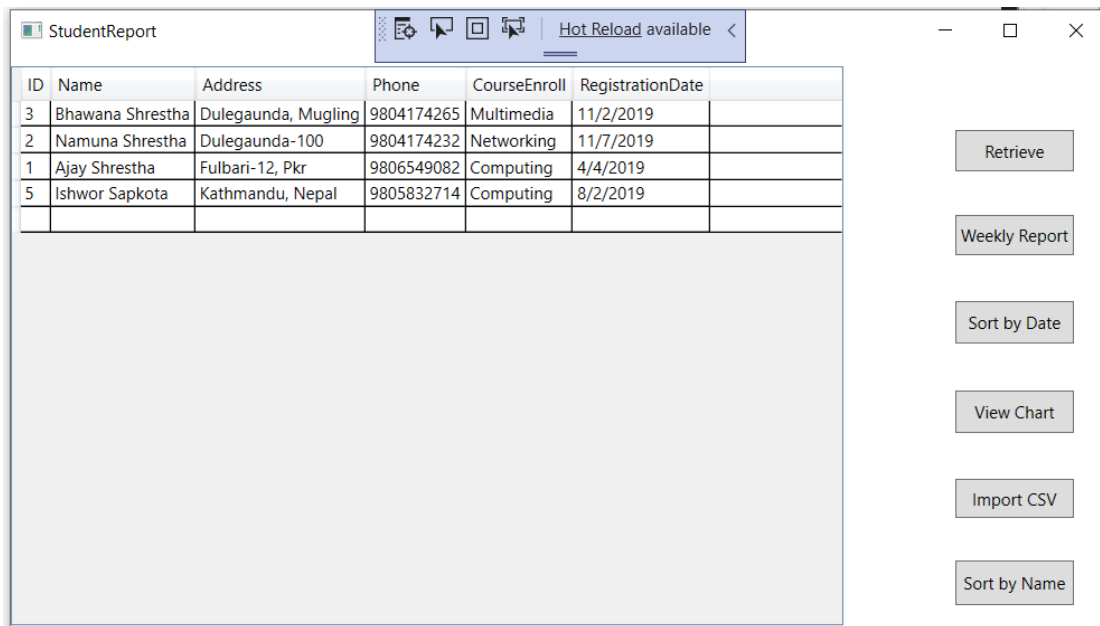
Figure 7: Weekly Report function

To see the weekly report of the student, all you have to do is close the student report window and go back to main window, press the back display report and click on weekly report button, it will show the accurate total number of students in each course enrol.
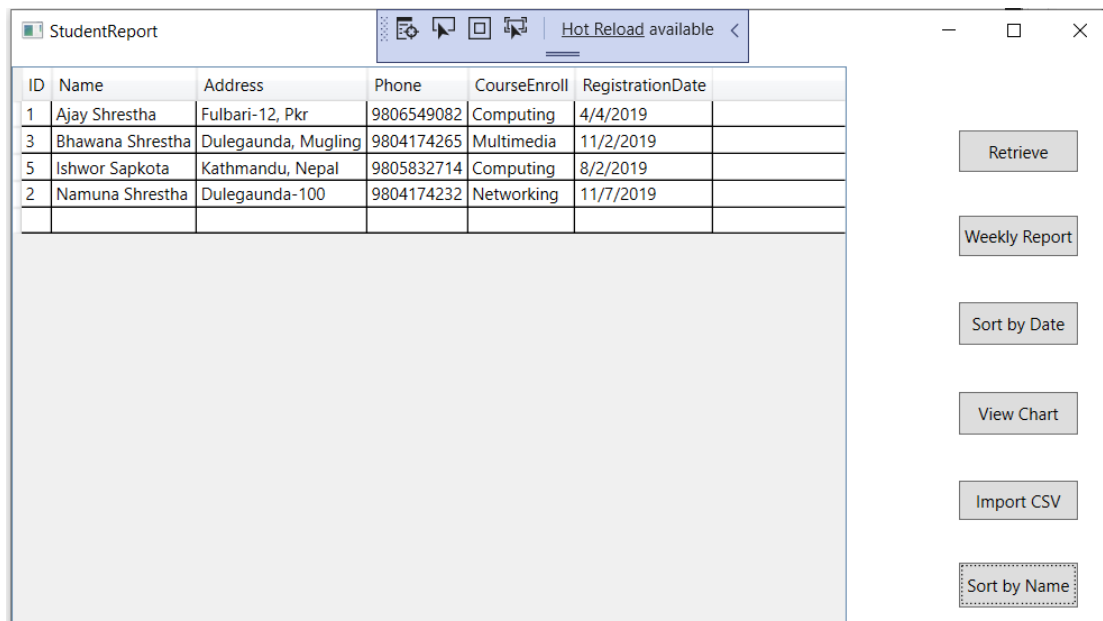
As shown in the figure 6, data are displayed in an unsorted format. So, to display in sorted format by registration date and name. A user should simply click on Sort by Date and Sort by Name as they want that list on which format.

Figure 8: Sort by Date Format

As the user click on Sort by Date button, all the details will be refreshed and it will sorted according to registration date.

Figure 9: Sort by Name Format

As the user click on Sort by Date button, all the details will be refreshed and it will sorted according to name basis.

