Marking Scheme

Informatics College Pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: **Prajal Rana**
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	User Interface is complete but not separated and have proper use of controls
Manual data entry or import from csv	appropriate use of data types but missing some properties required or missing CRUD operation
Data Validation	missing most of the validation
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 below average. Is textual only.

Application architecture & description of the classes ad methods sued	average work with very limited explanation of the classes and methods used
Flow chart, algoriathms and data sctructures used	average work with very limited explanation and missing diagramatic representation.
Reflective essay	Average work with un clear learnings, experience or findings.
C. Programming Style	
Clarity of code,Popper Naming convention & comments	very poorly written code and no comments at all
System Usability	very poorly developed application

Overall Grade:	**D**	**D**

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:		**Submitted To:**
Student Name:	Prajal Rana	Mr. Ishwor Sapkota
London Met ID:	17030731	
Group:	L3C2	Application Development
Date:	10-Jan-2020	

Table of Contents

Table of Figure

Table of Tables

1. Introduction

For this coursework we were given the task to develop a Student Management System fulfilling required functionalities. The system consist of three main courses Computing, Multimedia and Technology, and Networking allowing the user to register students based on those courses. Speaking of features the system consist features like registering new student to the system, retrieving student details, sorting student record in ascending order based on date and name, displaying weekly total student enrolment and chart displaying the no of students enrolled in each courses. For each process a different window is used for making the workflow easier and cleaner. The features mentioned above are well explained in different areas of the report.

1.1 Current Scenario

Even today schools or colleges still use traditional methods like paper based system to record student details which is found to be a risky way of recording student details. Loosing files results in data losses. Also, there are risk of those record getting stolen. There are some school that use computerized system for recording student details. However, those systems lack versatility and additional features suitable for today’s standards.

1.2 Proposed System

The proposed system aims to solve those issues through more versatile digital system. The graphical user interface has been designed keeping security and intuitive workflow in check. The system is a login system making it a safer option for data entry and retrieval.

2. User Manual

The screenshots below demonstrates how to operate the proposed system.

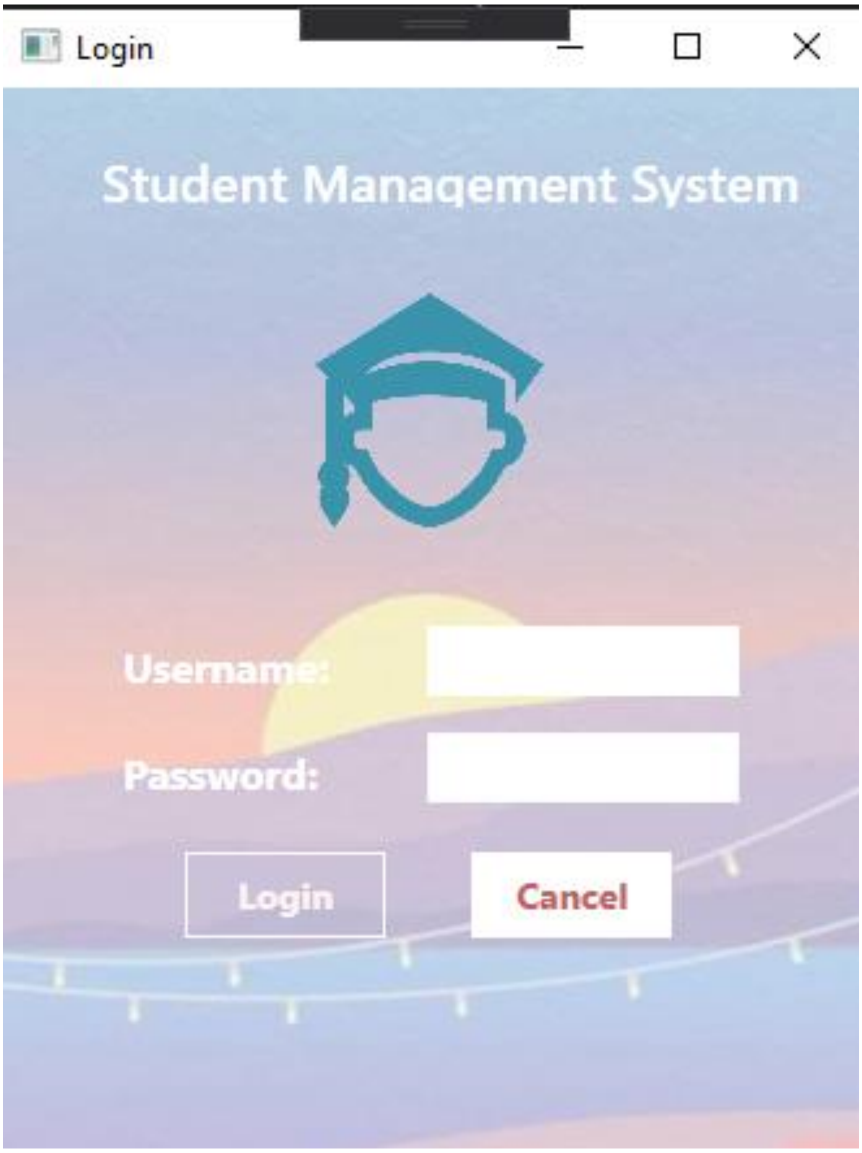


Figure 1: Login Window

When the system is launched the user is greeted by a login screen allowing authorized users are to access the system for data entry and manipulation. The username and password of the system is “admin”.

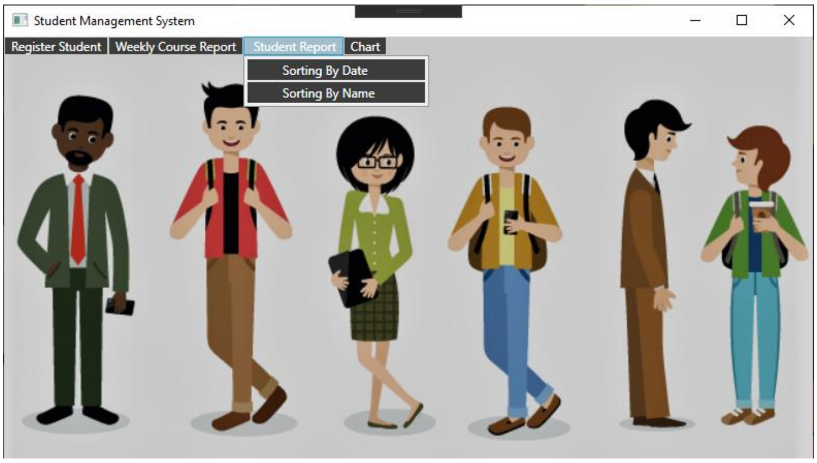


Figure 2: Student management system main window

After logging in to the system the end user is greeted by the main software, Student Management System. The inter consist of a dock panel with menu items Register Student, Weekly Course Report, Student Report and Chart. User can click on those menu items to go to a different window.

