Marking Scheme

Informatics College Pokhara

informatics
college pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: **Bijendra Parsi**
London Met ID: Enter ID Here

Submitted To: Ishwor Sapkota
Module Leader

Component Grade and Comments	
A. Implementation of Application	
User Interface and proper controls used for designing	Design is properly done and in mess
Manual data entry or import from csv	data types not taken care of and not properly executed functionally.
Data Validation	No validation at all
Enrollment Report & weekly report in tabular format	very poorly executed reports and data not shown accurately
Course wise enrollment report & Chart display	Very poorly designed and only contains one report format with in appropriate data
Algorithm used for sorting & proper sorting of data	Default sorting provided by .net is used
B. Documentation	
User Manual for running the application	User Manual is average. Includes description for all interfaces

Application architecture & description of the classes ad methods sued	very poorly explained.
Flow chart, algoriathms and data sctructures used	very poorly explained and no diagramatic representation
Reflective essay	Very poorly written

C. Programming Style

Clarity of code,Popper Naming convention & comments	very poorly written code and no comments at all
System Usability	unusable system

Overall Grade:	**F**	**F**

Overall Comment:

Code should be self explainable with less comments. Need some proper naming of the component and require to add comments on required area.

In overall the code is working and all the functionality seems working and system can be used

Informatics College Pokhara

Application Development

CS6004NP

Coursework 1

Submitted By:

Student Name: Bijendra Pasi

London Met ID: 18029130

Group: L3C2

Date: 10-Jan-2020

Submitted To:

Mr. Ishwor Sapkota

Contents

Table of Figure

Introduction

This coursework is about growing and imposing a C# desktop application that helps college to control the student's record. This application helps the school team of workers to input the student's fundamental element manually such as Name, Adress, Contact and gadget automatically import that detail to a CSV file. The enrollment date and course chosen by using scholar is also managed via the system. The gadget generates daily and weekly record and type them via the Enrollment date and Name.

- Current Scenario

Large wide variety of schools has their own software to hold the record of students but some of them are still preserving their file in historical standard device which is Paper-Based System. Despite many of schools having the software, they are now not updated and lack the facets wished for Schools.

- Purposed System

The proposed gadget is digitized device which is specifically designed to overcome the troubles mentioned. The device ensures protection with presence of login section. With the use of Graphical User Interface, entry of facts and show of records has been made easier.

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 is “admin”. Only a valid username and password can provide access to the system.

Figure 1 login screen

On inputting the correct username and password, the system will the main form screen.

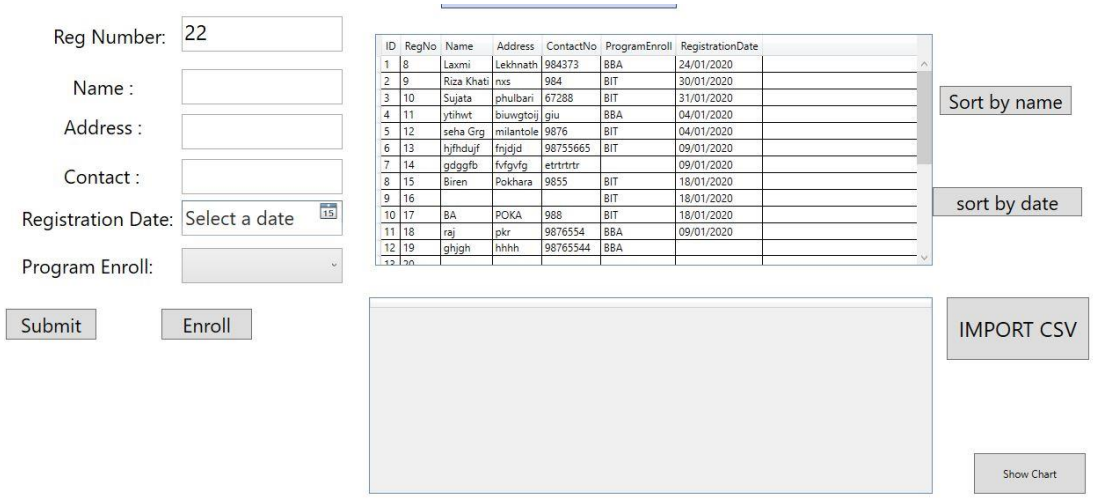


Figure 2 Main Screen

Entering student record and then saving it

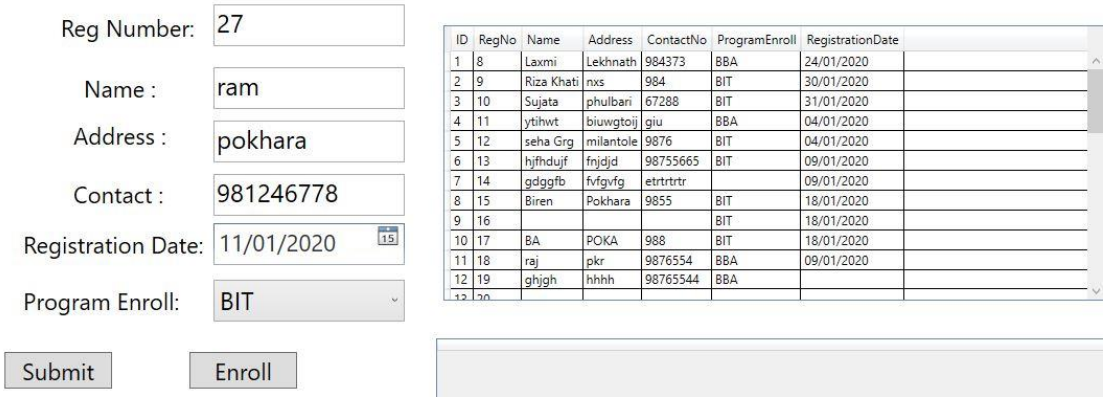


Figure 3 saving record of student

Importing the data to data grid

.