To view the detail in a chart section, a user should simply click on View Chart and following screen will appear.
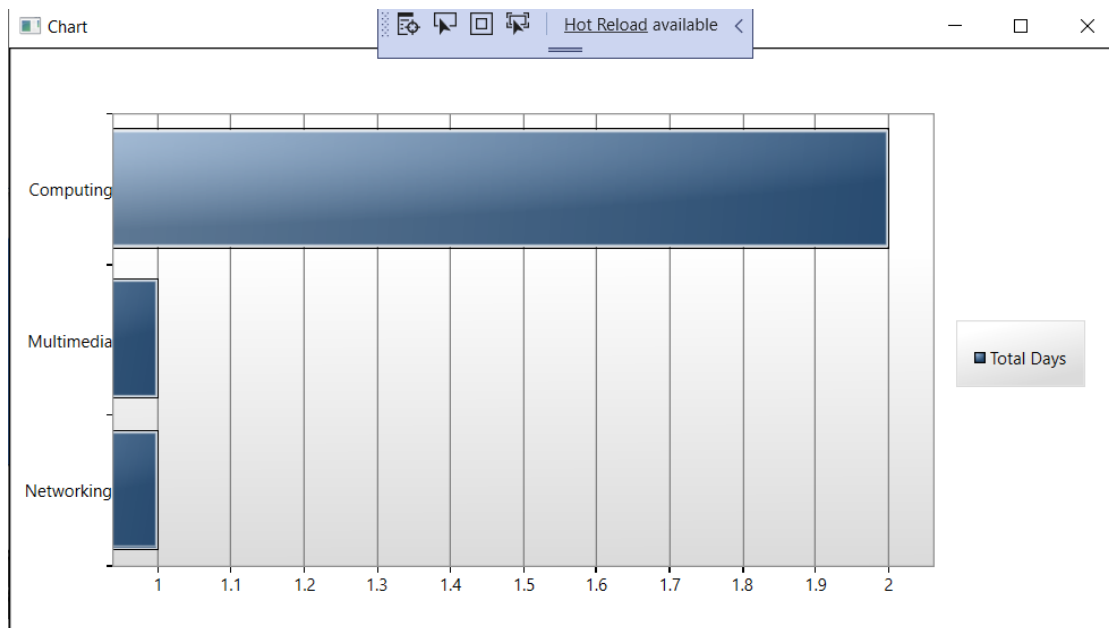
Figure 10: Chart Display of Report

A user can import the data from a CSV file into the grid as shown below.
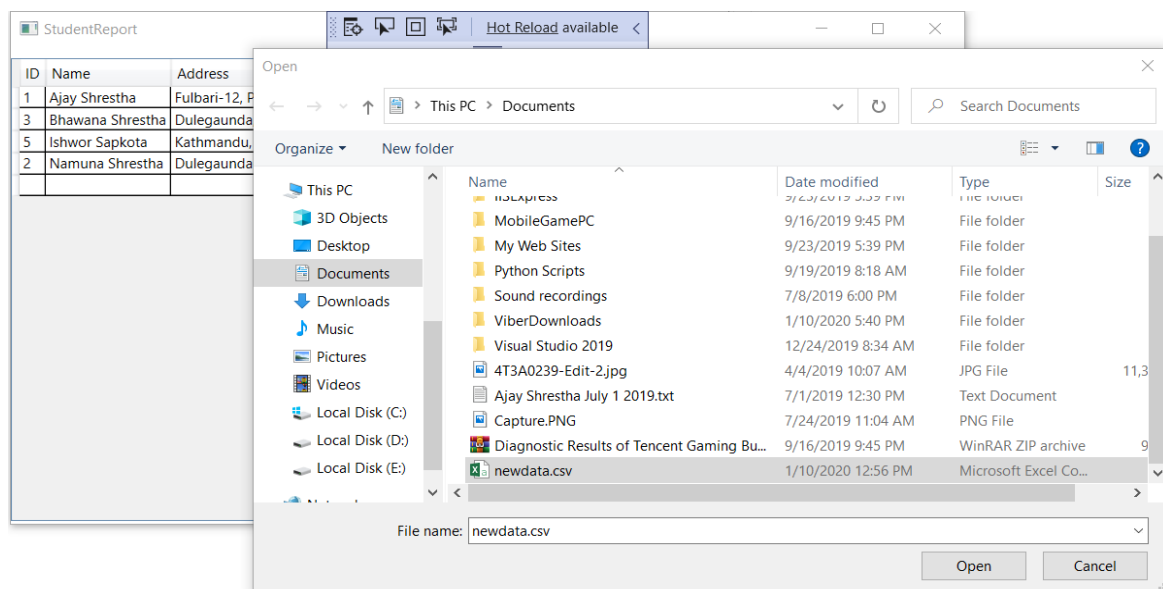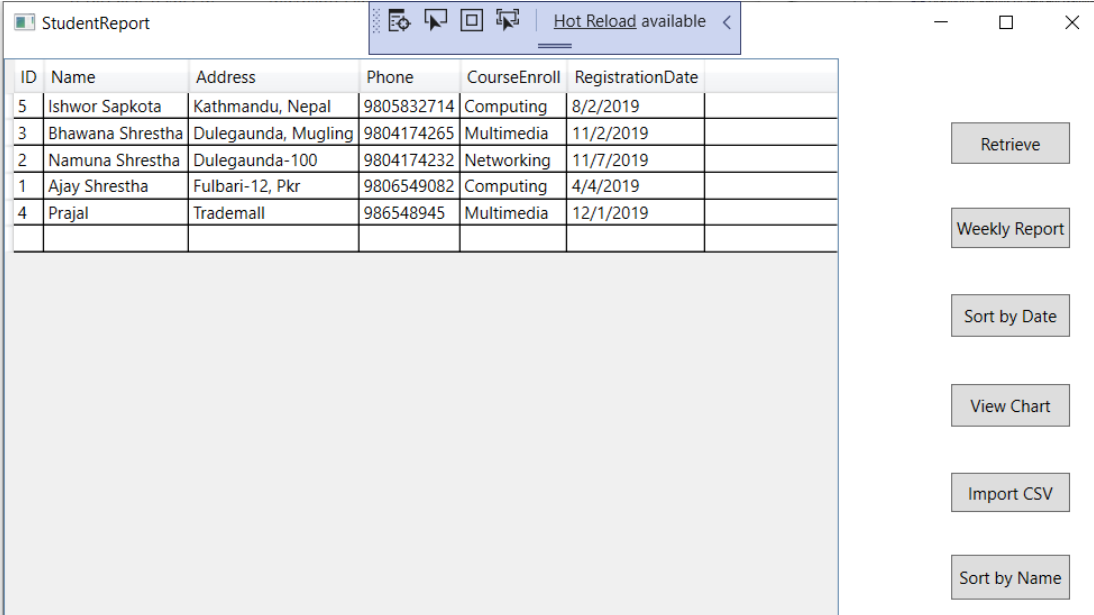
Figure 11: Import data from CSV

The data will be displayed in the grid as shown below. The last row has been added to the grid from CSV file.