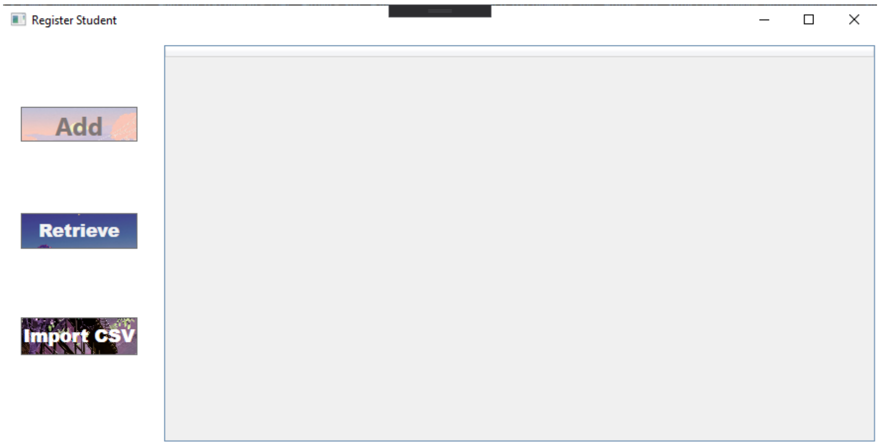


Figure 3: Register Student window

After selecting Register Student, Register Student window will open. The window consists of different buttons Add for registering new students, Import CSV for importing existing CSV file to the system and a Retrieve Button for displaying full student details in the data grid.

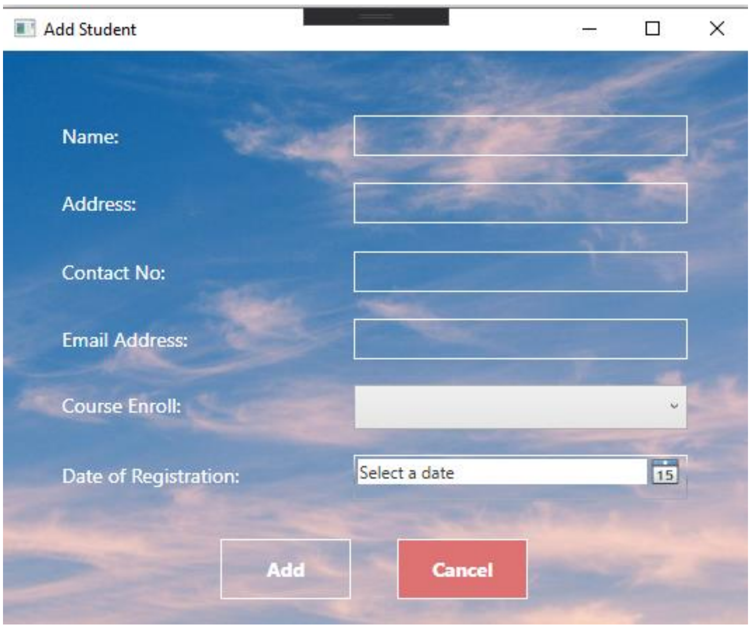
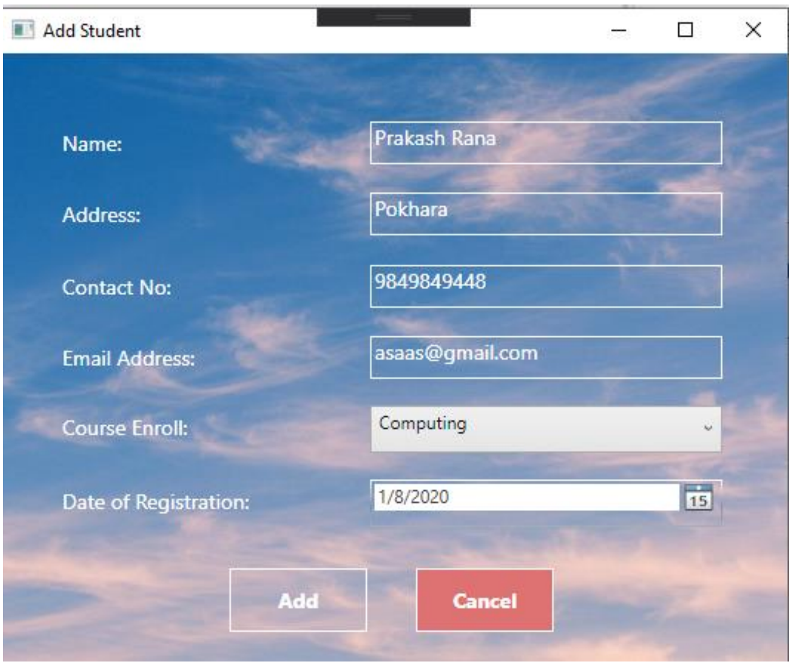


Figure 4: Add student window

When user click on the “Add” button in Register Student window, a new window Add Student opens. The user then can add the name, address, contact number, email address, course enroll and registration date for a new student entry on clicking the “Add” button. The user can also cancel the registration and terminate the window by clicking on the “Cancel” button. The figure below show data entry.



Name: Prakash Rana

Address: Pokhara

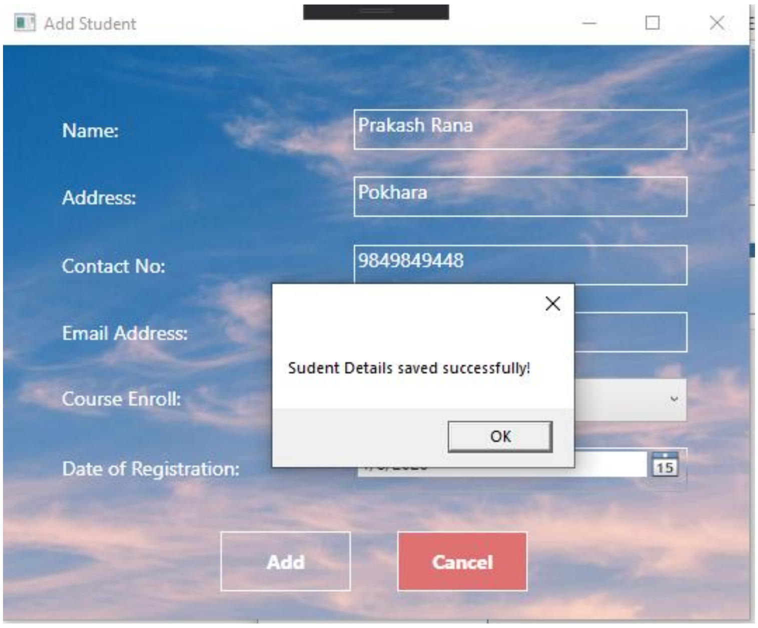
Contact No: 9849849448

Email Address: asaas@gmail.com

Course Enroll: Computing

Date of Registration: 1/8/2020

Figure 5: Student details entry



Name: Prakash Rana

Address: Pokhara

Contact No: 9849849448

Email Address:

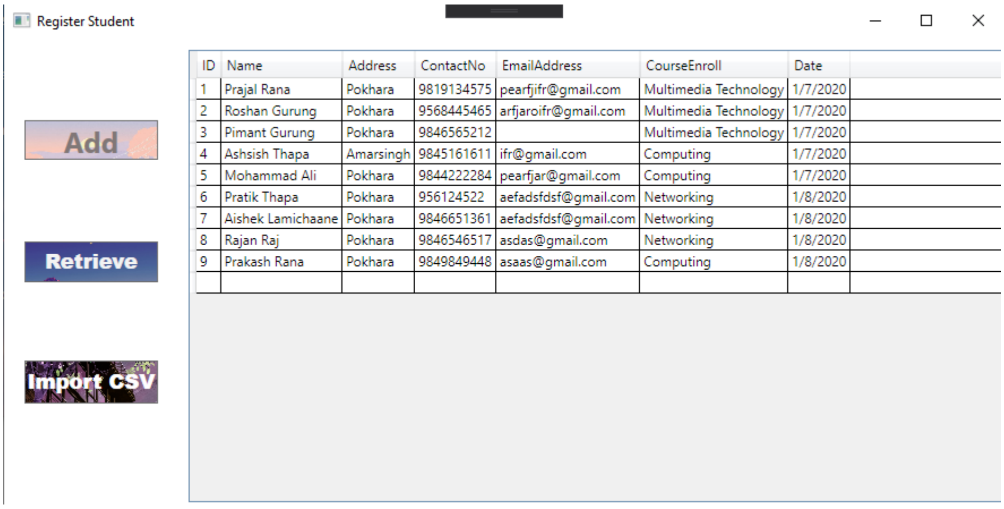
Course Enroll:

Date of Registration:

Sudent Details saved successfully!

Figure 6: Student details added

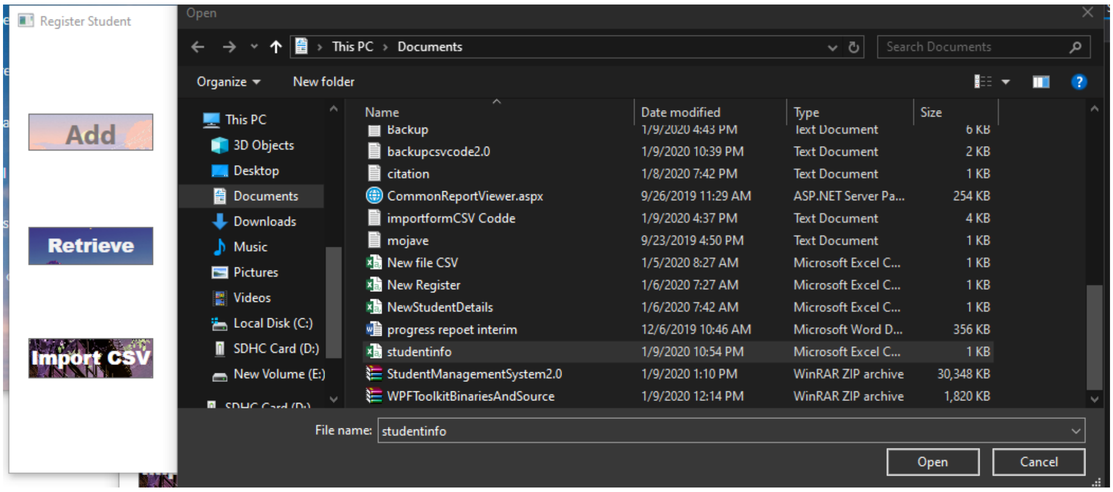
After clicking on the “Add” button a message is shown telling the data was successfully saved.



ID	Name	Address	ContactNo	EmailAddress	CourseEnroll	Date
1	Prajal Rana	Pokhara	9819134575	pearfjifr@gmail.com	Multimedia Technology	1/7/2020
2	Roshan Gurung	Pokhara	9568445465	arfjaroifr@gmail.com	Multimedia Technology	1/7/2020
3	Pimant Gurung	Pokhara	9846565212		Multimedia Technology	1/7/2020
4	Ashsish Thapa	Amarsingh	9845161611	ifr@gmail.com	Computing	1/7/2020
5	Mohammad Ali	Pokhara	9844222284	pearfjar@gmail.com	Computing	1/7/2020
6	Pratik Thapa	Pokhara	956124522	aefadsfdsf@gmail.com	Networking	1/8/2020
7	Aishek Lamichaane	Pokhara	9846651361	aefadsfdsf@gmail.com	Networking	1/8/2020
8	Rajan Raj	Pokhara	9846546517	asdas@gmail.com	Networking	1/8/2020
9	Prakash Rana	Pokhara	9849849448	asaas@gmail.com	Computing	1/8/2020

Figure 7: Retrieving Student Details

If we click on the retrieve button the data table of all user entries are displayed in the data grid.