ID	RegNo	Name	Address	ContactNo	ProgramEnroll	RegistrationDate
1	8	Laxmi	Lekhnath	984373	BBA	24/01/2020
2	9	Riza Khati	nxs	984	BIT	30/01/2020
3	10	Sujata	phulbari	67288	BIT	31/01/2020
4	11	ytihwt	biuwgtoij	giu	BBA	04/01/2020
5	12	seha Grg	milantole	9876	BIT	04/01/2020
6	13	hjfhdujf	fnjdjd	98755665	BIT	09/01/2020
7	14	gdggfb	fvfgvfg	etrtrtrtr		09/01/2020
8	15	Biren	Pokhara	9855	BIT	18/01/2020
9	16				BIT	18/01/2020
10	17	BA	POKA	988	BIT	18/01/2020
11	18	raj	pkr	9876554	BBA	09/01/2020
12	19	ghjgh	hhhh	98765544	BBA	

Figure 4 importing to data grid

In the Report button present in the Main Window, Display Report gets opened which has four more buttons.

Figure 5 display report window

Sorting the data according to name by clicking the Sort by Name button is done below:

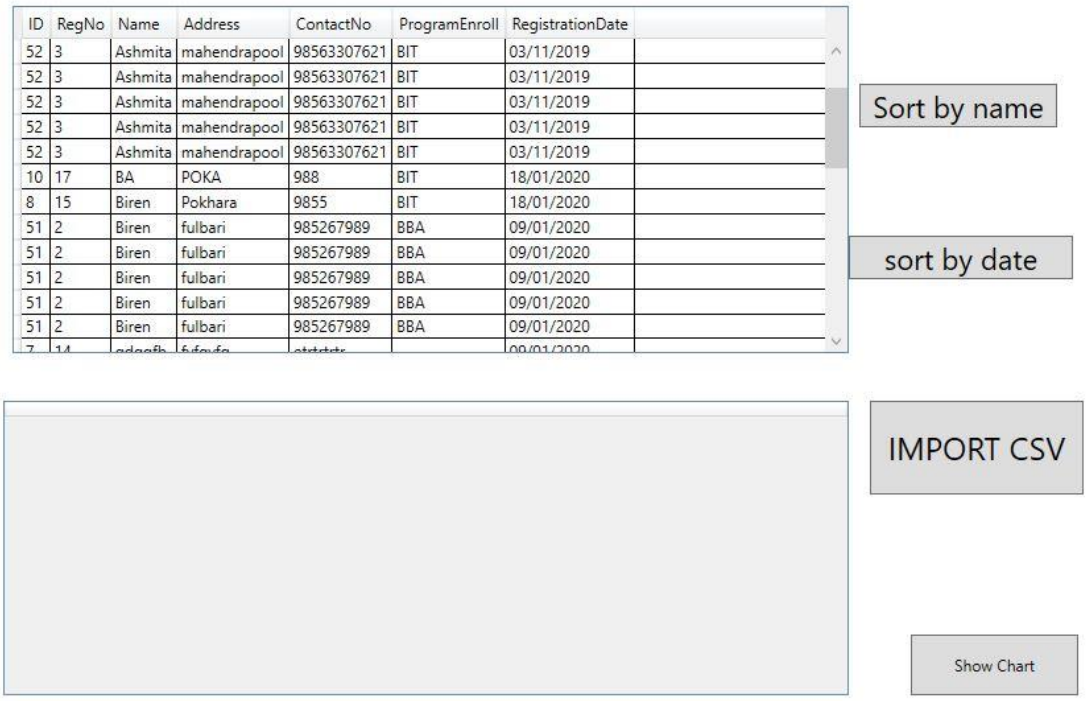


Figure 6 sort by name

Sorting the data according to name by clicking the Sort by date:

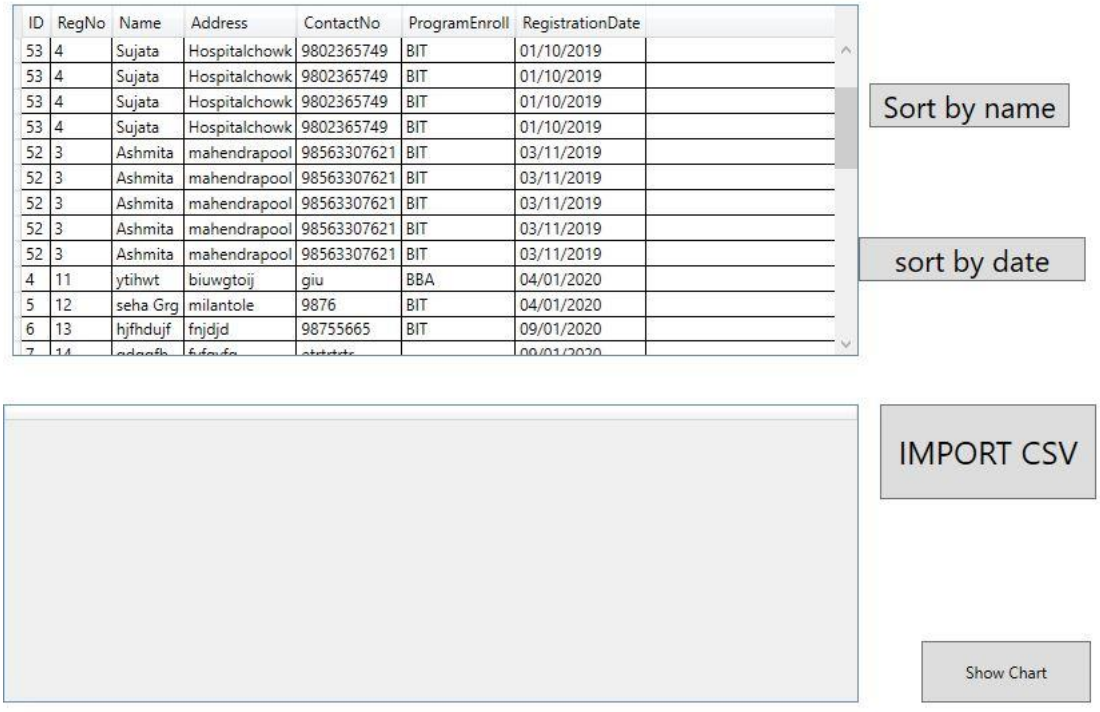


Figure 7 sorting by date

After clicking the Enroll button, we can see the total number of students enrolled in each Course. It is shown in the following screenshot.

Figure 8 weekly report

After clicking the Import button, we can go inside my computer for inserting data from csv file. It is shown in the following screenshot.

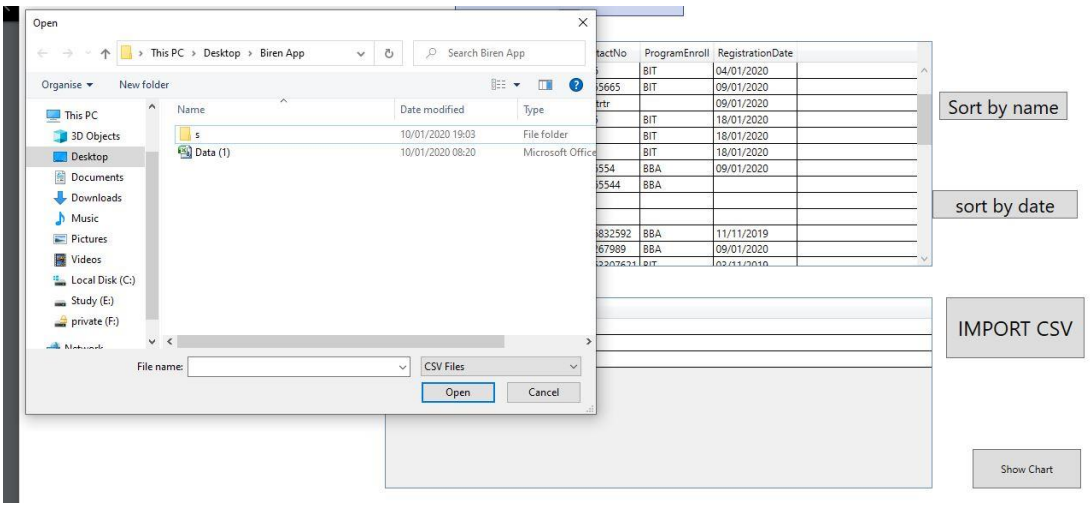


Figure 9 import from csv

After clicking the data file and opening the current file of csv, All record of students will directly import inside the data grid.It is shown in the following screenshot.

Figure 10 import data from csv file to datagrid

After clicking show chart button new window will open and creates chart where all the student of BBA and BIT as shown in the figure below:

Figure 11 create chart of students

Journal Articles

i. The management and provision of facts about the educational system is an vital section of wonderful management of the educational manner in the institutes of higher education. In this paper the requirements of a reliable student management system are analyzed, fashioned a use-case mannequin of student information management system, designed and implemented the architecture of the application. Regarding the implementation process, contemporary procedures have been used to increase and installation a reliable online software in cloud computing environments
specifically (Alameri, 2017).

ii. Despite the advantages of college administration records structures (SMIS), the notion of data-driven school lifestyle failed to materialize for many instructional institutions. Challenges posed by way of the fantastic of records in the big data era have averted many faculties from realizing the actual plausible of the SMIS. The paper analyses the uses, features, and inhibiting elements of SMIS. The paper proposes a five-phase conceptual model that aid administrators with making timely, first-rate decisions. The paper enriches the theoretical landscape of SMIS utilization in the technology of big data and lays a foundation for the future through establishing an academic decisionmaking mannequin (Forrester, 2019).Most of the Academic institutions face difficulty in managing records of students, attendance, accounts, admissions, etc., and track the information of their interest as they still rely on paperwork and manual processes. A web-based school management system will reduce the manual work by deploying centralized software incorporated with various loosely coupled services which interact with each other to address above mentioned issues and improves the communication between management and student/guardian CS6004NA Application Development Nirakar Sigdel 10 through notifications via email, SMS and push messages. As it is a server-

side enterprise application it is designed to support desktop browsers, mobile browsers and native mobile applications. The use of micro-service architecture and ReST (Representational State Transfer) architecture makes it easy for designing and developing loosely coupled web services, (K, 2019).

iii. System usability is one of the key elements that should be focused on, especially during the design and test phases of a system, because it provides feedback to system administrators in order to improve the system. In the literature, System Usability Scale (SUS) is widely used as the gold standard method to evaluate system usability. Today, machine learning, which is one of the subfields of artificial intelligence, also provide new perspectives on the evaluation of system usability. In this study, it is aimed to predict usability of a Student Information System (SIS) by using machine learning techniques. In the study method, the CrossIndustry Standard Process for Data Mining (CRISP-DM) steps have been followed. Analysis are performed on two different datasets namely “sus0” and “sus1”. “sus0” dataset is consisted of demographic characteristics (age, gender, department) of 324students using a SIS of a foundation university in Turkey, also their responses to the Turkish version of the SUS (SUS-TR). “sus 1” includes only responses to the SUS-TR. C4.5 Decision Tree Algorithm, Naive Bayes Classifier and k-Nearest Neighbor Algorithm are used to create models and performance of the models are evaluated. In the analysis with 80% to 20% hold-out method, the best performance was obtained on the “sus0” data set with k-Nearest Neighbor Algorithm (accuracy= 0.698, F-measure = 0.796 for k = 20) (Kartal, 2019).