Figure 12: Data imported from CSV file

Data imported from CSV file has been successfully exported to the XML file.

```xml
<?xml version="1.0" standalone="true"?>
<NewDataSet>
  - <StudentInfo>
        <ID>5</ID>
        <Name>Ishwor Sapkota</Name>
        <Address>Kathmandu, Nepal</Address>
        <Phone>9805832714</Phone>
        <CourseEnroll>Computing</CourseEnroll>
        <RegistrationDate>2019-08-02T00:00:00+05:45</RegistrationDate>
    </StudentInfo>
  - <StudentInfo>
        <ID>3</ID>
        <Name>Bhawana Shrestha</Name>
        <Address>Dulegaunda, Mugling</Address>
        <Phone>9804174265</Phone>
        <CourseEnroll>Multimedia</CourseEnroll>
        <RegistrationDate>2019-11-02T00:00:00+05:45</RegistrationDate>
    </StudentInfo>
  - <StudentInfo>
        <ID>2</ID>
        <Name>Namuna Shrestha</Name>
        <Address>Dulegaunda-100</Address>
        <Phone>9804174232</Phone>
        <CourseEnroll>Networking</CourseEnroll>
        <RegistrationDate>2019-11-07T00:00:00+05:45</RegistrationDate>
    </StudentInfo>
  - <StudentInfo>
        <ID>1</ID>
        <Name>Ajay Shrestha</Name>
        <Address>Fulbari-12, Pkr</Address>
        <Phone>9806549082</Phone>
        <CourseEnroll>Computing</CourseEnroll>
        <RegistrationDate>2019-04-04T00:00:00+05:45</RegistrationDate>
    </StudentInfo>
  - <StudentInfo>
        <ID>4</ID>
        <Name>Prajal</Name>
        <Address>Trademall</Address>
        <Phone>986548945</Phone>
        <CourseEnroll>Multimedia</CourseEnroll>
        <RegistrationDate>12/1/2019</RegistrationDate>
    </StudentInfo>
</NewDataSet>
```

Figure 13: CSV file data exported to Xml file

3. Journal Articles

a. 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 ByteSub, ShiftRow, and MixColumn 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 (Pengtao Yang, 2017).

b. The management and provision of information about the educational process is an essential part of effective management of the educational process in the institutes of higher education. In this paper the requirements of a reliable student management system are analyzed, formed a use-case model of student information management system, designed and implemented the architecture of the application. Regarding the implementation process, modern approaches were used to develop and deploy a reliable online application in cloud computing environments specifically (Alameri, 2017).

c. The mission of the Student Information Management system is to create an integrated information technology environment for students, HOD, faculty, staff and administration. Our goal is to focus on services and integration for end users. It is a web based self service environment for students, prospective students, and employees; an administrative transaction processing environment for yearly admissions; an informative environment for all levels of faculty and staff to do reporting, data extraction and information analysis.It is mainly useful for educational establishments to manage student data which also facilitates all individual associated information for easier navigation on daily basis. It provides capabilities for entering student test and other assessment scores, building student schedules, tracking student attendance and managing many other student-related data needs in a

college. Our easy-to-use, integrated college administration application would be used to reduce time spent on administrative tasks, as to concentrate on other skillful practical activities other than book worming. It can accept, process and generate reports at any given point of time accurately (Dipin Budhrani, 2018).

d. The study aimed to assess the contribution of Student Management Teams (SMTs) in a large undergraduate class and identify barriers and facilitators of the communication process and learning experience. SMTs were put into practice supported by an Online Learning Management System (OLMS: easyclass). It was hypothesized that students would positively evaluate their team participation, its role to the communication process and learning experience and there would be positive associations between the LMS, communication and learning experience. The results confirmed the hypotheses showing the effectiveness of SMTs, highlighting the use of OLMS and confirming the relationship between communications and learning experience satisfaction, also supported by qualitative data (Petkari, 2015).

4. System Architecture

Architecture Diagram
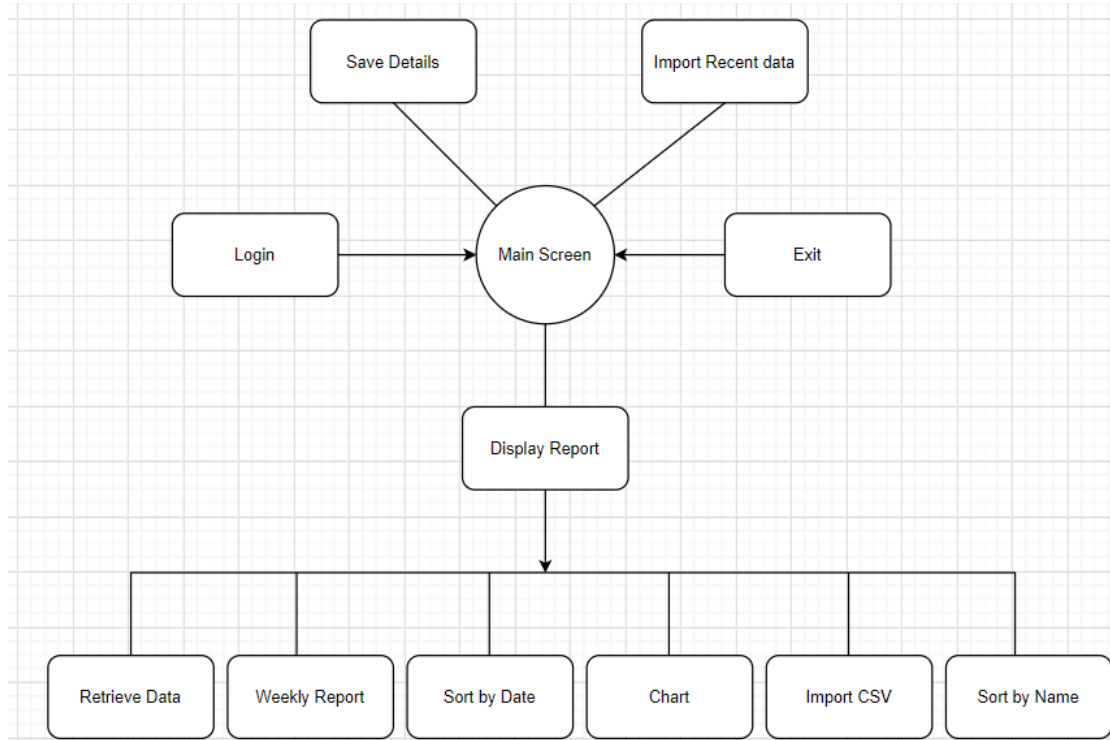
Figure 14: Architecture Diagram