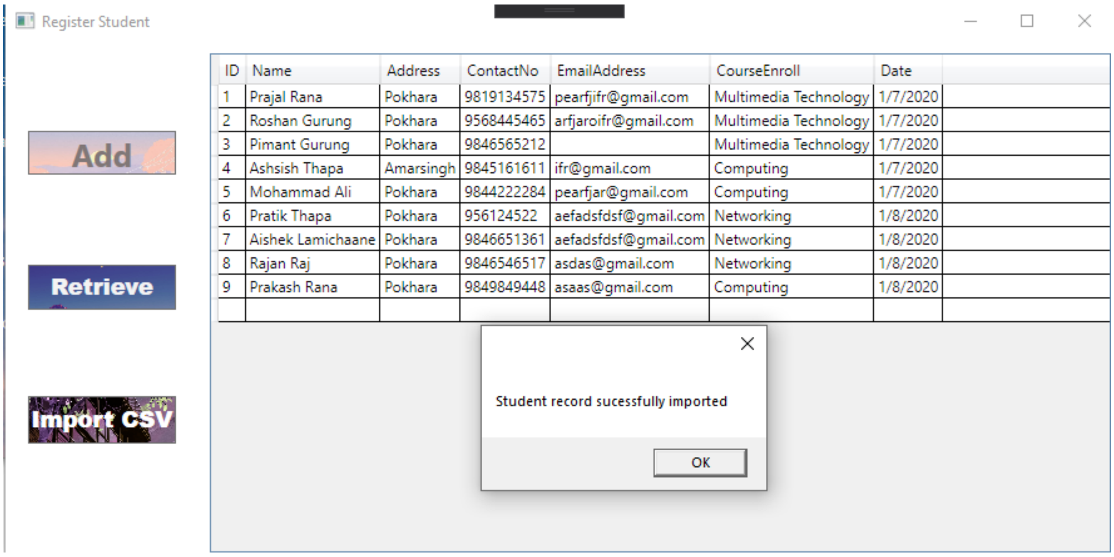


Name	Date modified	Type	Size
Backup	1/9/2020 4:43 PM	Text Document	6 KB
backupcsvcode2.0	1/9/2020 10:39 PM	Text Document	2 KB
citation	1/8/2020 7:42 PM	Text Document	1 KB
CommonReportViewer.aspx	9/26/2019 11:29 AM	ASP.NET Server Pa...	254 KB
importformCSV Codde	1/9/2020 4:37 PM	Text Document	4 KB
mojave	9/23/2019 4:50 PM	Text Document	1 KB
New file CSV	1/5/2020 8:27 AM	Microsoft Excel C...	1 KB
New Register	1/6/2020 7:27 AM	Microsoft Excel C...	1 KB
NewStudentDetails	1/6/2020 7:42 AM	Microsoft Excel C...	1 KB
progress repoet interim	12/6/2019 10:46 AM	Microsoft Word D...	356 KB
studentinfo	1/9/2020 10:54 PM	Microsoft Excel C...	1 KB
StudentManagementSystem2.0	1/9/2020 1:10 PM	WinRAR ZIP archive	30,348 KB
WPFToolkitBinariesAndSource	1/9/2020 12:14 PM	WinRAR ZIP archive	1,820 KB

File name: studentinfo

Figure 8 : Importing external CSV file

If we click on the “Import CSV” button a file dialog is opened. The user can then select any file with CSV extension to import in the system.

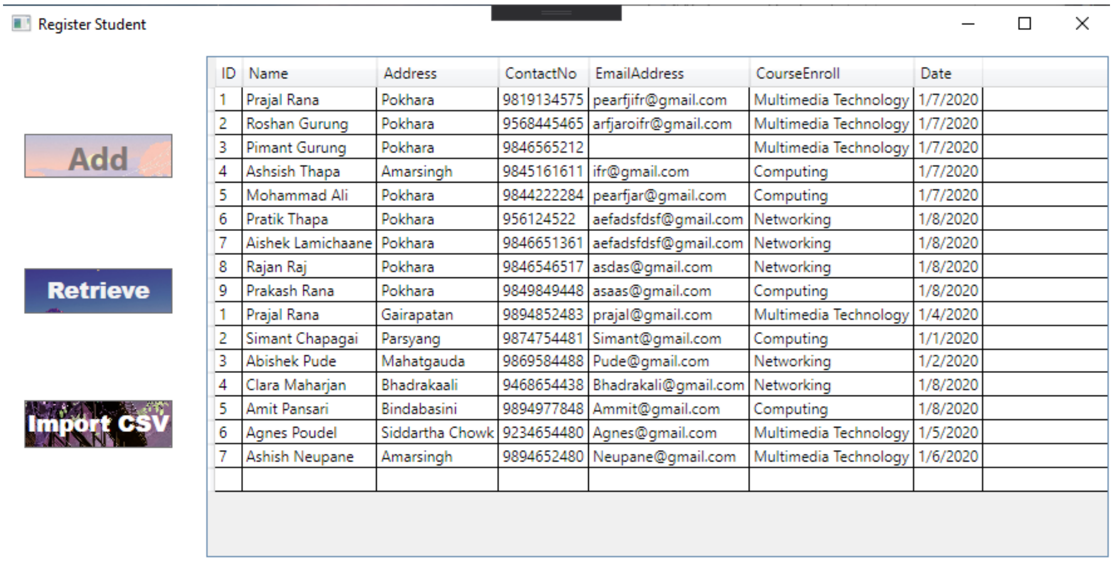


ID	Name	Address	ContactNo	EmailAddress	CourseEnroll	Date
1	Prajal Rana	Pokhara	9819134575	pearfjifr@gmail.com	Multimedia Technology	1/7/2020
2	Roshan Gurung	Pokhara	9568445465	arfjaroifr@gmail.com	Multimedia Technology	1/7/2020
3	Pimant Gurung	Pokhara	9846565212		Multimedia Technology	1/7/2020
4	Ashsish Thapa	Amarsingh	9845161611	ifr@gmail.com	Computing	1/7/2020
5	Mohammad Ali	Pokhara	9844222284	pearfjar@gmail.com	Computing	1/7/2020
6	Pratik Thapa	Pokhara	956124522	aefadsfdsf@gmail.com	Networking	1/8/2020
7	Aishek Lamichaane	Pokhara	9846651361	aefadsfdsf@gmail.com	Networking	1/8/2020
8	Rajan Raj	Pokhara	9846546517	asdas@gmail.com	Networking	1/8/2020
9	Prakash Rana	Pokhara	9849849448	asaas@gmail.com	Computing	1/8/2020

Student record sucessfully imported

Figure 9: CSV file imported

When a CSV file is selected and opened the system displays a message telling the file was imported successfully.