iv. The mission of the Student Information Management machine is to create an integrated facts technology environment for students, HOD, faculty, workforce and administration. Our purpose is to CS6004NA Application Development Nirakar Sigdel 11 focal point on offerings and integration for give up users. It is a web-based self-service environment for students, potential students, and

employees; administrative transaction processing environment for every year admissions; an informative surroundings for all degrees of college and group of workers to do reporting, information extraction and facts analysis. It is frequently useful and used for instructional enterprises to control student facts which additionally helps all man or woman associated facts for less complicated navigation on each day basis. It provides skills for coming into pupil take a look at and other evaluation scores, building pupil schedules, tracking scholar attendance and managing many different student-related statistics desires in a college. Our easy-to-use, built-in college administration software would be used to decrease time spent on administrative tasks, as to listen on other skillful realistic activities different than book worming. It can accept, manner and generate reviews at any given point of time precisely (Budhrani, 2018).

System Architecture

Architecture Diagram

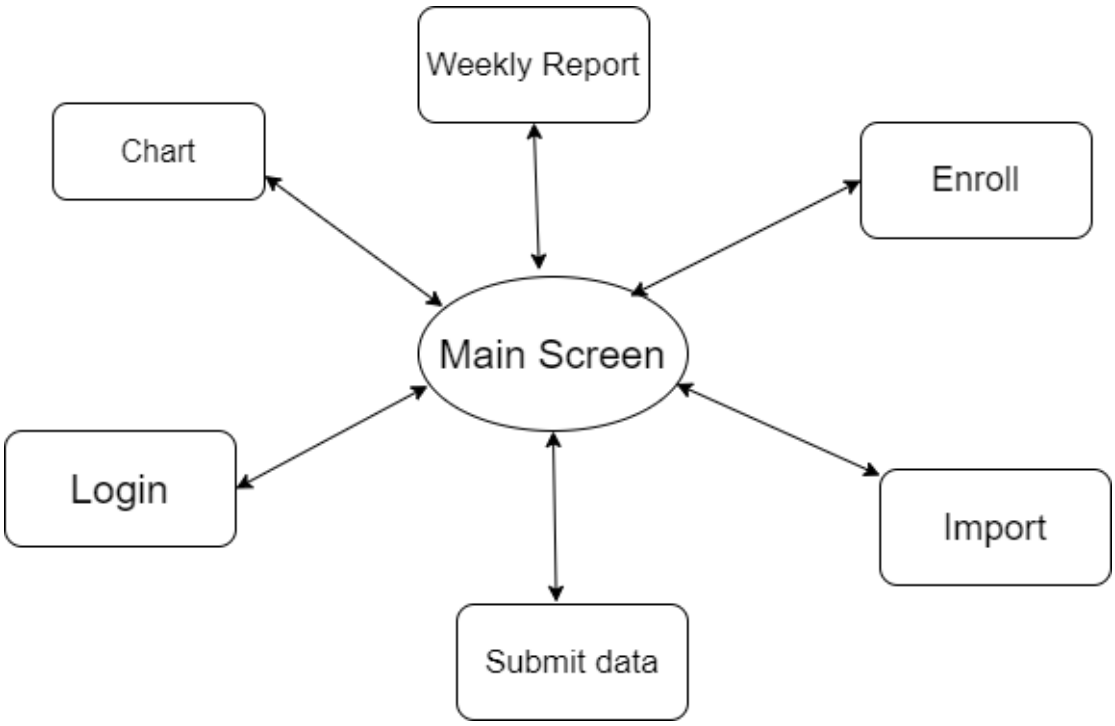


Figure 12 Architecture Diagram

Class Diagram

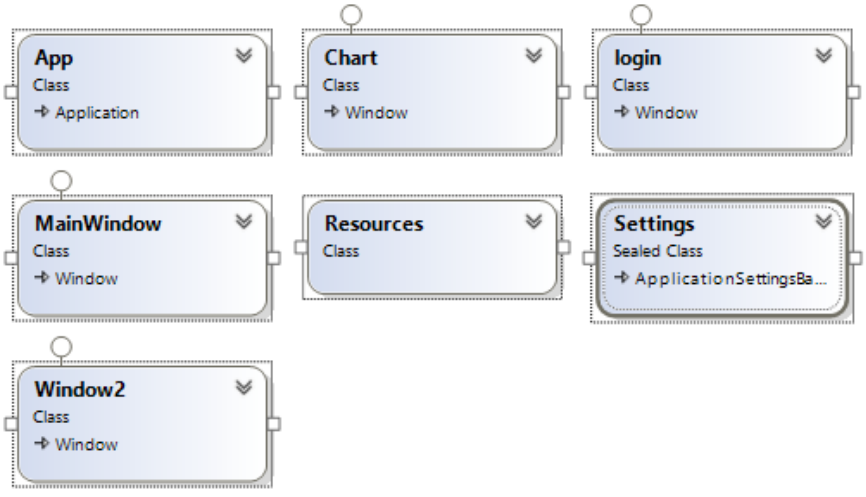


Figure 13 class diagram

Classes, Properties and Methods

Classes and Properties

This application consists of the following main classes:

Login.xaml: This is the important page of the desktop application. After strolling the utility Login.xaml type is known as and the login form is displayed the place the user enters the username and password in the respective text-box. The consumer has to login to enter the main page of desktop application and if the consumer leaves any text-boxes empty or wrong username or password then the pop-up message will be displayed.

