APPLICATION DEVELOPMENT

CU6051NA

Coursework 1

Student Name: Ankit Gurung (Ankit.gurung.4s18@icp.edu.np)

Student ID: 17031937

Course: BSc (Hons) Computing

Submitted To: Mr. Ishwor Sapkota

Subject: Application Development

Group: L3C2

Date: 10- jan-2020

Table of Contents

List of Figures

Table of tables

1 Introduction

The Main target of this project is to create and implement a “Student Information System”. The most important features (i.e. Student login from, Student details saving, Retrieving student details, Data registration sorting by date, Data registration sorting by name and data display chart showing total number of student enrolled so far through the CSV file) in an effective and flexible user interface manual. All the typed information holds details like name, address, contact no. , registration date, registration no. and course enrol. Moreover, the other feature is to demonstrate no. of student enrolled on particular courses weekly followed by the chart.

1.1 Current Scenario:

There are countless no of student management system on each and every teaching institution, which is very raw and unmanaged. However, if they have a well-managed digital record system. They lack the vital features to elevate the record management system to a very convenient, feasible and user friendly manual.

1.2 Proposed System:

The developed system is for overlapping all the backlashes of the previous student record management system through security login section with proper features display with user friendly interface.

2 User Manual

2.1 Login Form:

When the program code runs, it shows the login window with username and password for security purpose. Both should, match the system username and password, otherwise it depicts “Login failed!!” message.

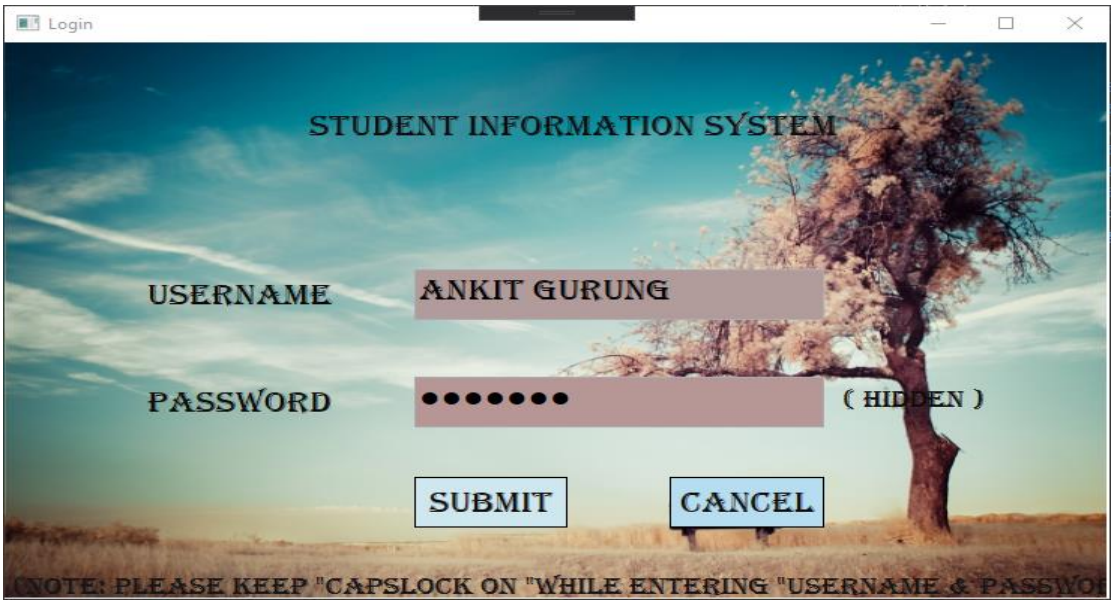


Figure 1: Login Form

2.2 Main Window:

When the Login id and password matches the system, man window opens with the mandatory to be filled text box meeting the requirement and press ‘save’ button.

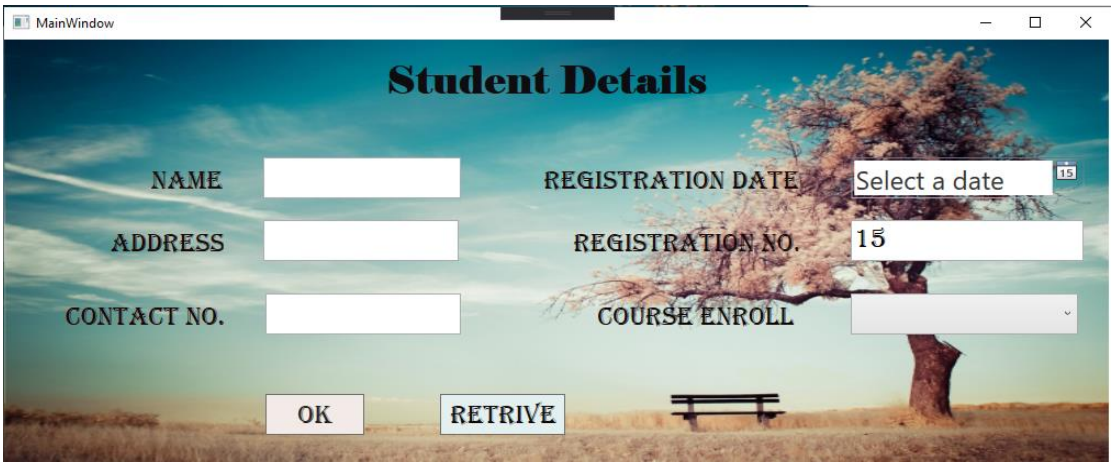


Figure 2: Main window.

2.3 Empty Data Error Validation:

If the any text box of the particular field is un-filled or kept empty, then it throws an exception with the message dialog box showing “ERROR!! PLEASE, FILL UP REQUIRED DETAILS!!”

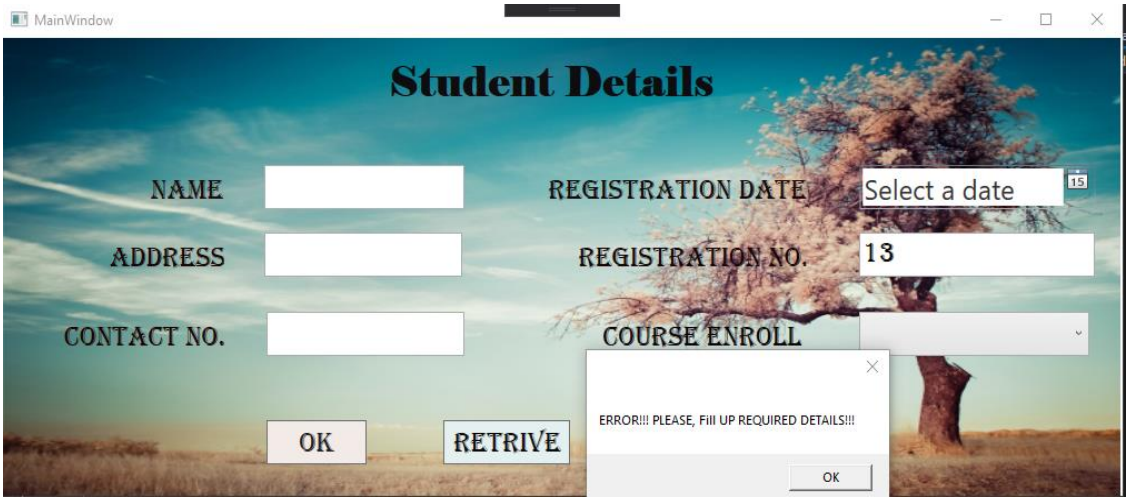


Figure 3: Empty Data Error validation.

2.4 Student Details fill up & saved message:

When the text box of the particular field is filled and ok button is pressed, then it save the typed data to the Student Report table showing “Student details is stored” message dialog box.