ID	Name	Address	ContactNo	EmailAddress	CourseEnroll	Date
1	Prajal Rana	Pokhara	9819134575	pearfjifr@gmail.com	Multimedia Technology	1/7/2020
2	Roshan Gurung	Pokhara	9568445465	arfjaroifr@gmail.com	Multimedia Technology	1/7/2020
3	Pimant Gurung	Pokhara	9846565212		Multimedia Technology	1/7/2020
4	Ashsish Thapa	Amarsingh	9845161611	ifr@gmail.com	Computing	1/7/2020
5	Mohammad Ali	Pokhara	9844222284	pearfjar@gmail.com	Computing	1/7/2020
6	Pratik Thapa	Pokhara	956124522	aefadsfdsf@gmail.com	Networking	1/8/2020
7	Aishek Lamichaane	Pokhara	9846651361	aefadsfdsf@gmail.com	Networking	1/8/2020
8	Rajan Raj	Pokhara	9846546517	asdas@gmail.com	Networking	1/8/2020
9	Prakash Rana	Pokhara	9849849448	asaas@gmail.com	Computing	1/8/2020
1	Prajal Rana	Gairapatan	9894852483	prajal@gmail.com	Multimedia Technology	1/4/2020
2	Simant Chapagai	Parsyang	9874754481	Simant@gmail.com	Computing	1/1/2020
3	Abishek Pude	Mahatgauda	9869584488	Pude@gmail.com	Networking	1/2/2020
4	Clara Maharjan	Bhadrakaali	9468654438	Bhadrakali@gmail.com	Networking	1/8/2020
5	Amit Pansari	Bindabasini	9894977848	Ammit@gmail.com	Computing	1/8/2020
6	Agnes Poudel	Siddartha Chowk	9234654480	Agnes@gmail.com	Multimedia Technology	1/5/2020
7	Ashish Neupane	Amarsingh	9894652480	Neupane@gmail.com	Multimedia Technology	1/6/2020

Figure 10: Student details after importing CSV file

After importing the CSV file the user can then click on retrieve button to display the CSV data in the table. The CSV file is generated in to an XML file and merged with previous report.

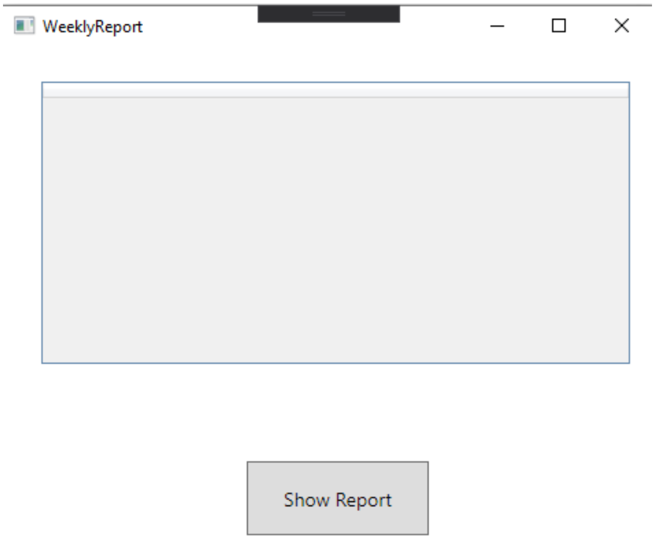
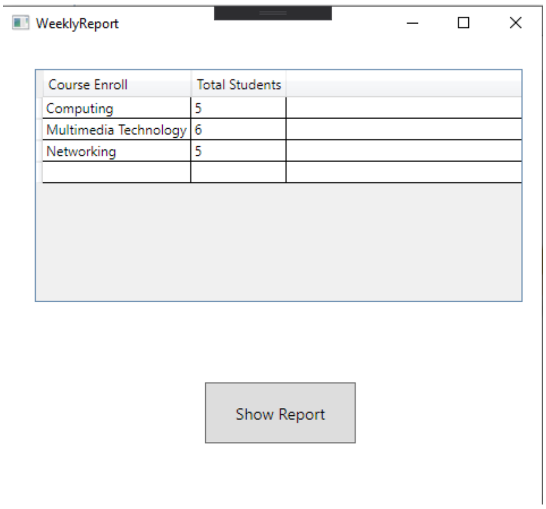


Figure 11: Weekly total report window

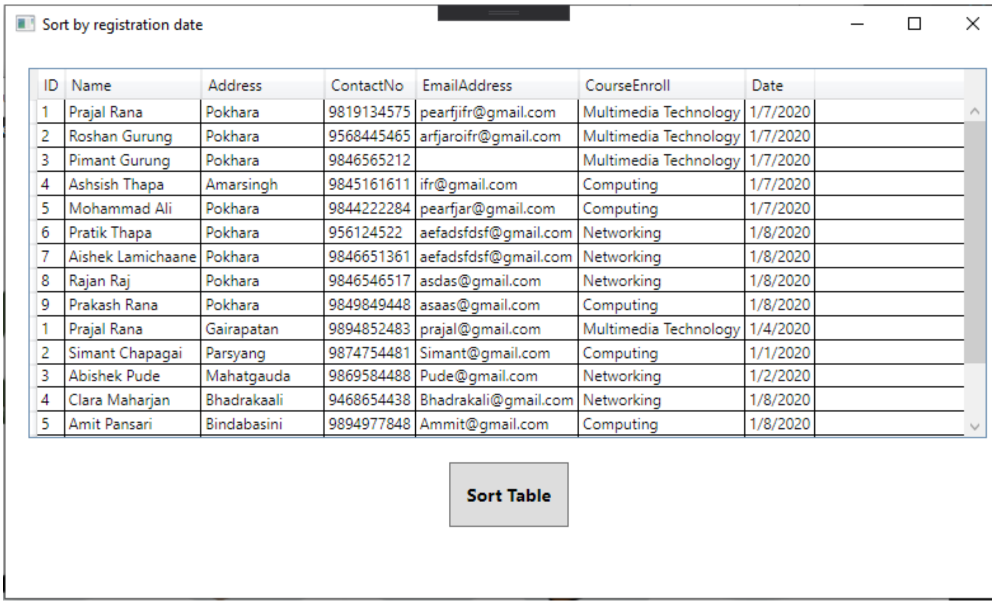
After closing the Register Student window user can select Weekly course report menu item in the Student Management System window to open Weekly Report window.