MainWindow.xaml: After logging into the system, MainWindow.xaml type is called. In this type the tabs controller is used for changing the tabs i.e. Save, Import, Report. In the Save Button, the student's important points manually entered by person is saved in XML file and the Import Button shows the details of students in records grid view. The Report button opens the new window DisplayReport.xaml. The Clear button clears the structure and Exit button exits the window.

DisplayReport.xaml: After clicking on the Report button of MainWindow, this category receives opened the place 5 buttons i.e. Retrieve Data, Sort through Name, Sort by Date, Weekly Report and Create chart; retrieves data, Sort through Name and Date, generates the weekly record and creates the chart respectively.

Methods

This application consists of following methods:

Methods	Description
private void Button_Click(object sender, RoutedEventArgs e)	When Button_Click is clicked, it will login into the application; only if the admin and password matches.
private void Exit_Click(object sender, RoutedEventArgs e)	When Exit_Click is clicked, it will close the application.
private void btnSave_Click(object sender, RoutedEventArgs e)	When btnSave_Click is clicked, it saves the data in XML file.

private void AddSampleData(DataSet dataSet)	It adds the sample data.
private void btnImport_Click(object sender, RoutedEventArgs e)	It imports the data into data grid.
private void btnClear_Click(object sender, RoutedEventArgs e)	It clears the uncleared data in the student record form
private void btnExit_Click(object sender, RoutedEventArgs e)	It exits the window.
private void btnReport_Click(object sender, RoutedEventArgs e)	It opens another window naming DisplayReport
privatevoid buttonRetrive_Click(object sender, RoutedEventArgs e)	It retrieves the data and shows in the data grid of DisplayReport
private void buttonSName_Click(object sender, RoutedEventArgs e)	It sorts the data according to Name in ascending order.
private void buttonSD_Click(object sender, RoutedEventArgs e)	It sorts the data according to Name in ascending order.
private void buttonWeekly_Click(object sender, RoutedEventArgs e)	It shows the total number of students added in respective course in a week.
private void buttonChart_Click(object sender, RoutedEventArgs e)	It leads to the new window to show the chart.

Table 1 method description

Flowchart for Report

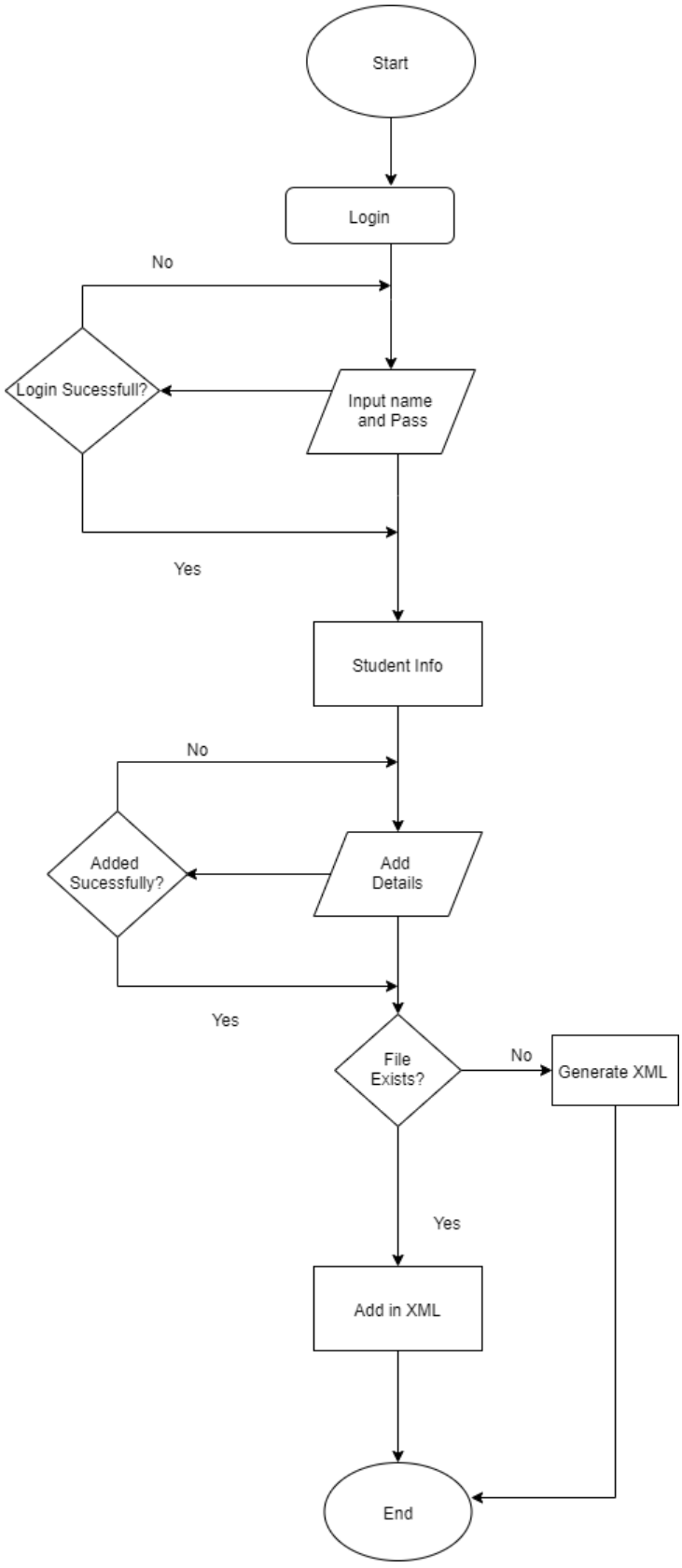


Figure 14 flow chart

Algorithm of Report

Steps

1. Start
2. Check if student details are there or not.
3. If No, display error message and exit.
4. If Yes, Read the file.
5. Find the count.
6. Display Count.
7. Stop

Sorting Algorithm

Bubble sort is a simple sorting algorithm which works on the comparisonbased algorithm. In this algorithm, the adjacent elements are compared and the elements are swapped if they are not in order. This algorithm is usually not suitable for large data sets as its average and worst-case complexity are of O(n2) where n is the number of items.