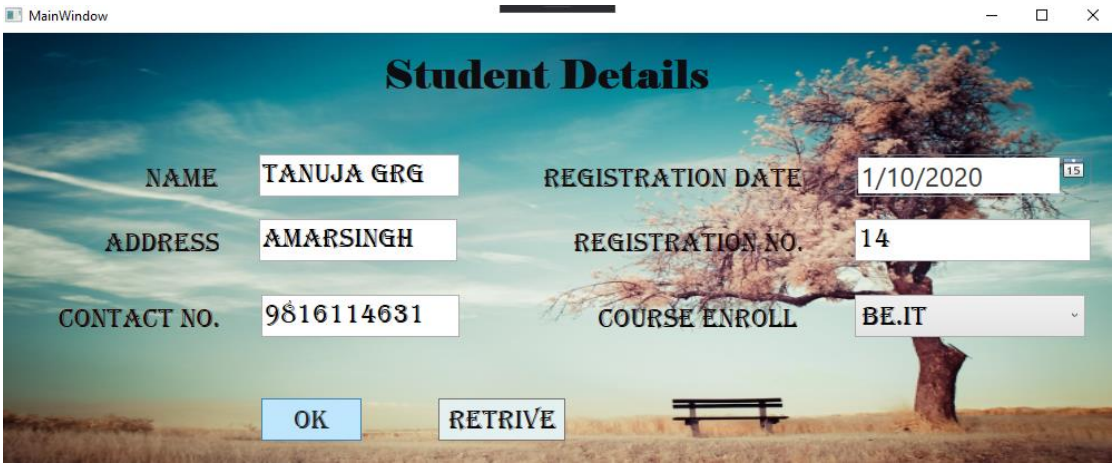


Figure 4: Students details filled up.

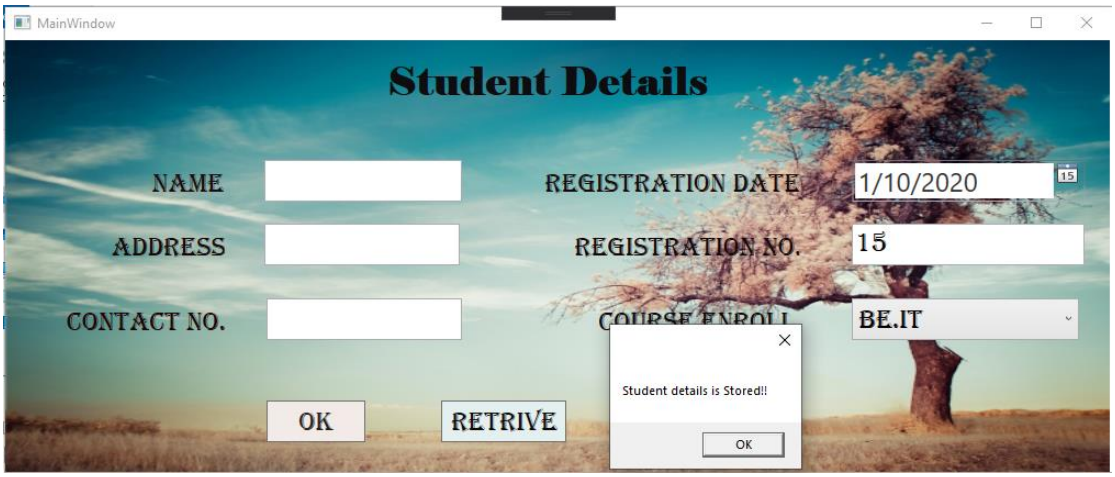


Figure 5: Student details saved message.

2.5 Retrieve button pressed and data retrieved:

When the retrieve button is pressed, it shows all the student data fill on 'Student Report' in the data grid with the message dialog box "Student Data is retrieved."

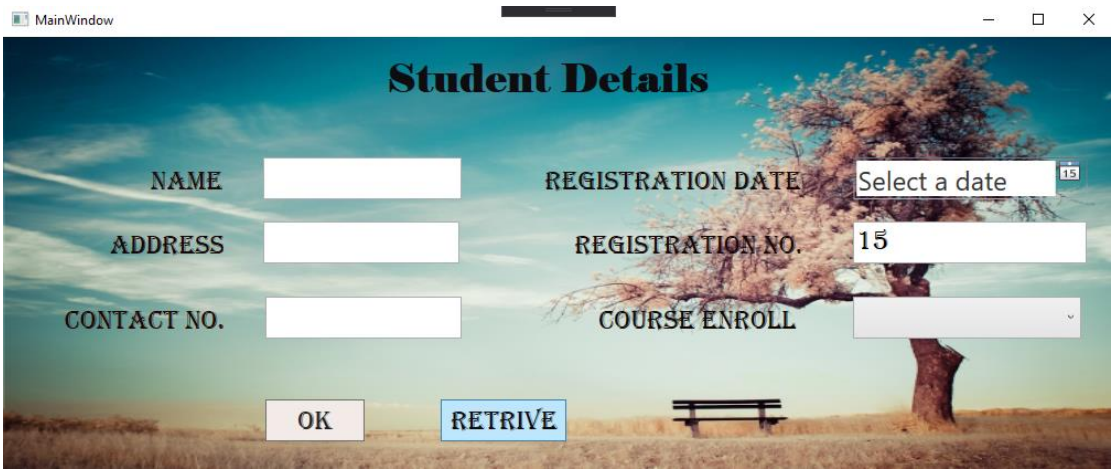


Figure 6: Retrieve button pressed.

Figure 7: Data Retrieved

2.6 Sorting By Date:

When the “Student Registration By Data” button is pressed. It opens a new window data grid depicting all the data entered, in date ascending order showing the particular enrolled date.

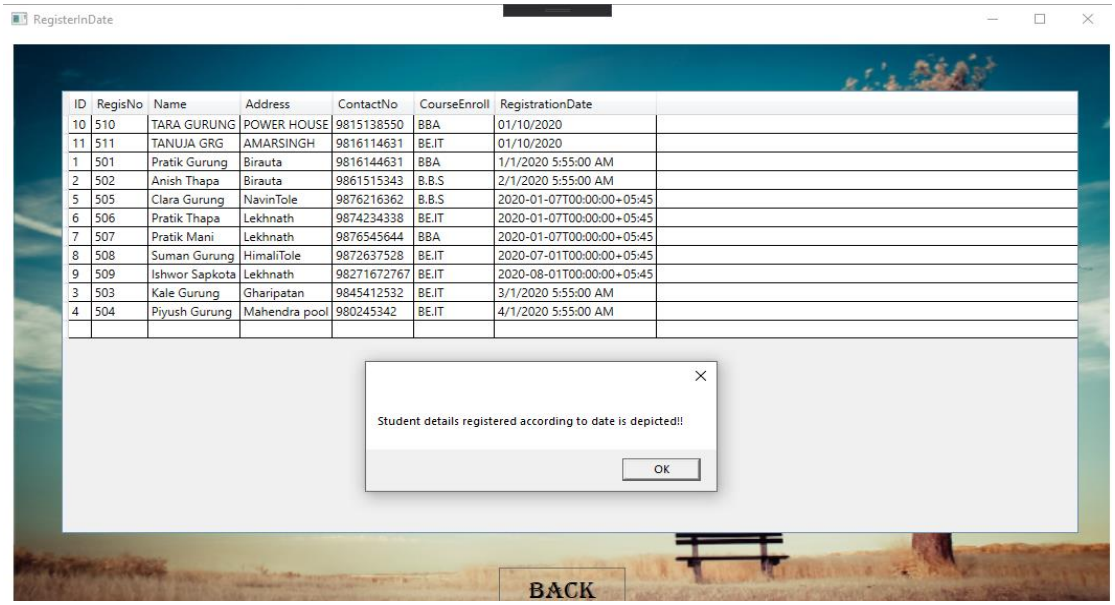


Figure 8: Sorting by date.

2.7 Sorting By Name:

When the “Student Registration By Name” button is pressed. It opens a new window data grid depicting all the data entered, in name ascending order showing the particular enrolled date.

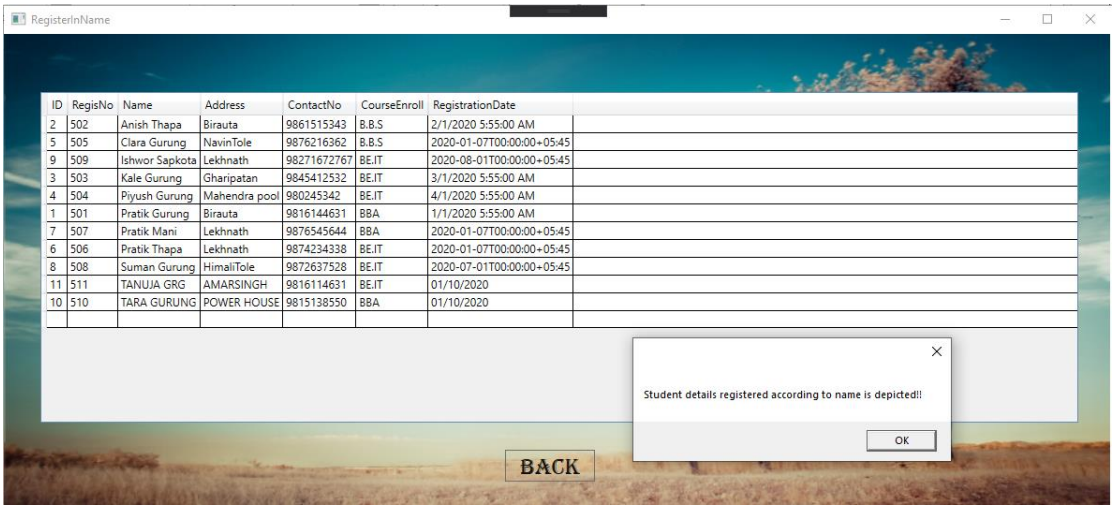


Figure 9: Sorting By Name.