Course Enroll	Total Students
Computing	5
Multimedia Technology	6
Networking	5

Figure 12: Total student enrollment in each courses including data from CSV file

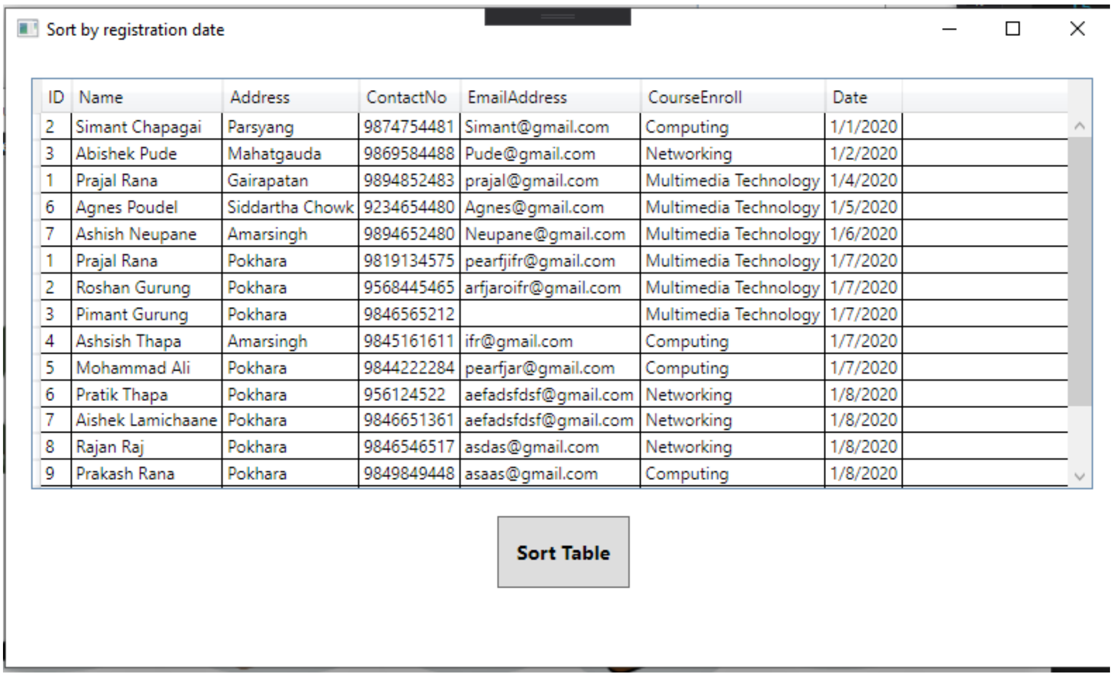
If you click on the “Show Report” button a data table is shown in the data grid. The data table includes columns Course Enroll and Total Students. The table shows the total number of students enrolled in each course.



ID	Name	Address	ContactNo	EmailAddress	CourseEnroll	Date
1	Prajal Rana	Pokhara	9819134575	pearfjifr@gmail.com	Multimedia Technology	1/7/2020
2	Roshan Gurung	Pokhara	9568445465	arfjaroifr@gmail.com	Multimedia Technology	1/7/2020
3	Pimant Gurung	Pokhara	9846565212		Multimedia Technology	1/7/2020
4	Ashsish Thapa	Amarsingh	9845161611	ifr@gmail.com	Computing	1/7/2020
5	Mohammad Ali	Pokhara	9844222284	pearfjar@gmail.com	Computing	1/7/2020
6	Pratik Thapa	Pokhara	956124522	aefadsfdsf@gmail.com	Networking	1/8/2020
7	Aishek Lamichaane	Pokhara	9846651361	aefadsfdsf@gmail.com	Networking	1/8/2020
8	Rajan Raj	Pokhara	9846546517	asdas@gmail.com	Networking	1/8/2020
9	Prakash Rana	Pokhara	9849849448	asaas@gmail.com	Computing	1/8/2020
1	Prajal Rana	Gairapatan	9894852483	prajal@gmail.com	Multimedia Technology	1/4/2020
2	Simant Chapagai	Parsyang	9874754481	Simant@gmail.com	Computing	1/1/2020
3	Abishek Pude	Mahatgauda	9869584488	Pude@gmail.com	Networking	1/2/2020
4	Clara Maharjan	Bhadrakaali	9468654438	Bhadrakali@gmail.com	Networking	1/8/2020
5	Amit Pansari	Bindabasini	9894977848	Ammit@gmail.com	Computing	1/8/2020

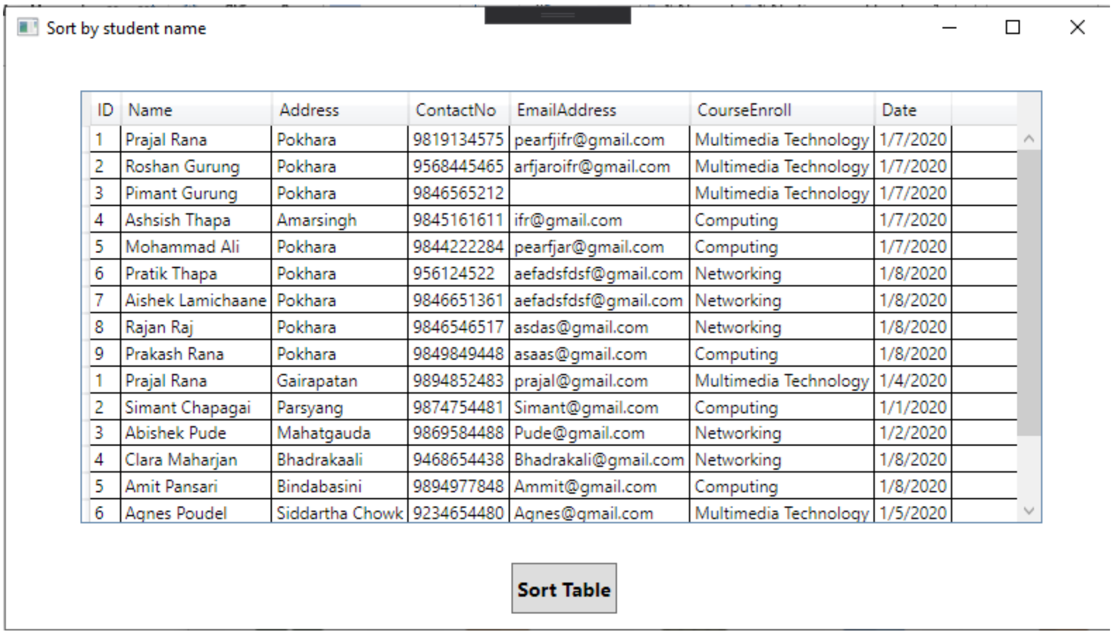
Figure 13: Sort by registration date window

User can select Sorting by Date menu item form the Student Management System window to open Sort by registration date window. The user is introduced with a final report in the data grid. When clicked on the “Sort Table” button the table is sorted according to ascending date format. The picture below illustrates the outcome.