Class Diagram
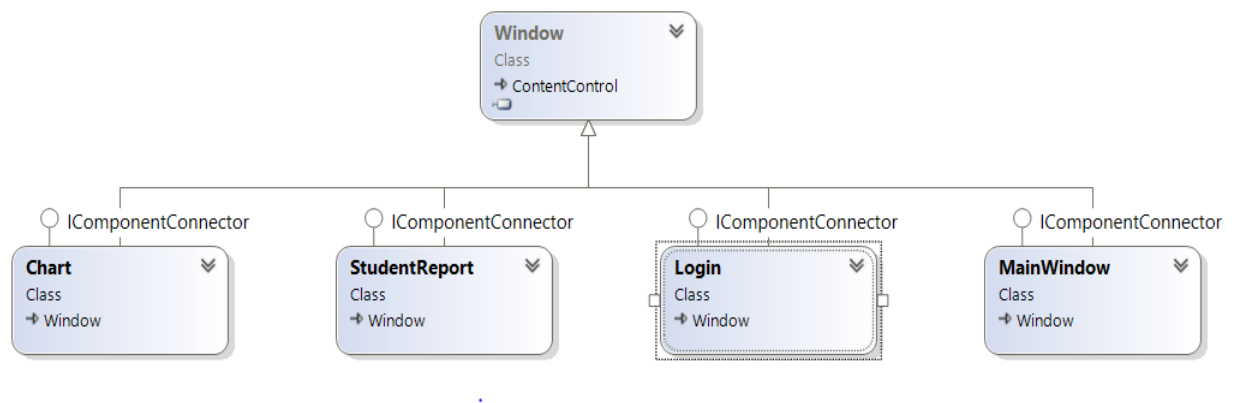
Figure 15: Class Diagram

Individual Diagram

1. Login

Methods	Description	Diagram
btn_exit_Click	Logout from the application.	
Button_Click	Check the username and password if its match or not.	
Login	If the username and password matches, it will move to Main Window.	

Figure 16: Login Individual Diagram
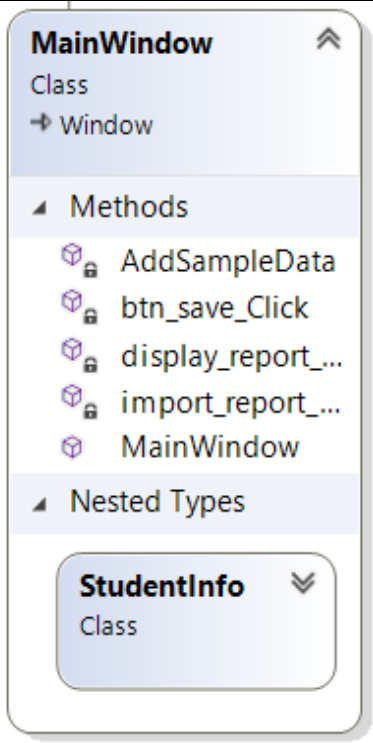
Methods	Description	Diagram
AddSampleData	To create new registration details.	**MainWindow** Class ➔ Window
btn_save_Click	It saves the data inserted in the required form.	◢ Methods
display_report_Click	It opens new window form in the application.	◢ Methods — AddSampleData, btn_save_Click, display_report_..., import_report_..., MainWindow
import_report_Click	It displays the recent details added to the system.	◢ Nested Types **StudentInfo** Class

Figure 17: Main Window Individual Diagram
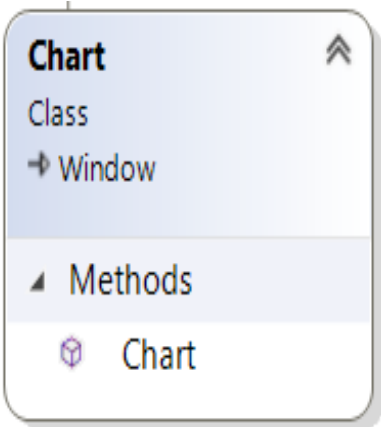
Methods	Description	Diagram
Chart	Chart window will appear after clicking view chart in Display Report window.	**Chart** Class ➔ Window ◢ Methods — Chart