2.8 Export Student Report On Weekly Basis:

When the “Report On Weekly basis” button is pressed. It opens a new window data grid depicting all the data entered, with the course enrolled and overall student column name holding data on particular enrolled courses.

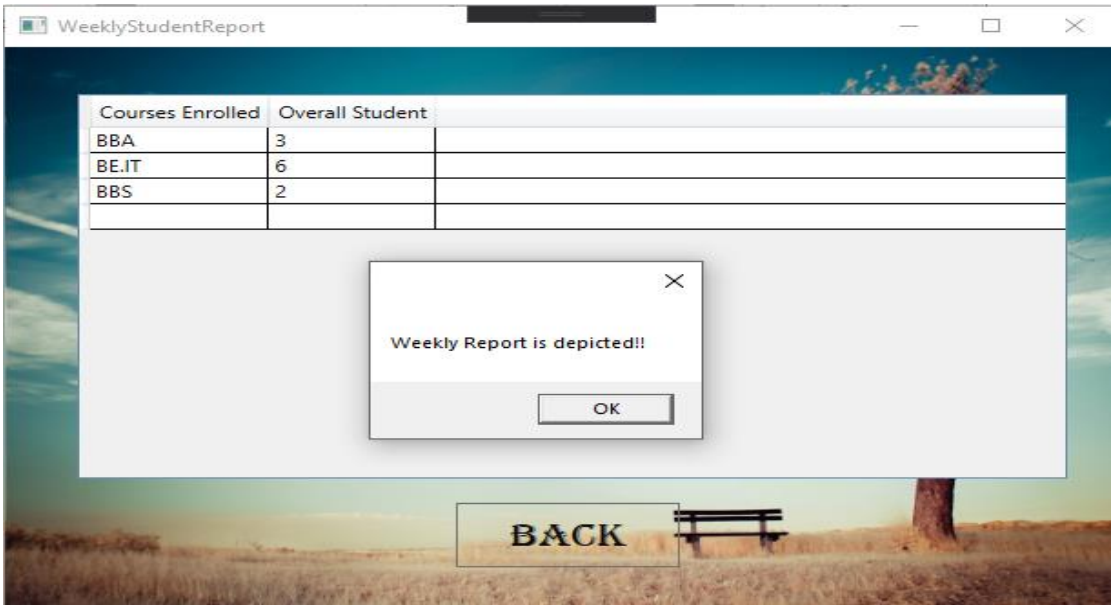


Figure 10: Weekly Report.

2.9 Extract Chart Diagram:

When the “Extract Chart Diagram” button is pressed. It opens a new window depicting all the data entered, in pie chart form of particular courses.

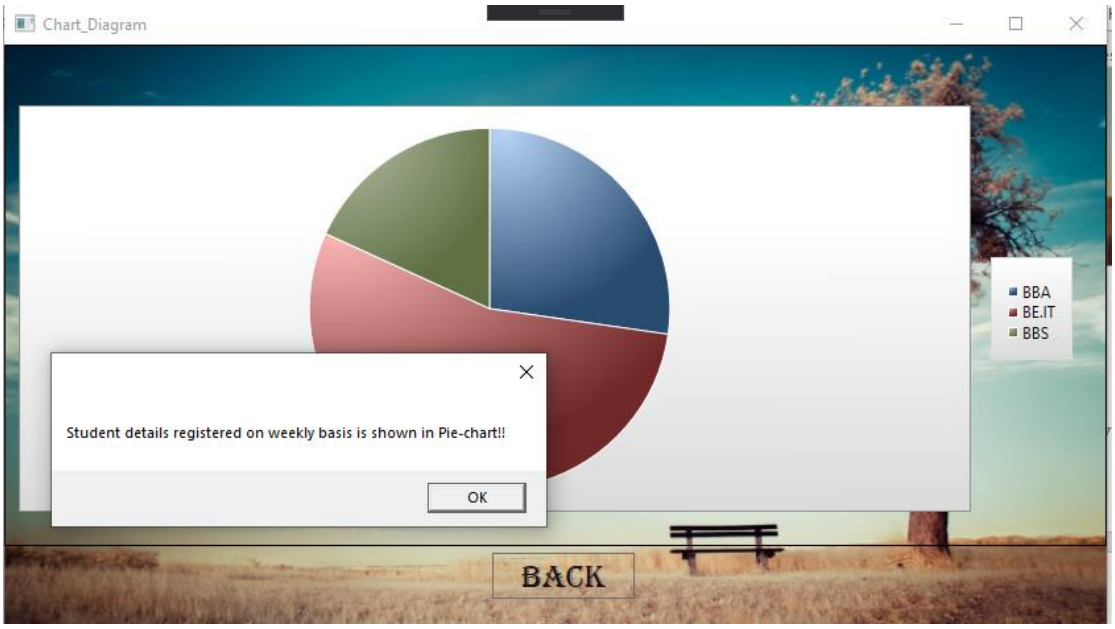


Figure 11: Graphical Representation of Student Report

2.10 Back to the respective connected window:

When the “Back” button is pressed. It returns the project to the previous window closing the respective window and showing the connected windowdata.

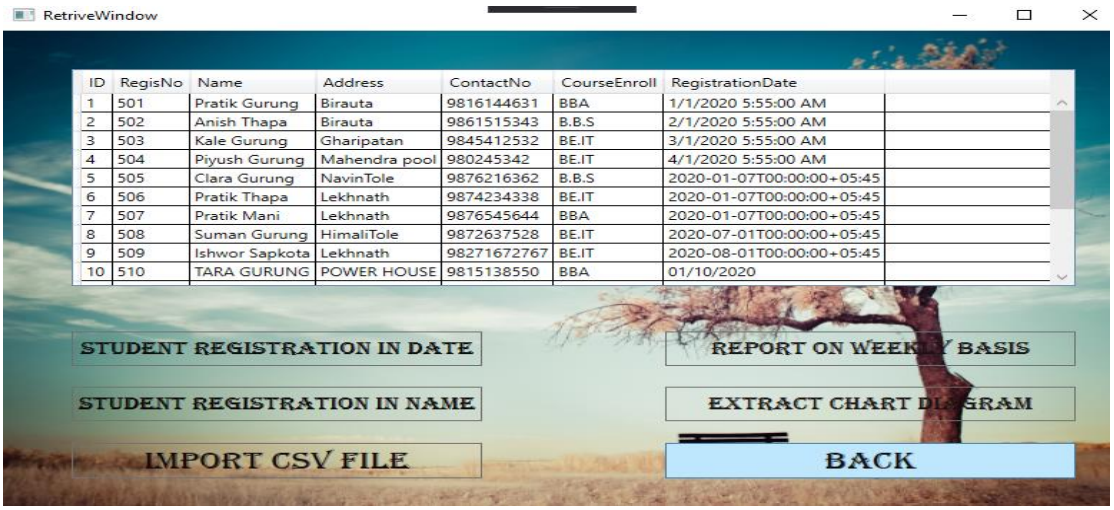


Figure 12: Back button pressed.