Working Methods:

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

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

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

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

The new array should look like this −

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

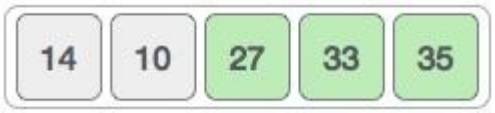
Then we move to the next two values, 35 and 10.

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

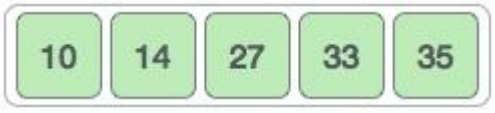
We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this −

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

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



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



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

Reflection

This coursework is about creating computer utility in C# for Student Information. I bought know-how about C# in my module, Application Development. It used to be my first-time using C# language for growing a computing device application. As it was once my first-time the usage of C#, I had many confusions but with the assist of module leader, I was able to increase desktop application using C#. C# is an object oriented and generalpurpose programming language created and developed via Microsoft. As C# is a high-level language, it is similar to Java and C++. Since, it has resemblance with Java, various methods had been effortless to develop. The laptop utility is developed in Microsoft Visual Studio which is an Integrated Development Environment (IDE). As I had used Microsoft Visual Studio previously, I felt less problems related to working on it but still had some troubles which got solved all through tutorial training and I was in a position to boost the application. While working for this project I discovered Microsoft Visual Studio a lot less difficult to enforce an utility as it

mechanically analyzes code to factor out blunders and offer suggestions.

Problem Experienced

While creating this computing device application, many issues have been faced. With the help of tutorial classes, it grew to become convenient to start the project. There was once a hassle working with toolbox and solution explorer for creating GUI for this application. Many mistakes had been confronted whilst importing CSV file. Generating document and showing the pie chart used to be a challenging section of this application.

Solution

The hassle confronted while growing this venture was once solved with the assist of a variety of web sites and video tutorials. Login and producing CSV file used to be executed with the help of example, which was executed in classroom. The lecture and tutorial training helped me in solving blunders and to work effectively in Visual Studio Platform. I performed many researches, to strengthen pie-chart. Thus, in this way, I completed this coursework efficiently.

Conclusion

As per the scenario, a computing device application for Student Information used to be developed with the help of C# programming language. Those codes have been validated and is successfully working although quite a few troubles passed off whilst growing this application. Through this coursework, it helped me recognize C# which will help me in future. This undertaking also helped me extra about the computing device application. As this task is efficaciously developed and tested and has consumer friendly GUI, this undertaking can be easily used in school in order to shop pupil details extra efficiently.

Bibliography

Forrester, V., 2019. School Management Information Systems. [Online] Available at: https://www.academia.edu/38700223/SCHOOL_MANAGEMENT_INORMATION_SYSTEMS_CHALLENGES_TO_EDUCATIONAL_DECISN-_MAKING_IN_THE_BIG_DATA_ERA [Accessed 10 01 2020].

Budhrani, D., 2018. Student Information Management System. [Online] Available at: https://www.ijedr.org/papers/IJEDR1801002.pdf [Accessed 10 01 2020].

Alameri, I., 2017. Development of Student Information Management System based on Cloud Computing Platform. [Online] Available at:

Kartal, E., 2019. Predicting Usability of a Student Information System by Using Machine Learning Techniques. [Online] Available at: https://www.academia.edu/39894115/Predicting_Usability_of_a_Student_Information_System_by_Using_Machine_Learning_Techniques [Accessed 10 01 2020].

K, S., 2019. Student Management System. [Online] Available at: https://www.ijert.org/student-management-system [Accessed 10 01 2020].

TutorialsPoint, n.d. Bubble Sort Algorithm. [Online] Available at: https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm [Accessed 10 01 2020]

https://www.researchgate.net/publication/319881143_Development_of_Student_Information_Management_System_based_on_Cloud_Computing_Platform [Accessed 10 01 2020].

Appendix

Login.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using AppDevCoursewrk;

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

        private void Btnlogin_Click(object sender, RoutedEventArgs e)
        {
            string user = txtuser.Text;
            string pass = txtpass.Password;
```