Figure 18: Chart Individual Diagram
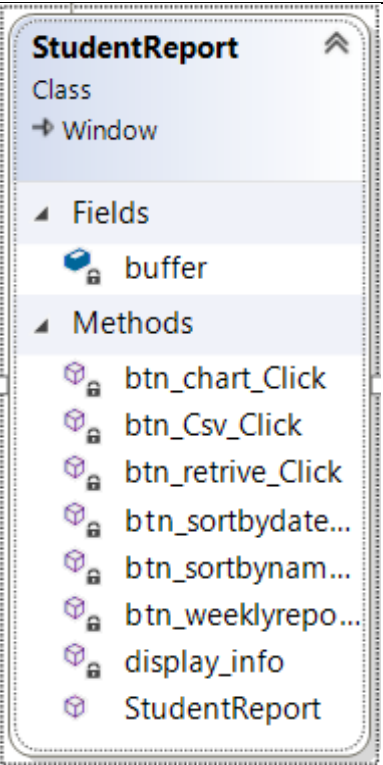
Methods	Description	Diagram
btn_chart_click	Displays chart of the report.	
btn_Csv_Click	Import csv file to the current report.	
btn_retrive_Click	Retrieve the data inserted in the registration form.	
btn_sortbydate_Click	Sort the registration date of the report in ascending order.	