3 Journals:

These are the following journal from which I took some references:

1. At the moment, increasingly ICT-related investments in an overgrowing competitive school setting, the deployment, maintenance, and predominantly the active use of ICT is in many ways a complex complicated managerial task involving several stakeholder groups. In a pilot study of a single school district in a Swedish municipality, we have interrogated governments from the municipal board of education, the municipal IT-support for schools, and two principals. We have used Technology Acceptance Model (TAM3) as our analytical lens to explore how school principals' and municipal IT-managers perceive ICT adoption, usefulness, and the potential role of ICT. We conclude that the barriers for a successful integration of ICT into school-related activities requires a holistic managerial thinking in order to overcome the lack of coordinated ICT investment strategy and tracking of ICT **(ResearchGate, 2018).**

2. The subdivision concentrates on a cross-national comparison of mediatized schools in Germany and England. Based on the postulation that both school systems follow the same goal of providing good school education, the question arises as to why the mediatized equipment is so different. Our practical results show that English schools are far more mediatized, exhibiting a higher number of computers, notebooks and tablets in schools as well as digital systems and services. Non-mediatized communication forms dominate in German schools with a high usage of pen and paper or pigeon holes. The different mediatized practices also affect communication with pupils and parents, following the same characteristics as inter-teacher communication. On the other hand, teachers in both countries emphasize the importance of face-to-face contact and direct personal communication. One reason for the differences may be founded in the different educational governance of both countries **(ResearchGate, 2018)**

3. ACM Transactions on Management Information Systems (TMIS) is a scholarly quarterly journal that focuses on publishing high quality information systems research. TMIS welcomes innovative work on the design, development, assessment, and management of information technology and systems within organizations, businesses, and societies. TMIS welcomes submissions on a full range of MIS and information technology related areas and strongly encourages submissions with technical and technological ingredients, such as algorithmic, analytical modelling, design science, and system-oriented research, as well as submissions in emerging multidisciplinary MIS research topics that may span several traditional academic disciplines **(Digital, 2010)**

4. Developments in information technologies have been impacting upon educational organizations. Principals have been using management information systems to improve the efficiency of administrative services. The aim of this research is to explore principals' perceptions about management information systems and how school management information systems are used in primary schools. The respondents of this study were 98 elementary school principals in Edirne. Data were gathered using a five-part questionnaire. The first part collected demographic information about respondents. The others had statements about school management information systems. The data were analysed using frequency, percentage, mean and standard deviation. Results indicated that although technologic infrastructures of elementary schools are insufficient, school management information systems have an important contribution to school management **(ResearchGate, 2006)**

4 System Architecture

4.1 Architecture Diagram

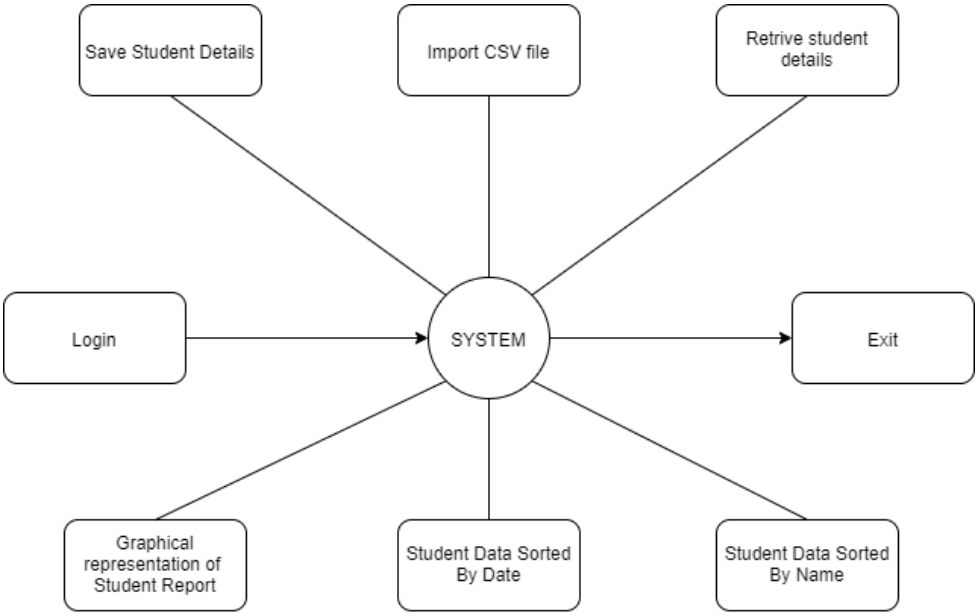


Figure 13: Architecture Diagram.

5 Class Diagram

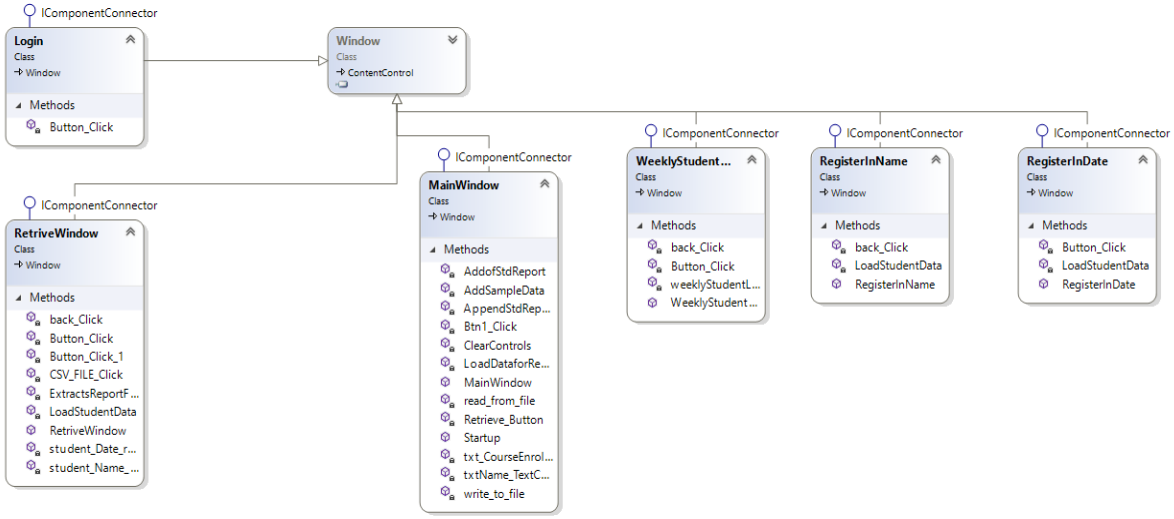
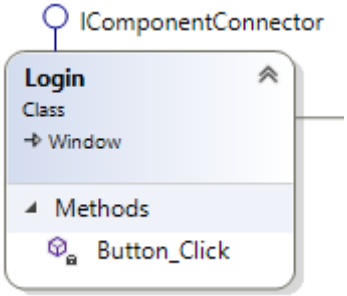


Figure 14: Class Diagram

5.1 Login

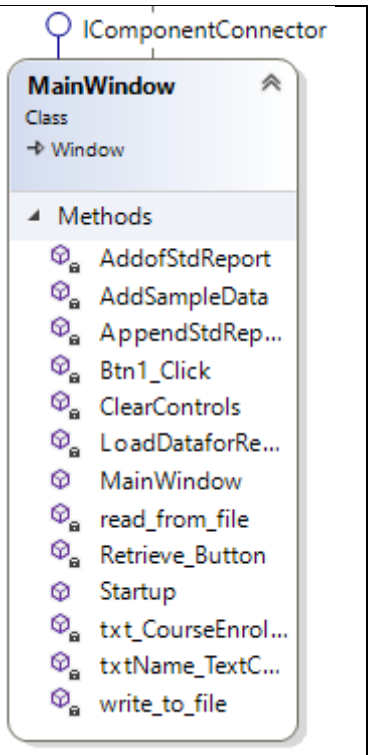
Table 1: Login Form

Methods	Description	
Button_Click	If the username and password matches to the system then it allows login with message dialog box otherwise it throws exception showing message dialog box.	

5.2 MainWindow

Table 2: MainWindow.

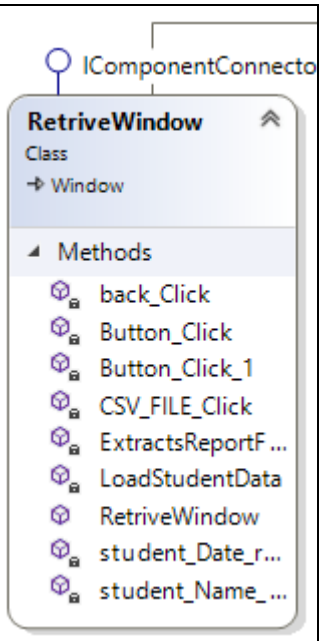
Methods	Description
AddofStdReport	Reads previous data from the xml file and adds the new added data taken from the text box and stores in xml file.
Btn1_click	Contains validation of the text field and if all text field are filled then the details are stored in xml file.
ClearControls	Clears the text field after inserting the value in xml.
LoadDataforReport	This method creates the table for weekly enrolment of the students and adds the student's details on it.
write_to_file	This method is for writing content in files.
read_from_file	This method increments the registration number so that no students get same registration number.



Retrieve_Button	This buttons creates the object of the RetrieveWindow.Xaml and shows the student details by closing the main window showing the dialog box.	