ID	Name	Address	ContactNo	EmailAddress	CourseEnroll	Date
2	Simant Chapagai	Parsyang	9874754481	Simant@gmail.com	Computing	1/1/2020
3	Abishek Pude	Mahatgauda	9869584488	Pude@gmail.com	Networking	1/2/2020
1	Prajal Rana	Gairapatan	9894852483	prajal@gmail.com	Multimedia Technology	1/4/2020
6	Agnes Poudel	Siddartha Chowk	9234654480	Agnes@gmail.com	Multimedia Technology	1/5/2020
7	Ashish Neupane	Amarsingh	9894652480	Neupane@gmail.com	Multimedia Technology	1/6/2020
1	Prajal Rana	Pokhara	9819134575	pearfjifr@gmail.com	Multimedia Technology	1/7/2020
2	Roshan Gurung	Pokhara	9568445465	arfjaroifr@gmail.com	Multimedia Technology	1/7/2020
3	Pimant Gurung	Pokhara	9846565212		Multimedia Technology	1/7/2020
4	Ashsish Thapa	Amarsingh	9845161611	ifr@gmail.com	Computing	1/7/2020
5	Mohammad Ali	Pokhara	9844222284	pearfjar@gmail.com	Computing	1/7/2020
6	Pratik Thapa	Pokhara	956124522	aefadsfdsf@gmail.com	Networking	1/8/2020
7	Aishek Lamichaane	Pokhara	9846651361	aefadsfdsf@gmail.com	Networking	1/8/2020
8	Rajan Raj	Pokhara	9846546517	asdas@gmail.com	Networking	1/8/2020
9	Prakash Rana	Pokhara	9849849448	asaas@gmail.com	Computing	1/8/2020

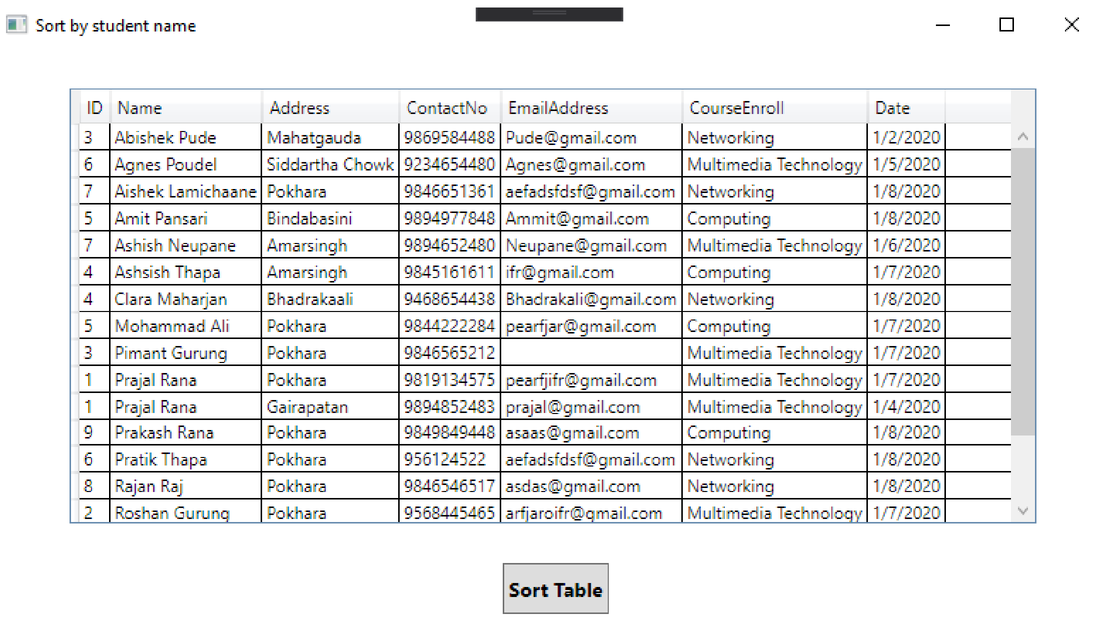
Figure 14: Data sorted with ascending registration date after clicking sort table button



ID	Name	Address	ContactNo	EmailAddress	CourseEnroll	Date
1	Prajal Rana	Pokhara	9819134575	pearfjifr@gmail.com	Multimedia Technology	1/7/2020
2	Roshan Gurung	Pokhara	9568445465	arfjaroifr@gmail.com	Multimedia Technology	1/7/2020
3	Pimant Gurung	Pokhara	9846565212		Multimedia Technology	1/7/2020
4	Ashsish Thapa	Amarsingh	9845161611	ifr@gmail.com	Computing	1/7/2020
5	Mohammad Ali	Pokhara	9844222284	pearfjar@gmail.com	Computing	1/7/2020
6	Pratik Thapa	Pokhara	956124522	aefadsfdsf@gmail.com	Networking	1/8/2020
7	Aishek Lamichaane	Pokhara	9846651361	aefadsfdsf@gmail.com	Networking	1/8/2020
8	Rajan Raj	Pokhara	9846546517	asdas@gmail.com	Networking	1/8/2020
9	Prakash Rana	Pokhara	9849849448	asaas@gmail.com	Computing	1/8/2020
1	Prajal Rana	Gairapatan	9894852483	prajal@gmail.com	Multimedia Technology	1/4/2020
2	Simant Chapagai	Parsyang	9874754481	Simant@gmail.com	Computing	1/1/2020
3	Abishek Pude	Mahatgauda	9869584488	Pude@gmail.com	Networking	1/2/2020
4	Clara Maharjan	Bhadrakaali	9468654438	Bhadrakali@gmail.com	Networking	1/8/2020
5	Amit Pansari	Bindabasini	9894977848	Ammit@gmail.com	Computing	1/8/2020
6	Agnes Poudel	Siddartha Chowk	9234654480	Agnes@gmail.com	Multimedia Technology	1/5/2020

Figure 15: Sort by name window

User can select Sorting by Name menu item form the Student Management System window to open Sort by student name window. The user is introduced with a final report in the data grid. When clicked on the “Sort Table” button the table is sorted according to ascending alphabetic name format. The picture below illustrates the outcome.