btn_sortbyname_Click	Sort the name of the report in ascending order.	
btn_weeklyreport_Click	Displays the weekly report of the data in grid.	

Figure 19: Student Report Individual Diagram

Flowchart for Reports
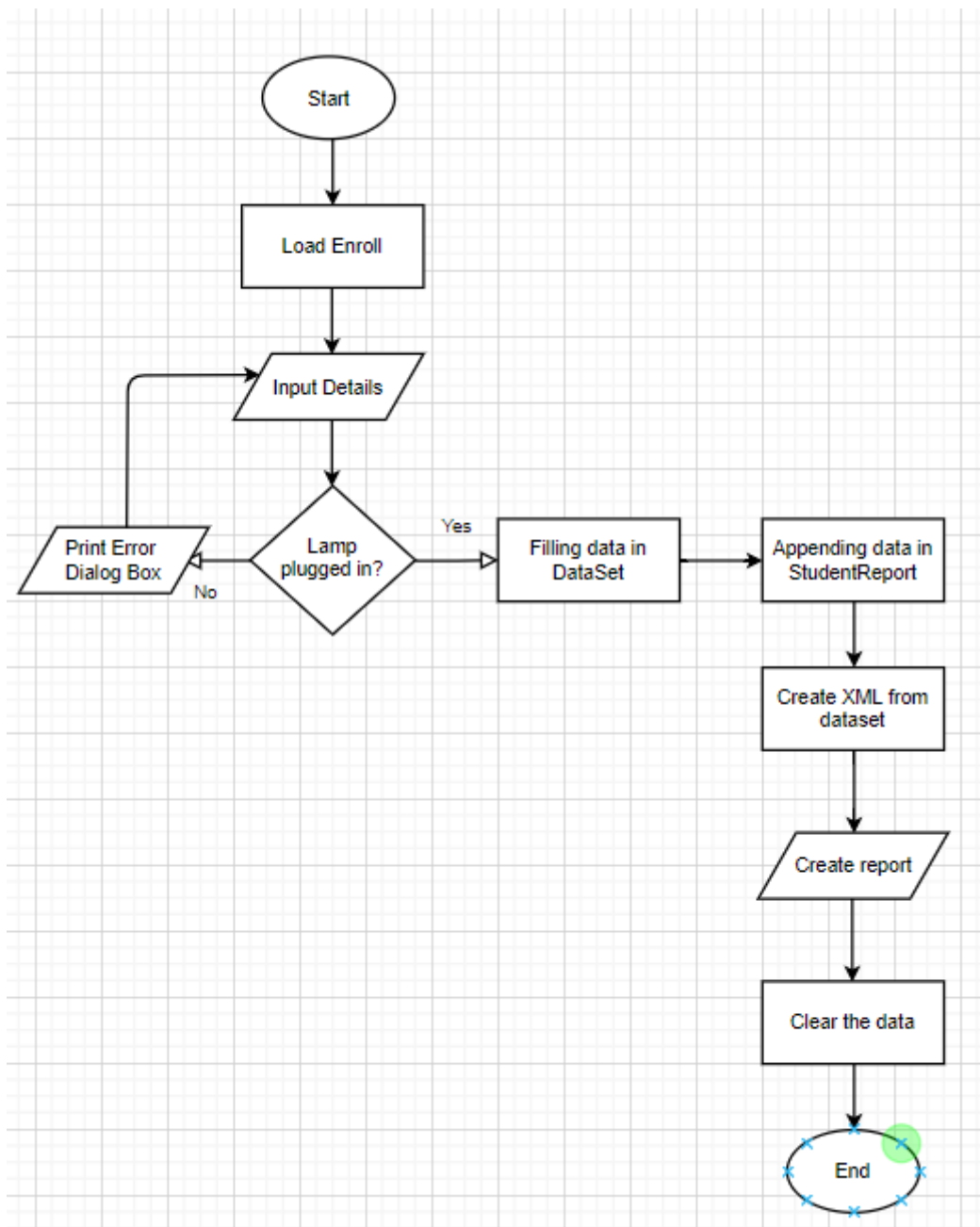
Figure 20: Flowchart of Enrol Students
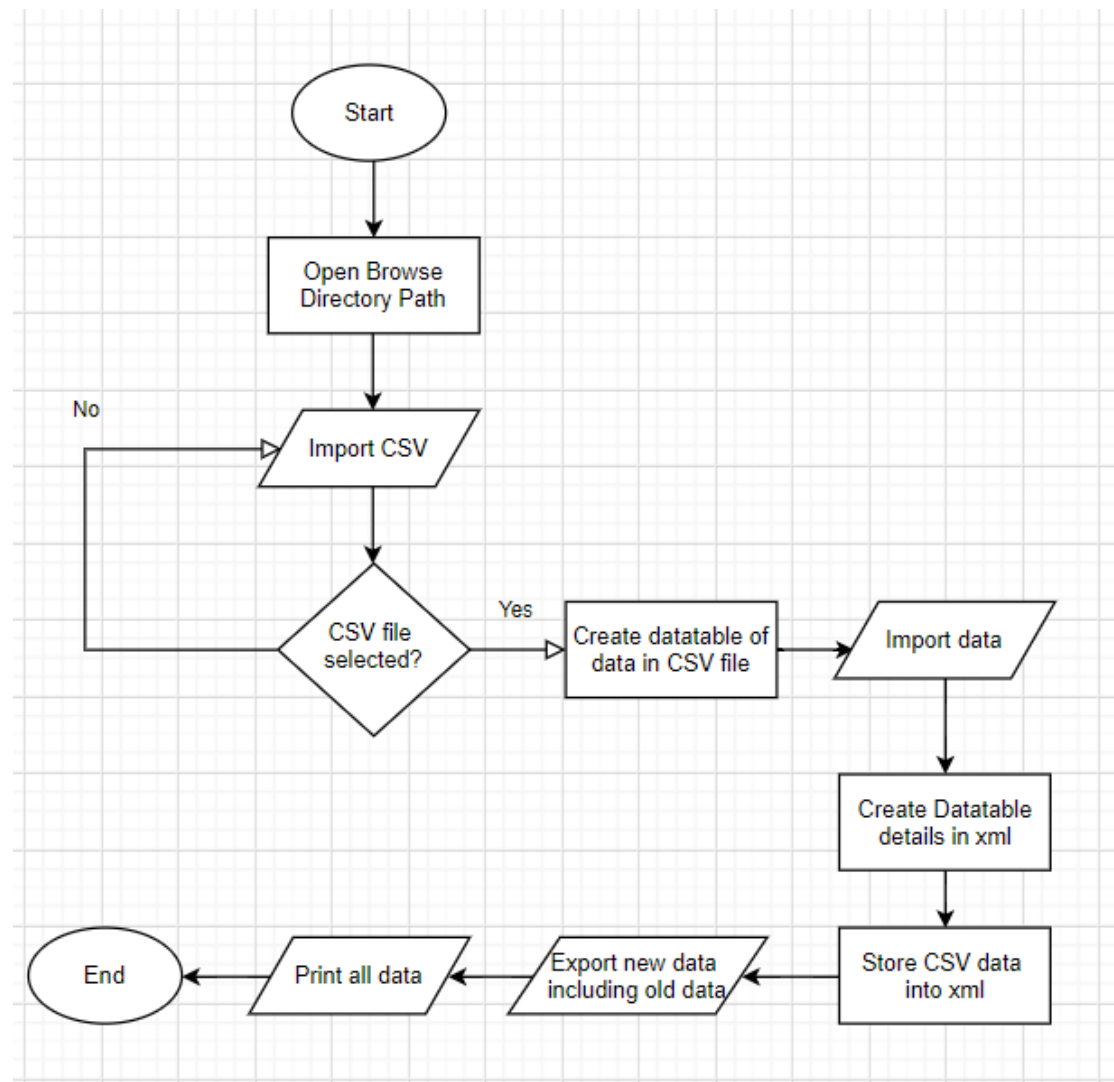
Figure 21: Import CSV flowchart