5.3 RetrieveWindow

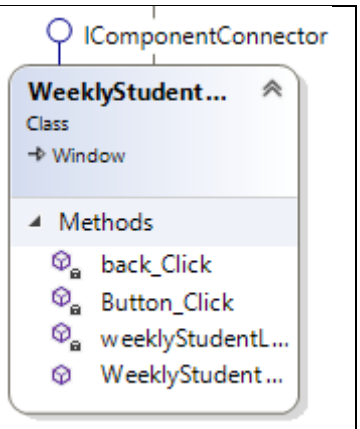
Table 3: RetrieveWindow.

Methods	Description	
back_Click	This buttons creates the object of the MainWindow.Xaml and open the MainWindow by closing the RetrieveWindow.Xaml.	
Button_Click	This buttons creates the object of the Chart_Diagram.Xaml and open the Chart_Diagram by closing the RetrieveWindow.Xaml.	
Button_Click1	This buttons creates the object of the WeeklyStudentReport.Xaml and open the WeeklyStudentReport.Xaml by closing the RetrieveWindow.Xaml.	
CSV_FILE_Click	Open File Dialog box and calls getCsv method to append the data in data grid.	

ExtractReportFromCSV	Select data from the csv file and append it in data table.	
LoadStudentData	If the StudentReport file exists then it new dataTable dtstdReport and adds it to the Data grid.	
RetrieveWindow	This method calls the LoadStudentData & intializeComponent.	
Student_Date_regis_Click	This buttons creates the object of RegisterInDate.Xaml and open the RegisterInDate.Xaml by closing the RetrieveWindow.Xaml	
Student_Name_regis_Click	This buttons creates the object of RegisterInName.Xaml and open the RegisterInName.Xaml by closing the RetrieveWindow.Xaml	

5.4 WeeklyStudentReport

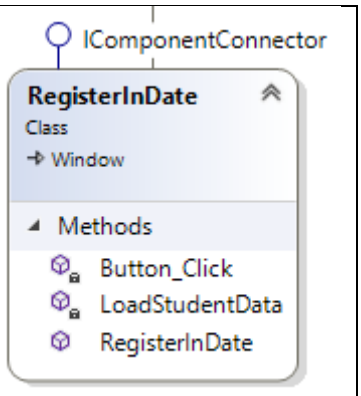
Table 4: WeeklyStudentReport

Methods	Description	
back_click	This buttons creates the object of RetrieveWindow.Xaml and open the RetrieveWindow.Xaml by closing the WeeklyStudentReport .Xaml	
Button_Click	This buttons creates the object of RetrieveWindow.Xaml and open the RetrieveWindow.Xaml by closing the WeeklyStudentReport .Xaml	
WeeklyStudentReport	This method calls IntializeComponent and weeklyStudentList function.	
WeeklyStudentList	Reads the data in the table and gives the number of student enrolled in specific course.	

5.5 RegisterInDate

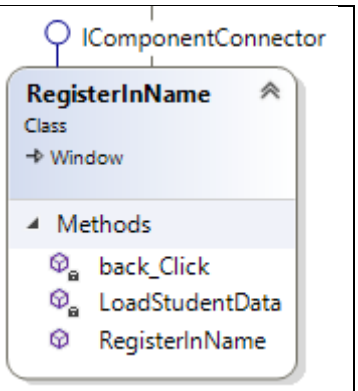
Table 5: RegisterInDate.

Methods	Description
back_click	This buttons creates the object of RetrieveWindow.Xaml and open the RetrieveWindow.Xaml by closing the RegisterInDate.Xaml
LoadStudentData	If the StudentReport file exists then it new dataTable dtstdReport with sorting of student list through date and added to the Data grid.
RegisterInName	This method calls IntializeComponent and LoadStudentData function.



5.6 RegisterInName

Table 6: RegisterInName

Methods	Description	
back_click	This buttons creates the object of RetrieveWindow.Xaml and open the RetrieveWindow.Xaml by closing the RegisterInDate.Xaml	
LoadStudentData	If the StudentReport file exists then it new dataTable dtstdReport with sorting of student list through name and added to the Data grid.	
RegisterInName	This method calls IntializeComponent and LoadStudentData function.	

6 Sorting Algorithm

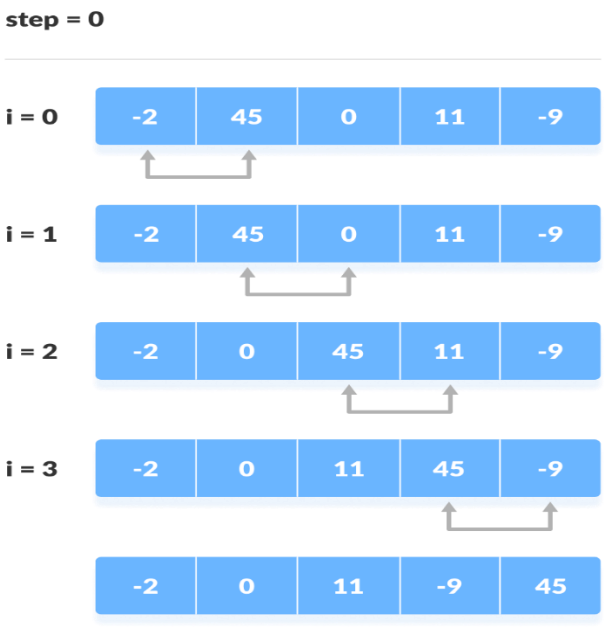
The sorting Algorithm used in this project is Bubble Sort Algorithm. It sorts by comparing the array one by one on the basis of order requirement of the project.

Bubble Sort Algorithm is used to arrange N elements in ascending order, and for that, you have to begin with 0th element and compare it with the first element. If the 0th element is found greater than the 1st element, then the swapping operation will be performed, i.e., the two values will get interchanged. In this way, all the elements of the array get compared. **(programiz, 2018)**

1. Starting from the first index, compare the first and the second elements. If the first element is greater than the second element, they are swapped.

   Now, compare the second and the third elements. Swap them if they are not in order.

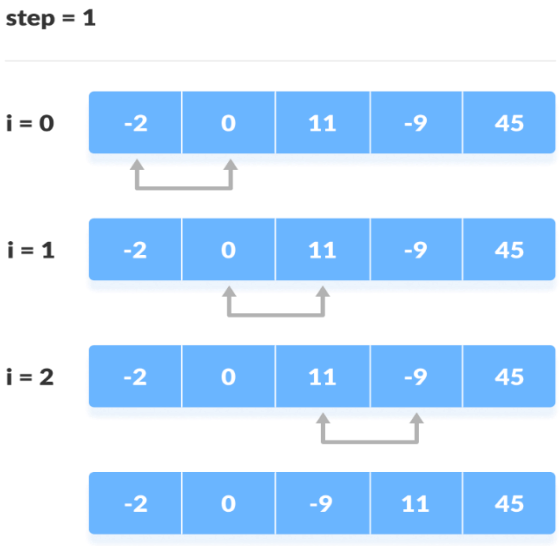
   The above process goes on until the last element.



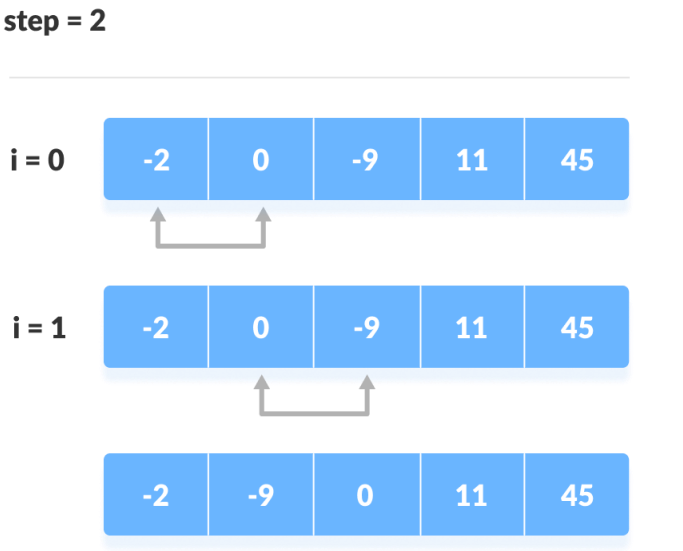
2. The same process goes on for the remaining iterations. After each iteration, the largest element among the unsorted elements is placed at the end.