```
            if (user =="Admin" && pass == "admin")
            {
                this.Hide();
                MainWindow mainWindow = new MainWindow();
                mainWindow.Show();
            }
        }
    }
}
```

MainWindow.xaml.cs

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

namespace AppDevCoursewrk
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        private DataTable buffer;
```

```
public MainWindow()
{
    InitializeComponent();
    Startup();

    txtRegNo.Text = read_from_file();

    LoadStudentData();



}

public void Startup()
{
    //var handler = new Handler();
    //var dataSet = handler.CreateDataSet();
    //AddSampleData(dataSet);
    //dataSet.WriteXmlSchema(@"D:\StudentCWSchema.xml");
    //dataSet.WriteXml(@"D:\StudentCWData.xml");

    //var dataSet = new DataSet();
    //dataSet.ReadXmlSchema(@"D:\StudentCWSchema.xml");
    //dataSet.ReadXml(@"D:\StudentCWData.xml");
```

```
        //var i = 0;
    }

    private void AddSampleData(DataSet dataSet)
    {

        var dr = dataSet.Tables["Course"].NewRow();
        dr["Name"] = "BBA";
        dr["DisplayText"] = "BBA Hons";
        dataSet.Tables["Course"].Rows.Add(dr);

        dr = dataSet.Tables["Course"].NewRow();
        dr["Name"] = "Network & Communication";
        dr["DisplayText"] = "BCA Network";
        dataSet.Tables["Course"].Rows.Add(dr);

        dr = dataSet.Tables["Course"].NewRow();
        dr["Name"] = "Programming & Application Development";
        dr["DisplayText"] = "BSc CSIT Application Development";
        dataSet.Tables["Course"].Rows.Add(dr);

        dr = dataSet.Tables["Student"].NewRow();
        dr["Name"] = "Ishwor Sapkota";
        dr["Address"] = "Kathmandu";
```

```
        dr["ContactNo"] = "9851220845";
        dr["CourseEnroll"] = 1;
        dr["RegistrationDate"] = DateTime.Today.AddDays(-2);
        dataSet.Tables["Student"].Rows.Add(dr);

        dr = dataSet.Tables["Student"].NewRow();
        dr["Name"] = "Samyam Sapkota";
        dr["Address"] = "Kathmandu";
        dr["ContactNo"] = "9851220846";
        dr["CourseEnroll"] = 2;
        dr["RegistrationDate"] = DateTime.Today.AddDays(-1);
        dataSet.Tables["Student"].Rows.Add(dr);

        dr = dataSet.Tables["Student"].NewRow();
        dr["Name"] = "Safal Sapkota";
        dr["Address"] = "Kathmandu";
        dr["ContactNo"] = "9851220847";
        dr["CourseEnroll"] = 3;
        dr["RegistrationDate"] = DateTime.Today.AddDays(-3);
        dataSet.Tables["Student"].Rows.Add(dr);

    }

    private void AddSampleDataforStd(DataSet dataSet)
```

```
{

    var dr = dataSet.Tables["Course"].NewRow();
    dr["Name"] = "BBA";
    dr["DisplayText"] = "BBA Hons";
    dataSet.Tables["Course"].Rows.Add(dr);

    var dr1 = dataSet.Tables["Student"].NewRow();
    dr1["Name"] = txtName.Text;
    dr1["Address"] = txtAddress.Text;
    dr1["ContactNo"] = txtContact.Text;
    dr1["ProgramEnroll"] = combo.Text;
    dr1["RegistrationDate"] = DateTime.Today.AddDays(-2);
    dataSet.Tables["Student"].Rows.Add(dr1);



}

private void AppendStdReport(DataSet dataSet)
{
    if (File.Exists(@"C:\Appxml\StudentReport.xml"))
    {
        var handler = new Handler();
```

```
dataSet.Tables["StudentReport"].ReadXml(@"C:\Appxml\StudentReport.xml")
;

            var dr2 = dataSet.Tables["StudentReport"].NewRow();
            dr2["RegNo"] = txtRegNo.Text;
            dr2["Name"] = txtName.Text;
            dr2["Address"] = txtAddress.Text;
            dr2["ContactNo"] = txtContact.Text;
            dr2["ProgramEnroll"] = combo.Text;
            dr2["RegistrationDate"] = txtdate.Text;
            dataSet.Tables["StudentReport"].Rows.Add(dr2);


dataSet.Tables["StudentReport"].WriteXml(@"C:\Appxml\StudentReport.xml")
;
          }
          else {

dataSet.Tables["StudentReport"].WriteXml(@"C:\Appxml\StudentReport.xml")
;
              AppendStdReport(dataSet);
          }
      }
      private void Button_Click_1(object sender, RoutedEventArgs e)
      {
```

```
            var handler = new Handler();
            var dataSet = handler.CreateDataSet();
            AddSampleDataforStd(dataSet);
            AppendStdReport(dataSet);

            var regno = txtRegNo.Text;
            var name = txtName.Text;
            dataSet.WriteXmlSchema(@"C:\Appxml\StudentCWSchema1.xml");
            dataSet.Tables["Student"].WriteXml(@"E:\APPXML\" + name + "CWData" + regno + ".xml");

            dataSet.Tables[2].WriteXml(@"C:\Appxml\StudentReport.xml");

            write_to_file(txtRegNo.Text);

            txtRegNo.Text = read_from_file();

            ClearControls();
            LoadStudentData();
        }

        private void write_to_file(string text)
        {
```

```
        File.WriteAllText(@"C:\Appxml\count.txt", text);



    }
    private string read_from_file()
    {
        /*
        string text = System.IO.File.ReadAllText(@"C:\Appxml\count.txt");
        int i;
        i = int.Parse(text.ToString());
        i = i + 1;
        return i.ToString();*/
        int i = 1;
        if (File.Exists(@"C:\Appxml\count.txt"))
        {
            string text = File.ReadAllText(@"C:\Appxml\count.txt");
            i = int.Parse(text.ToString());
            i = i+1;
        }
        else {
            //File.WriteAllText(@"E:\APPXML\count.txt", "text");
        }
        return i.ToString();
```

```
    }

    private void ClearControls()
    {
        txtName.Text = "";
        txtAddress.Text = "";
        txtContact.Text = "";
    }

    private void LoadStudentData()
    {

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



            var dataSet = new DataSet();

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

            DataTable dtStdReport = new DataTable();
            dtStdReport = dataSet.Tables[0];
            grdStd.DataContext = dtStdReport.DefaultView;
```

```
        }


    }


    private void Button_Click(object sender, RoutedEventArgs e)
    {
        var dataSet = new DataSet();
        dataSet.ReadXml(@"C:\Appxml\StudentReport.xml");
        DataTable dtStdReport = dataSet.Tables[0];

        int total_BIT = 0;
        int total_BBA = 0;

        DataTable dt = new DataTable("newTable");
        dt.Columns.Add("ProgramEnroll", typeof(string));
        dt.Columns.Add("Total Students", typeof(int));

        for (int i = 0; i < dtStdReport.Rows.Count; i++) {
            string col = dtStdReport.Rows[i]["ProgramEnroll"].ToString();
            if (col == "BIT")
            {
```

```
            total_BIT++;
        }
        else if (col == "BBA")
        {
            total_BBA++;
        }

    }
    dt.Rows.Add("BBA", total_BBA);
    dt.Rows.Add("BIT", total_BIT);
    grdreport.DataContext = dt.DefaultView;


}


private void Srtname_Click(object sender, RoutedEventArgs e)
{
    var dataSet = new DataSet();
    dataSet.ReadXml(@"C:\Appxml\StudentReport.xml");
    DataTable DataTable = dataSet.Tables["StudentReport"];
    DataTable.DefaultView.Sort = "Name Asc";
    grdStd.DataContext = DataTable.DefaultView;
}
```

```
private void SortBtn_Click(object sender, RoutedEventArgs e)
{
    var dataSet = new DataSet();
    dataSet.ReadXml(@"C:\Appxml\StudentReport.xml");
    DataTable DataTable = dataSet.Tables["StudentReport"];
    DataTable.DefaultView.Sort = "RegistrationDate Asc";
    grdStd.DataContext = DataTable.DefaultView;
}
private void DataShow()
{
    string dataXMLFile = @"C:\Appxml\StudentReport.xml";
    System.Data.DataSet dataset = new DataSet();
    dataset.ReadXml(dataXMLFile);

    buffer = new DataTable("dt");
    buffer.Columns.Add("RegNo", typeof(string));
    buffer.Columns.Add("Name", typeof(string));
    buffer.Columns.Add("Address", typeof(string));
    buffer.Columns.Add("ContactNo", typeof(string));
    buffer.Columns.Add("ProgramEnroll", typeof(string));
    buffer.Columns.Add("RegistrationDate", typeof(string));

    for (int i = 0; i < dataset.Tables[0].Rows.Count; i++)
```

```
        {
            string s = dataset.Tables[0].Rows[i][5].ToString();
            DateTime dtime = DateTime.Parse(s);
            buffer.Rows.Add(
                dataset.Tables[0].Rows[i][0].ToString(),
                dataset.Tables[0].Rows[i][1].ToString(),
                dataset.Tables[0].Rows[i][2].ToString(),
                dataset.Tables[0].Rows[i][3].ToString(),
                dataset.Tables[0].Rows[i][4].ToString(),
                dtime.ToShortDateString());
        }
        DataView dataView = new DataView(buffer);
        grdreport.ItemsSource = dataView;
    }

    private void Button_Click_2(object sender, RoutedEventArgs e)
    {
        DataShow();
    }

    private void Button_Click_3(object sender, RoutedEventArgs e)
    {       var dataSet = new DataSet();
            dataSet.ReadXml(@"C:\Appxml\StudentReport.xml");
            OpenFileDialog openFileDialog = new OpenFileDialog();
```

```
openFileDialog.Filter = "CSV Files|*.csv";
openFileDialog.DefaultExt = ".csv";

bool? fileselect = openFileDialog.ShowDialog();
if (fileselect == true)
{
    string filePath = openFileDialog.FileName;
    //read all std from file code copy

    using (var reader = new StreamReader(filePath))
    {
        reader.ReadLine();
        while (!reader.EndOfStream)
        {
            var line = reader.ReadLine();
            var values = line.Split(',');
            var newRow = dataSet.Tables["StudentReport"].NewRow();
            newRow["ID"] = values[0];
            newRow["RegNo"] = values[1];
            newRow["Name"] = values[2];
            newRow["Address"] = values[3];
            newRow["ContactNo"] = values[4];
            newRow["ProgramEnroll"] = values[5];
            newRow["RegistrationDate"] = values[6];
```