5. Sorting Algorithm

The sorting Algorithm used in Student Information System is bubble sorting 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 (Anon., 2020).

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.

Sorting takes place by stepping through all the elements one-by-one and comparing it with the adjacent element and swapping them if required.

Implementing Bubble Sort Algorithm

Following are the steps involved in bubble sort (for sorting a given array in ascending order):

1. Starting with the first element (index = 0), compare the current element with the next element of the array.
2. If the current element is greater than the next element of the array, swap them.
3. If the current element is less than the next element, move to the next element. Repeat Step 1.

Let's consider an array with values {5, 1, 6, 2, 4, 3}

Below, we have a pictorial representation of how bubble sort will sort the given array.

| 5>1 so interchange | 5 | 1 | 6 | 2 | 4 | 3 |

| 5<6 No swapping | 5 | 1 | 6 | 2 | 4 | 3 |

This is first insertion

| 6>2 so interchange | 1 | 5 | 6 | 2 | 4 | 3 |

| 6>4 so interchange | 1 | 5 | 2 | 6 | 4 | 3 |

similarly, after all the iterations, the array gets sorted

| 6>3 so interchange | 1 | 5 | 2 | 4 | 6 | 3 |

| | 1 | 5 | 2 | 4 | 3 | 6 |
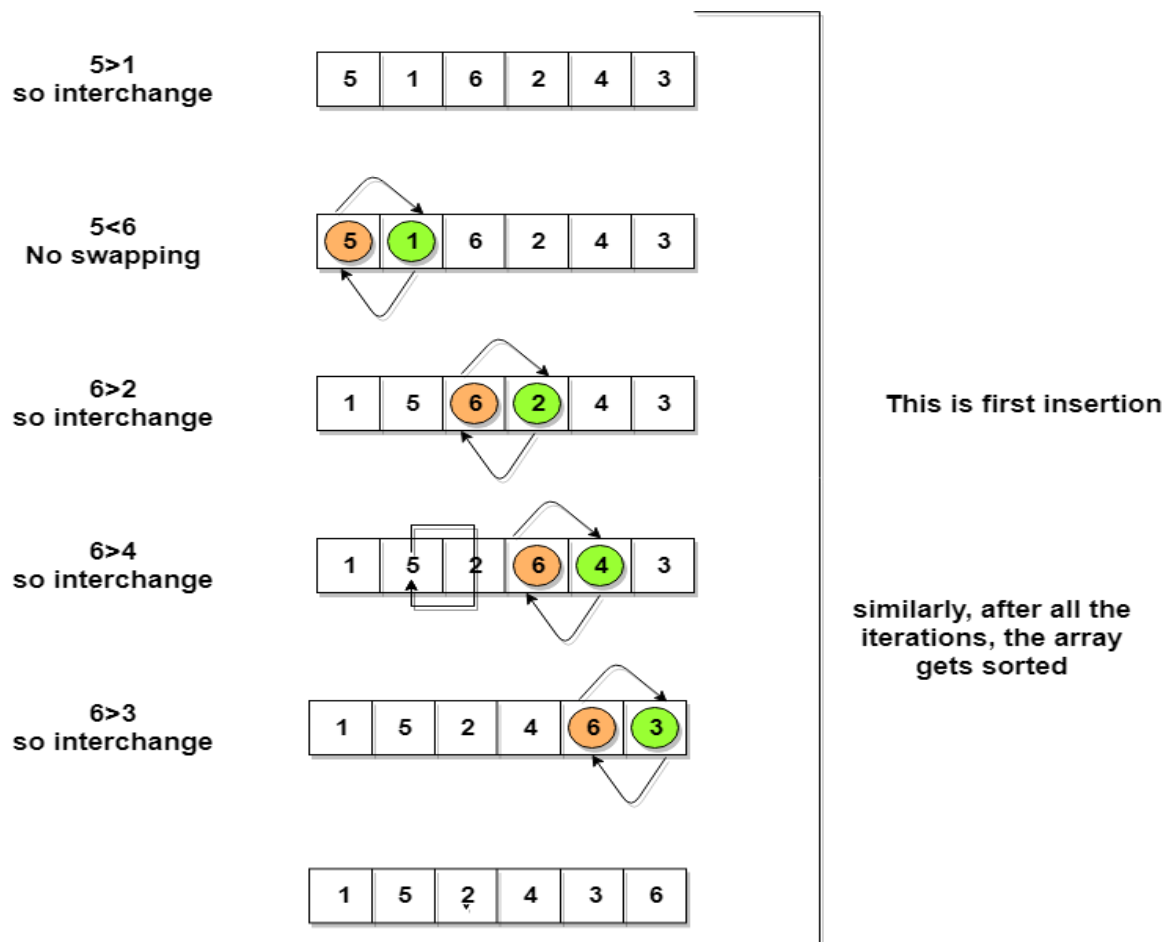
So as we can see in the representation above, after the first iteration, 6 is placed at the last index, which is the correct position for it.

Similarly after the second iteration, 5 will be at the second last index, and so on (Anon., 2020).

6. Reflection

The developed system is Student Information System. Student Information System is a comprehensive University Student Administration System, enabling any academic or educational organization to capture, maintain, update and provide accurate personal and academic information about all current and past students with any University. The system has modules that help accurately maintain Enrol, records and registration date.

The features of adding the new record through online and CSV file has the outstanding flexibility and performance to maintain the records. Accuracy of report showing with sorted listing the table inquiries and graphical chart and also the weekly report.

My experience is fairly exclusive with Visual Studio, in that it's the only IDE I've ever properly used. I came to know deeply and professionally with the language so far. Accuracy, less time consuming, flexibility etc. were found in this application. Features like creating chart, generating list in grid view. Additionally, sorting of data from the grid was a new achievement for me. Furthermore, import and exporting to CSV file was new and overall evaluation with the great support of Mr. Ishwor Sapkota. The coursework has come to an end.

7. Conclusion

For module CS6004NP Application Development, the initial coursework was to design and implement Student Information System in C# - desktop application (not a web-based or database application) for a company. It took a long time to build up the task in Visual Studio 2019 utilizing the required programming dialect. The framework has login screen to add security to the system. After login, the framework shows a primary screen where every functionalities is done. Aside from various shape components, class outline for every structures and classes were utilized.

And finally the coursework has been completed.

8. Bibliography

Alameri,   I.,   2017.   [Online]
Available   at:
https://www.researchgate.net/publication/319881143_Development_of_Stude
nt_Information_Management_System_based_on_Cloud_Computing_Platform
[Accessed 10 01 2020].

Anon.,   2020.   [Online]
Available   at: https://www.studytonight.com/data-structures/bubble-sort
[Accessed 10 01 2020].

Dipin   Budhrani,   V.   M.   G.,   2018.   [Online]
Available   at: https://www.ijedr.org/papers/IJEDR1801002.pdf
[Accessed 10 01 2020].

Pengtao   Yang,   G.   S.   J.   H.   P.   Z.   L.,   2017.   [Online]
Available   at: https://www.hindawi.com/journals/jece/2017/9598581/
[Accessed 10 01 2020].

Petkari,   E.,   2015.   [Online]
Available   at:
https://www.sciencedirect.com/science/article/pii/S1877042815041932
[Accessed 10 01 2020].

9. Appendix

Login.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 Login.xaml
    /// </summary>
    public partial class Login : Window
    {
        public Login()
        {
            InitializeComponent();
        }

        private        void        TextBox_TextChanged(object        sender,
TextChangedEventArgs e)
        {

        }
```

```csharp
        private void Button_Click(object sender, RoutedEventArgs e)
        {
            string user, pass;
            user = txt_username.Text;
            pass = txt_password.Text;

            if (user == "admin" & pass == "admin")
            {
                MessageBox.Show("Welcome to Student Management System");
                MainWindow mw = new MainWindow();
                mw.Show();
                this.Close();
            }
            else
            {
                MessageBox.Show("Username and Password did not match");
            }
        }

        private void btn_exit_Click(object sender, RoutedEventArgs e)
        {
            Application.Current.Shutdown();
        }
    }
}
```

MainWindow.cs

```csharp
using System;
using DataHandler;
using System.Collections.Generic;
using System.Data;
using System.Linq;
```

```csharp
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.IO;

namespace StudentInformationSystem
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            StudentInfo studentinfo = new StudentInfo();
            DataGrid.Items.Add(studentinfo);
        }


        public class StudentInfo
        {
            public string ID { get; set; }
            public string Name { get; set; }
            public string Address { get; set; }
            public string Phone { get; set; }
            public string CourseEnroll { get; set; }
```