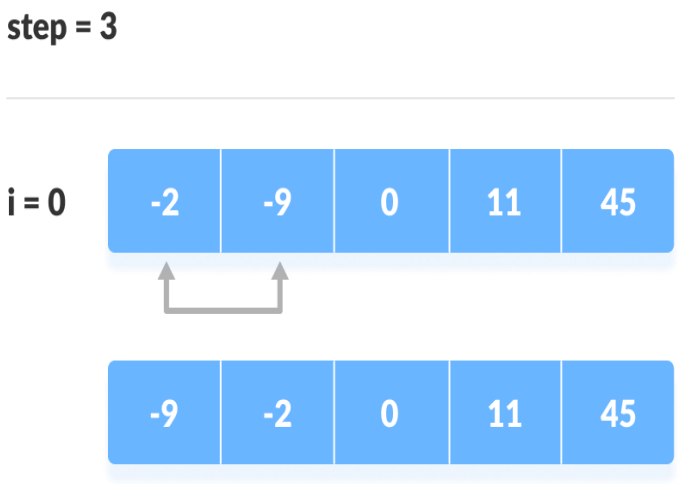
In each iteration, the comparison takes place up to the last unsorted element.

The array is sorted when all the unsorted elements are placed at their correct positions.



3.


step = 2
i = 0
-2
0
-9
11
45
i = 1
-2
0
-9
11
45
-2
-9
0
11
45


step = 3
i = 0
-2
-9
0
11
45
-9
-2
0
11
45

7 Flowchart

7.1 Student Enrol:

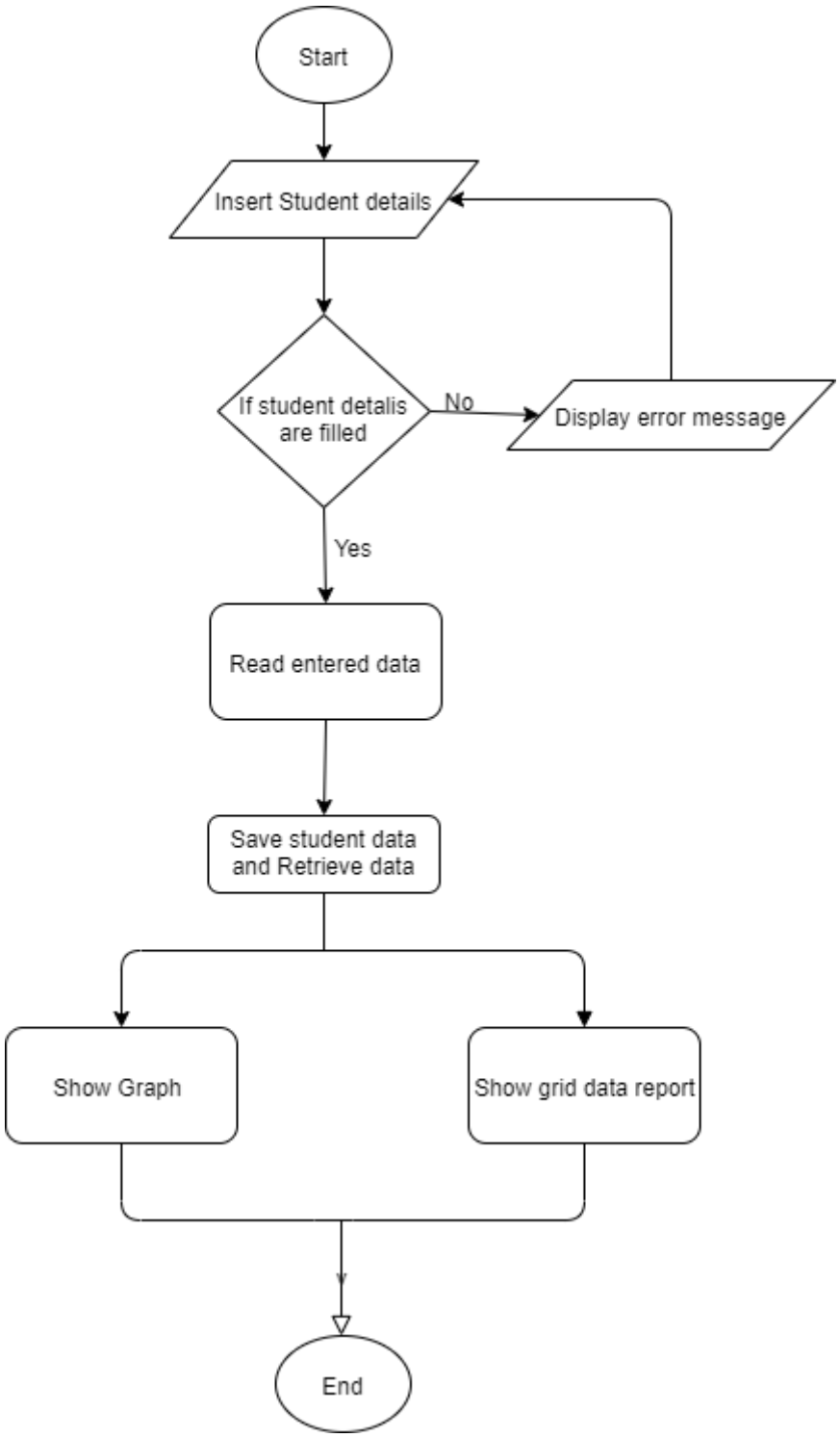


Figure 15: Student Enrol.

7.2 Importing CSV File

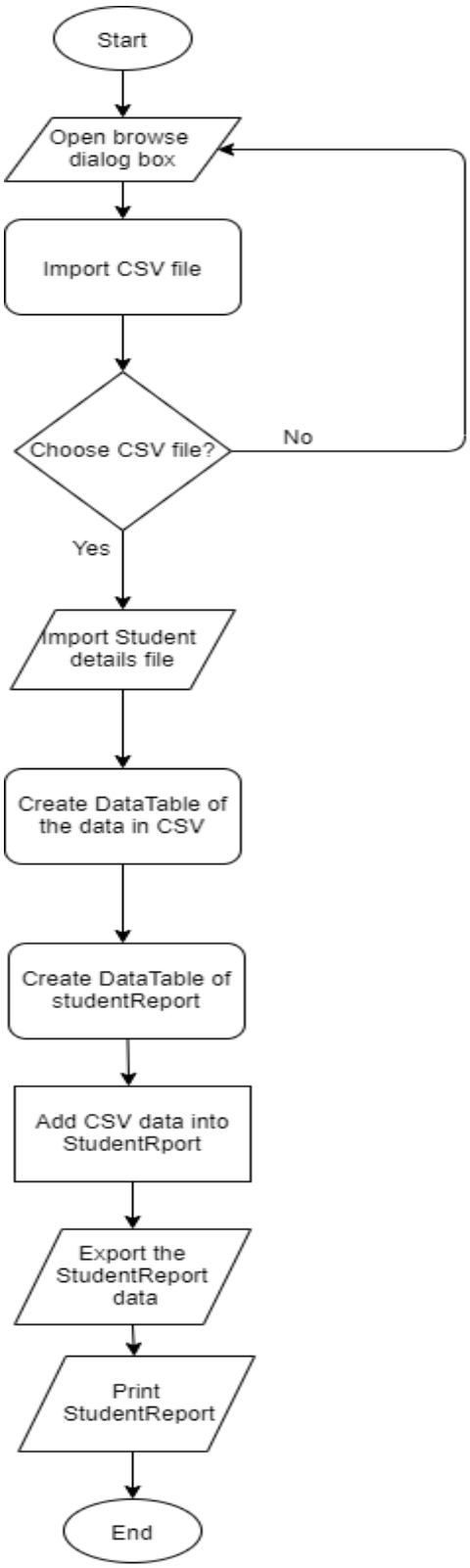


Figure 16: Importing CSV file.

8 Reflection

This was a fully functionalized Student Information System with proper management of the details. It demonstrates the main rationality behind the appropriate environs on management system. Moreover, it has an attractive & user-friendly user interface functioning through the system.

The system holds several functionalities for the end-users to operate like inserting the student details and extracting the required results through the respective button click. In addition, the project itself gathers other function holding the prime tasks (i.e. preparing the student report, sorting by date, sorting by name and weekly enrolled student details) on respective buttons with attractive GUI representation to work on project.

At first, it was tough call to work on. It was completely new environment of creating management system in Visual Studio. But after, numerous researches and help from our module tutor and colleagues, project was finally completed. Various functioning features on the project like data serialization (Save), data de-serialization (Retrieve), data sorting through name), data sorting by name, data sorting by data and importing Csv file was extra learning point on the coursework. Furthermore, it was a great experience to functionalize the method on the project.