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

            }

            dataSet.WriteXml(@"C:\Appxml\StudentReport.xml");
            grdStd.ItemsSource = dataSet.Tables["StudentReport"].DefaultView;

        }

    }
}

private void WindowsFormsHost_ChildChanged(object sender, System.Windows.Forms.Integration.ChildChangedEventArgs e)
{

}

private void Button_Click_4(object sender, RoutedEventArgs e)
{
```

```
            Chart mainWindow = new Chart();
            mainWindow.Show();
        }
    }
}
```

Chart.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;


namespace AppDevCoursewrk
{
    /// <summary>
    /// Interaction logic for Chart.xaml
    /// </summary>
    public partial class Chart : Window
    {
        public Chart()
        {
```

```
        InitializeComponent();


        var dataSet = new DataSet();


//dataSet.ReadXml(@"C:\\Users\\Ceema\\Desktop\\ApplicationDevelopmentCl
ass\\StudentReport.xml");
        dataSet.ReadXml(@"C:\Appxml\\StudentReport.xml");

        DataTable dtStdReport = dataSet.Tables[0];

        int total_BIT = 0;
        int total_BBA = 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]["ProgramEnroll"].ToString();
            if (col == "BIT")
            {
                total_BIT++;
            }
            else if (col == "BBA")
            {
                total_BBA++;
            }

        }

        dt.Rows.Add("BBA", total_BBA);
```

```
            dt.Rows.Add("BIT", total_BIT);

            bittop.Width = total_BIT*4;
            bbtop.Width = total_BBA*4;

            bbasts.Content = "Total BBA: " +total_BBA;
            bitsts.Content = "Total BIT" + total_BIT;

            //    ((ColumnSeries)chartEnroll).ItemsSource =
            //new KeyValuePair<string, int>[]{
            //new KeyValuePair<string,int>("BBA", total_BBA),
            //new KeyValuePair<string,int>("BIT", total_BIT)};



        }
    }
}
```