```csharp
        public string RegistrationDate { get; set; }


    }


    private void btn_save_Click(object sender, RoutedEventArgs e)
    {
        var handler = new Handler();
        var dataSet = handler.CreateDataSet();
        AddSampleData(dataSet);
        MessageBox.Show("Data saved successfully !!");
        if (File.Exists(@"D:\studentData.xml"))
        {
            dataSet.ReadXml(@"D:\studentData.xml");
            dataSet.WriteXml(@"D:\studentData.xml");
        }
        else
        {
            dataSet.WriteXml(@"D:\studentData.xml");
        }
    }
    private void AddSampleData(DataSet dataSet)
    {
        var dr1 = dataSet.Tables["StudentInfo"].NewRow();
        dr1["ID"] = std_id.Text;
        dr1["Name"] = std_name.Text;
        dr1["Address"] = std_address.Text;
        dr1["Phone"] = std_phone.Text;
        dr1["CourseEnroll"] = course_enroll.Text;
        dr1["RegistrationDate"] = std_regdate.Text;
        dataSet.Tables["StudentInfo"].Rows.Add(dr1);
    }


    private void import_report_Click(object sender, RoutedEventArgs e)
```

```
    {
        StudentInfo studentInfo = new StudentInfo();
        studentInfo.ID = std_id.Text;
        studentInfo.Name = std_name.Text;
        studentInfo.Address = std_address.Text;
        studentInfo.Phone = std_phone.Text;
        studentInfo.CourseEnroll = course_enroll.Text;
        studentInfo.RegistrationDate = std_regdate.Text;


        DataGrid.Items.Add(studentInfo);
    }


    private void display_report_Click(object sender, RoutedEventArgs e)
    {
        StudentReport studentReport = new StudentReport();
        studentReport.Show();
    }
  }
}
```

StudentReport.cs

```
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 StudentReport.xaml
    /// </summary>
    public partial class StudentReport
    {
        DataTable buffer;
        public StudentReport()
        {
            InitializeComponent();
        }
        private void display_info()
        {
            string displayreport = @"D:\studentData.xml";
            DataSet dataset = new DataSet();
            dataset.ReadXml(displayreport);

            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("Phone", 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 datainfo = new DataView(buffer);
    DataGridReport.ItemsSource = datainfo;
}
private void btn_retrive_Click(object sender, RoutedEventArgs e)
{
    display_info();
}


private void btn_sortbydate_Click(object sender, RoutedEventArgs e)
{
    DataView view = new DataView(buffer);
    view.Sort = "RegistrationDate ASC";
    DataGridReport.ItemsSource = view;
}


private void btn_sortbyname_Click(object sender, RoutedEventArgs e)
{
    DataView view = new DataView(buffer);
    view.Sort = "Name ASC";
    DataGridReport.ItemsSource = view;
}


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

```csharp
dataSet.ReadXml(@"D:\studentData.xml");

DataTable dtStdReport = dataSet.Tables[0];

int total_Networking = 0;
int total_Multimedia = 0;
int total_Computing = 0;

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

for (int i = 0; i < dtStdReport.Rows.Count; i++)
{
    String col = dtStdReport.Rows[i]["CourseEnroll"].ToString();
    if (col == "Networking")
    {
        total_Networking++;
    }
    else if (col == "Multimedia")
    {
        total_Multimedia++;
    }
    else if (col == "Computing")
    {
        total_Computing++;
    }

}

dt.Rows.Add("Networking", total_Networking);
dt.Rows.Add("Multimedia", total_Multimedia);
dt.Rows.Add("Computing", total_Computing);
```

```
          DataGridReport.DataContext = dt.DefaultView;
      }


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


      private void btn_Csv_Click(object sender, RoutedEventArgs e)
      {
          try
          {
              var dataSet = new DataSet();
              dataSet.ReadXml(@"D:\studentData.xml");
              Microsoft.Win32.OpenFileDialog        dialog        =        new
Microsoft.Win32.OpenFileDialog();
              if (dialog.ShowDialog() == true)
              {
                  string filename = dialog.FileName;
                  using (var read = new StreamReader(filename))
                  {
                      read.ReadLine();
                      while (!read.EndOfStream)
                      {
                          var line = read.ReadLine();
                          var values = line.Split(',');
                          var newRow = dataSet.Tables["StudentInfo"].NewRow();

                          newRow["ID"] = values[0];
                          newRow["Name"] = values[1];
                          newRow["Address"] = values[2];
                          newRow["Phone"] = values[3];
                          newRow["CourseEnroll"] = values[4];
```

```
                        newRow["RegistrationDate"] = values[5];
                        dataSet.Tables["StudentInfo"].Rows.Add(newRow);


                        dataSet.WriteXml(@"D:\studentData.xml");
                    }
                }
                MessageBox.Show("Student record sucessfully imported");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
  }
}
```

Handler.cs

```
using System;
using System.Data;

namespace DataHandler
{
  public class Handler
  {
    public DataSet CreateDataSet()
    {
      var ds = new DataSet();

      ds.Tables.Add(CreateStudentInfoTable());
      ds.Tables.Add(CreateStudentReport());
      return ds;
    }
```

```
    private DataTable CreateStudentInfoTable()
    {
        var dt = new DataTable("StudentInfo");
        dt.Columns.Add("ID", typeof(string));
        dt.Columns.Add("Name", typeof(string));
        dt.Columns.Add("Address", typeof(string));
        dt.Columns.Add("Phone", typeof(string));
        dt.Columns.Add("CourseEnroll", typeof(string));
        dt.Columns.Add("RegistrationDate",typeof(DateTime));
        return dt;
    }


    private DataTable CreateStudentReport()
    {
        var dt = new DataTable("StudentReport");
        dt.Columns.Add("ID", typeof(string));
        dt.Columns.Add("Name", typeof(string));
        dt.Columns.Add("Address", typeof(string));
        dt.Columns.Add("Phone", typeof(string));
        dt.Columns.Add("CourseEnroll", typeof(string));
        dt.Columns.Add("RegistrationDate", typeof(DateTime));
        return dt;
    }
  }
}
```