9 Conclusion

The is the first coursework of Application Development to develop Student Information System. This project helped to build up my confidence that I can build another identical type of information system on visual studio using WPF form.

However, it was not an easy task to overcome but with the helping of our module tutor Mr. Ishwor Sapkota and colleagues. It came to an end with proper functionality.

Bibliography

Digital, A., 2010. *journal.* . ACM Digital liabrary. ed. s.l.:Zeng.

programiz, 2018. *Bubble Sort Algorithm.* https://www.programiz.com/dsa/bubble-sort ed. s.l.:www.programiz.com.

ResearchGate, 2006. *journal.* ResearchGate. School management information systems. ed. s.l.:Demir.

ResearchGate, 2018. *journal.* ResearchGate,. Managing the Digitalization of Schools. ed. s.l.:babaheidari.

ResearchGate, 2018. *journal.* . ResearchGate. Governing the Figurations of Mediatized Schools. ed. s.l.:Breiter.

Appendix

1 LOGIN FORM

```
using CourseWorkSample;
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 APPLICATION_DEVELOPMENT
{
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Window
    {
        private void Button_Click(object sender, RoutedEventArgs e)
        {
                if (Txt_username.Text == "ANKIT GURUNG" &&
Txt_Password.Password == "ANKIT22")
                {
                    MessageBox.Show("Logged in Successfully", "Alert");
                    MainWindow Log = new MainWindow();
                     Log.Show();
                     Close();

                }
                else
                {
                    MessageBox.Show("Login Fail!", "Alert");
                    Txt_username.Clear();
                    Txt_Password.Clear();
                }

        }
    }
}
```

2 Main Window

```
using DataHandler;
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;
using System.Data;
using APPLICATION_DEVELOPMENT;

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

            txtRegisNo.Text = read_from_file();//Registered regisno. readedfrom
the saved file.
           // txt_Date.Text = DateTime.Now.ToString("dd/MM/yyyy");//Extractes
the actual date.

        }

       public void Startup()
        {
            var i = 0;//Variables initialized to 0.
        }


        private void AddSampleData(DataSet dataSet)
        {
            var dr = dataSet.Tables["Course"].NewRow();//Course table created.

            // Data extrated from the text Field//
            dr["Name"] = " txtName.Text";
            dr["Address"] = "txtAddress.Text";
            dr["ContactNo"] = " txtContact.Text ";
            dr["CourseEnroll"] = " txtCourseEnroll.Text ";
            dr["RegsitrationDate"] =
txt_date.SelectedDate.Value.ToString("MM/dd/yyyy");
        }

        private void AddofStdReport(DataSet dataSet)
        {
```

```
            var handler = new Handler();//New handler created for student
report.

dataSet.Tables["StudentReport"].ReadXml(@"F:\StudentReport.xml");//Data stored
in student report is readed.

            var dr1 = dataSet.Tables["StudentReport"].NewRow();//New data row
on the smae table is created.
            dr1["RegisNo"] = txtRegisNo.Text;
            dr1["Name"] = txtName.Text;
            dr1["Address"] = txtAddress.Text;
            dr1["ContactNo"] = txtContact.Text;
            dr1["CourseEnroll"] = txt_CourseEnroll.Text;
            dr1["RegistrationDate"] =
txt_date.SelectedDate.Value.ToString("MM/dd/yyyy");
            dataSet.Tables["StudentReport"].Rows.Add(dr1);


dataSet.Tables["StudentReport"].WriteXml(@"F:\StudentReport.xml");//All data
added to the StudentReport through variable dr1.


        }
        private void AppendStdReport(DataSet dataSet)
        {
                var handler = new Handler();//New handler created for data to
be appended on student report.


dataSet.Tables["StudentReport"].ReadXml(@"D:\StudentReport.xml");

                var dr2 = dataSet.Tables["StudentReport"].NewRow();
                dr2["RegisNo"] = txtRegisNo.Text;
                dr2["Name"] = txtName.Text;
                dr2["Address"] = txtAddress.Text;
                dr2["ContactNo"] = txtContact.Text;
                dr2["CourseEnroll"] = txt_CourseEnroll.Text;
                dr2["RegsitrationDate"] =
txt_date.SelectedDate.Value.ToString("MM/dd/yyyy");

            dataSet.Tables["StudentReport"].Rows.Add(dr2);

dataSet.Tables["StudentReport"].WriteXml(@"F:\StudentReport.xml");//All data
added to the StudentReport through variable dr2.

        }

        private void Btn1_Click(object sender, RoutedEventArgs e)
        {
            if (txtName.Text == "")
            {
                String check1 = string.Empty;
                MessageBox.Show("ERROR!!! PLEASE, Fill UP REQUIRED
DETAILS!!!");
                return;
            }
            else if (txtAddress.Text == "")
            {
                String check2 = string.Empty;
                MessageBox.Show("ERROR!!! PLEASE, Fill UP REQUIRED
DETAILS!!!");
```

```
                return;
            }
            else if (txtContact.Text == "")
            {
                String check2 = string.Empty;
                MessageBox.Show("ERROR!!!PLEASE, Fill UP REQUIRED DETAILS!!!");
                return;
            }
            else if (txt_CourseEnroll.Text == "")
            {
                String check2 = string.Empty;
                MessageBox.Show("ERROR!!!PLEASE, Fill UP REQUIRED DETAILS!!!");
                return;
            }



            var handler = new Handler();
            var dataSet = handler.CreateDataSet();

            AddofStdReport(dataSet);


            var regno = txtRegisNo.Text;
            var name = txtName.Text;


dataSet.Tables["Student"].WriteXml(@"F:\files\"+name+""+regno+".xml");


            write_to_file(txtRegisNo.Text);

            txtRegisNo.Text = read_from_file();


            ClearControls();
            MessageBox.Show("Student details is Stored!!");

        }

        private void txtName_TextChanged(object sender, TextChangedEventArgs e)
        {

        }


        private void write_to_file(string text)
        {
            System.IO.File.WriteAllText(@"F:\count.txt", text);
        }
        private string read_from_file()
        {

            string text = System.IO.File.ReadAllText(@"F:\count.txt");

            int i;

            i = int.Parse(text.ToString());
            i = i + 1;

            return i.ToString();
```

```
        }

        private void ClearControls()
        {
            txtName.Text = "";
            txtAddress.Text = "";
            txtContact.Text = "";
            //txt_Date.Text = DateTime.Now.ToString("dd/MM/yyyy");
        }


        private void LoadDataforReport1()
        {
            var handler = new Handler();
            var dataSet = new DataSet();

            var dr2 = dataSet.Tables["WeeklyReport"].NewRow();
            dr2["RegisNo"] = txtRegisNo.Text;
            dr2["Name"] = txtName.Text;
            dr2["Address"] = txtAddress.Text;
            dr2["ContactNo"] = txtContact.Text;
            dr2["CourseEnroll"] = txt_CourseEnroll.Text;
            dr2["RegsitrationDate"] =
txt_date.SelectedDate.Value.ToString("MM/dd/yyyy");
            dataSet.Tables["StudentReport"].Rows.Add(dr2);//All data added to
the student through variable dr2.
        }

        private void Retrieve_Button(object sender, RoutedEventArgs e)
        {
            RetriveWindow Ret = new RetriveWindow();
            Ret.Show();
            Close();
            MessageBox.Show("Student details is retrived");

        }

        private void txt_CourseEnroll_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {

        }
    }
}
```

4 Retrieve

```
using CourseWorkSample;
using DataHandler;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.OleDb;
using System.Globalization;
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 APPLICATION_DEVELOPMENT
{
    /// <summary>
    /// Interaction logic for RetriveWindow.xaml
    /// </summary>
    public partial class RetriveWindow : Window
    {
        public RetriveWindow()
        {
            InitializeComponent();
            LoadStudentData();// method call

        }

        private void LoadStudentData()
        {

            if (System.IO.File.Exists(@"F:\StudentReport.xml"))
            {
                var handler = new Handler();


                var dataSet = new DataSet();

                dataSet.ReadXml(@"F:\StudentReport.xml");

                DataTable dtStdReport = new DataTable();
                dtStdReport = dataSet.Tables[0];
                grtd_Retrive.DataContext = dtStdReport.DefaultView;
            }

        }

        private void student_Date_regis_Click(object sender, RoutedEventArgs e)
        {

            RegisterInDate Date = new RegisterInDate();
            Date.Show();
            Close();
```

```
            MessageBox.Show("Student details registered according to date is
depicted!!");
        }

        private void student_Name_regis_Click(object sender, RoutedEventArgs e)
        {
            RegisterInName Name = new RegisterInName();
            Name.Show();
            Close();
            MessageBox.Show("Student details registered according to name is
depicted!!");
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {

            Chart_Diagram chart = new Chart_Diagram();
            chart.Show();
            Close();
            MessageBox.Show("Student details registered on weekly basis is
shown in Pie-chart!!");
        }

        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
            WeeklyStudentReport report = new WeeklyStudentReport();
            report.Show();
            Close();
            MessageBox.Show("Weekly Report is depicted!!");
        }

        private void CSV_FILE_Click(object sender, RoutedEventArgs e)
        {

            Microsoft.Win32.OpenFileDialog dlg = new
Microsoft.Win32.OpenFileDialog();
            dlg.DefaultExt = ".csv";
            Nullable<bool> result = dlg.ShowDialog();

            if (result == true)
            {

                DataTable tableStd = ExtractsReportFromCSV(dlg.FileName, true);

                DataSet dataSet = new DataSet();
                dataSet.ReadXml(@"F:\StudentReport.xml");
                foreach(DataRow dr in tableStd.Rows)
                {
                    var newRow = dataSet.Tables["StudentReport"].NewRow();//New
data row on the smae table is created.
                    newRow["RegisNo"] = dr["RegisNo"];
                    newRow["Name"] = dr["Name"];
                    newRow["Address"] = dr["Address"];
                    newRow["ContactNo"] = dr["ContactNo"];
                    newRow["CourseEnroll"] = dr["CourseEnroll"];
                    newRow["RegistrationDate"] = dr["RegistrationDate"];
                    dataSet.Tables["StudentReport"].Rows.Add(newRow);

                }

dataSet.Tables["StudentReport"].WriteXml(@"F:\StudentReport.xml");
```

```
                //var datasett = new DataSet();
                //datasett.ReadXml(@"F:\StudentReport.xml");
                DataTable table = dataSet.Tables["StudentReport"];
                grtd_Retrive.DataContext = table.DefaultView;

                // lblWindowName.Content = "DataTable showing CSV files.";

            }

        }

        static DataTable ExtractsReportFromCSV(string path, bool
isFirstRowHeader)
        {
            string header = isFirstRowHeader ? "Yes" : "No";

            string pathAddress = System.IO.Path.GetDirectoryName(path);
            string fileName = System.IO.Path.GetFileName(path);

            string sql = @"SELECT * FROM [" + fileName + "]";

            using (OleDbConnection join = new OleDbConnection(
                      @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
pathAddress +
                      ";Extended Properties=\"Text;HDR=" + header + "\""))
            using (OleDbCommand command = new OleDbCommand(sql, join))
            using (OleDbDataAdapter adapter = new OleDbDataAdapter(command))
            {
                DataTable dataTable = new DataTable();
                dataTable.Locale = CultureInfo.CurrentCulture;
                adapter.Fill(dataTable);
                return dataTable;
            }
        }
        private void back_Click(object sender, RoutedEventArgs e)
        {
            MainWindow back = new MainWindow();
            back.Show();
            Close();

        }

    }
}
```

5 Sorting By Date

```
using DataHandler;
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 APPLICATION_DEVELOPMENT
{
    /// <summary>
    /// Interaction logic for RegisterInDate.xaml
    /// </summary>
    public partial class RegisterInDate : Window
    {
        public RegisterInDate()
        {
            InitializeComponent();
            LoadStudentData();
        }

        private void LoadStudentData()
        {

            if (System.IO.File.Exists(@"F:\StudentReport.xml"))
            {
                var handler = new Handler();


                var dataSet = new DataSet();

                dataSet.ReadXml(@"F:\StudentReport.xml");

                DataTable dtStdReport = new DataTable();
                dtStdReport = dataSet.Tables[0];
                dtStdReport.DefaultView.Sort = "RegistrationDate ASC";
                RegisInDate.DataContext = dtStdReport.DefaultView;
            }
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {

            RetriveWindow back = new RetriveWindow();
            back.Show();
            Close();

        }
    }

}
```

6 Sorting By Name:

```
using DataHandler;
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 APPLICATION_DEVELOPMENT
{
    /// <summary>
    /// Interaction logic for RegisterInName.xaml
    /// </summary>
    public partial class RegisterInName : Window
    {
        public RegisterInName()
        {
            InitializeComponent();
            LoadStudentData();
        }


        private void LoadStudentData()
        {

            if (System.IO.File.Exists(@"F:\StudentReport.xml"))
            {
                var handler = new Handler();


                var dataSet = new DataSet();

                dataSet.ReadXml(@"F:\StudentReport.xml");

                DataTable dtStdReport = new DataTable();
                dtStdReport = dataSet.Tables[0];
                dtStdReport.DefaultView.Sort = "Name ASC";
                RegisInName.DataContext = dtStdReport.DefaultView;
            }
        }

        private void back_Click(object sender, RoutedEventArgs e)
        {
            RetriveWindow back = new RetriveWindow();
            back.Show();
            Close();
        }
    }
}
```

7 Weekly Report

```
using DataHandler;
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 APPLICATION_DEVELOPMENT
{
    /// <summary>
    /// Interaction logic for WeeklyStudentReport.xaml
    /// </summary>
    public partial class WeeklyStudentReport : Window
    {

        public WeeklyStudentReport()
        {
            InitializeComponent();
            weeklyStudentList();
        }

        private void weeklyStudentList() {
            var dataSet = new DataSet();

            dataSet.ReadXml(@"F:\StudentReport.xml");

            DataTable dtStdReport = dataSet.Tables[0];

            int Total_BBA = 0;
            int Total_BEIT = 0;
            int Total_BBS= 0;

            DataTable Week = new DataTable("WeekTable1");
            Week.Columns.Add("Courses Enrolled", typeof(String));
            Week.Columns.Add("Overall Student", typeof(int));


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

                String column = dtStdReport.Rows[i]["CourseEnroll"].ToString();

                if (column == "BBA")
                {
                    Total_BBA++;

                }
                else if (column == "BE.IT")
                {
                    Total_BEIT++;
```

```
                }
                else if (column == "B.B.S")
                {
                    Total_BBS++;

                }
            }
            Week.Rows.Add("BBA", Total_BBA);
            Week.Rows.Add("BE.IT", Total_BEIT);
            Week.Rows.Add("BBS", Total_BBS);


            StudentReport.DataContext = Week.DefaultView;


        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            RetriveWindow back = new RetriveWindow();
            back.Show();
            Close();
        }

        private void back_Click(object sender, RoutedEventArgs e)
        {
            RetriveWindow back = new RetriveWindow();
            back.Show();
            Close();
        }
    }
}
```

8 Chart Diagram

```
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.Controls.DataVisualization.Charting;
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 APPLICATION_DEVELOPMENT
{
    /// <summary>
    /// Interaction logic for Chart_Diagram.xaml
    /// </summary>
    public partial class Chart_Diagram : Window
    {
        public Chart_Diagram()
        {
```

```
            InitializeComponent();


            var dataSet = new DataSet();

            dataSet.ReadXml(@"F:\StudentReport.xml");

            DataTable dtStdReport = dataSet.Tables[0];

            int Total_BBA = 0;
            int Total_BEIT = 0;
            int Total_BBS = 0;

            DataTable Week = new DataTable("WeekTable1");
            Week.Columns.Add("Courses Enrolled", typeof(String));
            Week.Columns.Add("Overall Student", typeof(int));


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

                String column = dtStdReport.Rows[i]["CourseEnroll"].ToString();

                if (column == "BBA")
                {
                    Total_BBA++;

                }
                else if (column == "BE.IT")
                {
                    Total_BEIT++;

                }
                else if (column == "B.B.S")
                {
                    Total_BBS++;

                }
            }
            Week.Rows.Add("BBA", Total_BBA);
            Week.Rows.Add("BE.IT", Total_BEIT);
            Week.Rows.Add("BBS", Total_BBS);

            ((PieSeries)Chart_Diagram1).ItemsSource =
         new KeyValuePair<string, int>[]{
            new KeyValuePair<string,int>("BBA", Total_BBA),
            new KeyValuePair<string,int>("BE.IT", Total_BEIT),
            new KeyValuePair<string,int>("BBS", Total_BBS) };

        }

        private void back_Click(object sender, RoutedEventArgs e)
        {
            RetriveWindow back = new RetriveWindow();
            back.Show();
            Close();
        }
    }
}
```