ID	Name	Address	ContactNo	EmailAddress	CourseEnroll	Date
3	Abishek Pude	Mahatgauda	9869584488	Pude@gmail.com	Networking	1/2/2020
6	Agnes Poudel	Siddartha Chowk	9234654480	Agnes@gmail.com	Multimedia Technology	1/5/2020
7	Aishek Lamichaane	Pokhara	9846651361	aefadsfdsf@gmail.com	Networking	1/8/2020
5	Amit Pansari	Bindabasini	9894977848	Ammit@gmail.com	Computing	1/8/2020
7	Ashish Neupane	Amarsingh	9894652480	Neupane@gmail.com	Multimedia Technology	1/6/2020
4	Ashsish Thapa	Amarsingh	9845161611	ifr@gmail.com	Computing	1/7/2020
4	Clara Maharjan	Bhadrakaali	9468654438	Bhadrakali@gmail.com	Networking	1/8/2020
5	Mohammad Ali	Pokhara	9844222284	pearfjar@gmail.com	Computing	1/7/2020
3	Pimant Gurung	Pokhara	9846565212		Multimedia Technology	1/7/2020
1	Prajal Rana	Pokhara	9819134575	pearfjifr@gmail.com	Multimedia Technology	1/7/2020
1	Prajal Rana	Gairapatan	9894852483	prajal@gmail.com	Multimedia Technology	1/4/2020
9	Prakash Rana	Pokhara	9849849448	asaas@gmail.com	Computing	1/8/2020
6	Pratik Thapa	Pokhara	956124522	aefadsfdsf@gmail.com	Networking	1/8/2020
8	Rajan Raj	Pokhara	9846546517	asdas@gmail.com	Networking	1/8/2020
2	Roshan Gurung	Pokhara	9568445465	arfjaroifr@gmail.com	Multimedia Technology	1/7/2020

Figure 16: Data sorted with ascending names after clicking sort table button

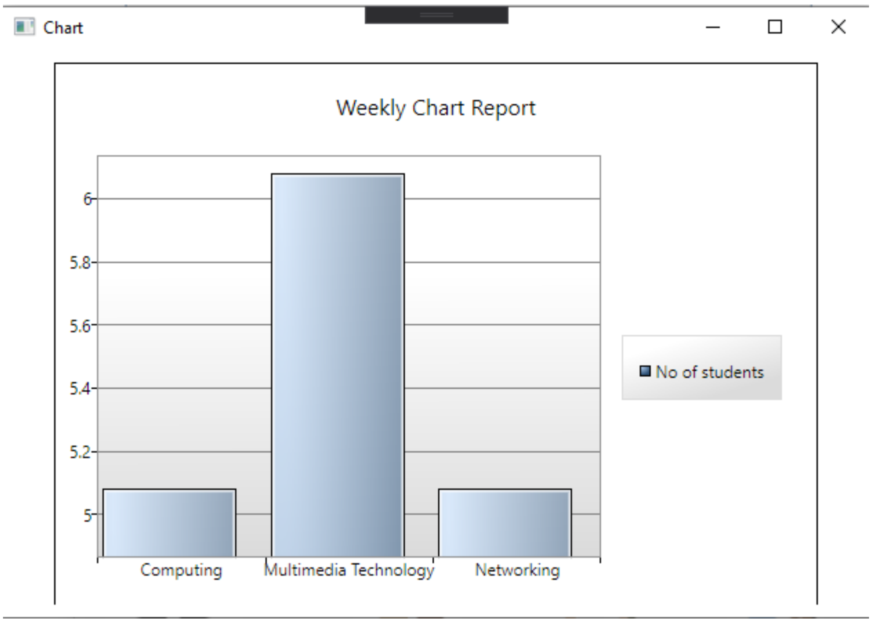
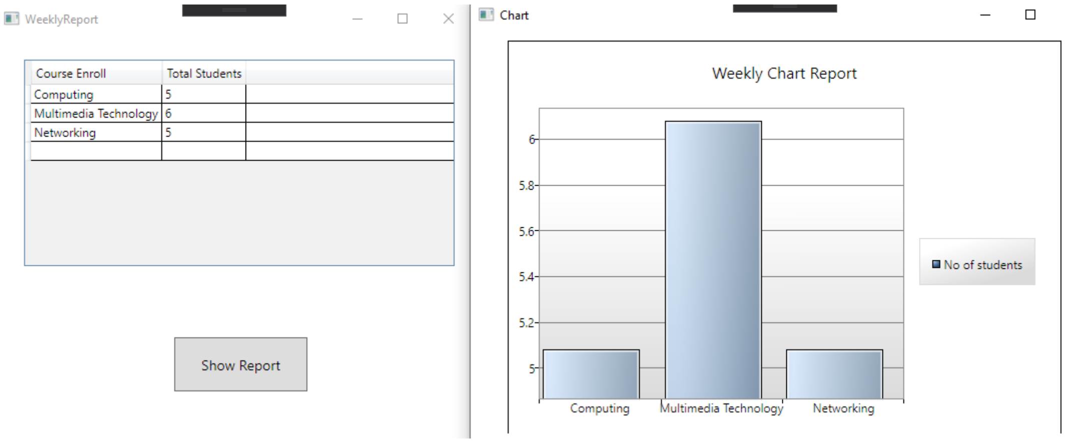


Figure 17: Chart window

User can select the Chart menu item form the Student Management System window to open the Chart window. The chart illustrates the total number of students enrolled in each course in a diagram.



Course Enroll	Total Students	
Computing	5	
Multimedia Technology	6	
Networking	5	

Figure 18: Chart developed based on weekly total student enrollment details

3. Journal Articles

1. This article discusses on student information management system in the department of student affairs. The system will solve problems caused by traditional system of student information management. Transforming the department from manually to a digital based computer system leads to provide accuracy, efficiency and security. The benefits of using a computer based student management system are making all information relating to students in one place, speed in completing the registration process, speed in completing the task of the department, give each student an id number which facilitates the process of administration of information, supports the administration of college the ability to speed decision-making, the system provides protection to the data by preventing unauthorized users from accessing to the data stored in the system, if the department suffers from a lack of staff, the adoption of this system does not require a large number of employees (Hassan, 2018).

2. The design and implementation of student information system is to replace traditional paper-based records. School and college administration can access all aspects of student’s academic progress, attendance-details and various activities of students through a secure platform. All data are validated in the system before any alteration occurs. The system ensures highest possible level of security. The system features a logging system to track all Users-access and ensure conformity to data access guidelines and is expected to increase the efficiency of the school or college record management thereby lowering the work hours needed to access and deliver student records to the user. The system thus increases the efficiency of the school or college record management, decreasing time required to access and deliver student records, to make the system more secure (Dipin Budhrani, 2018).

3. The school administration computer application development, started in the late 1970s. The use of information technology in academic institutions was used mainly to improve the efficiency of school offices, such as store student and employee data. The use of school management system was developed to fill the gap, due to its feature to generate data and its efficiency and effectiveness to save time and develop solutions for sophisticated problems (Forrester, 2019).

4. System Architecture

4.1 Architecture Diagram

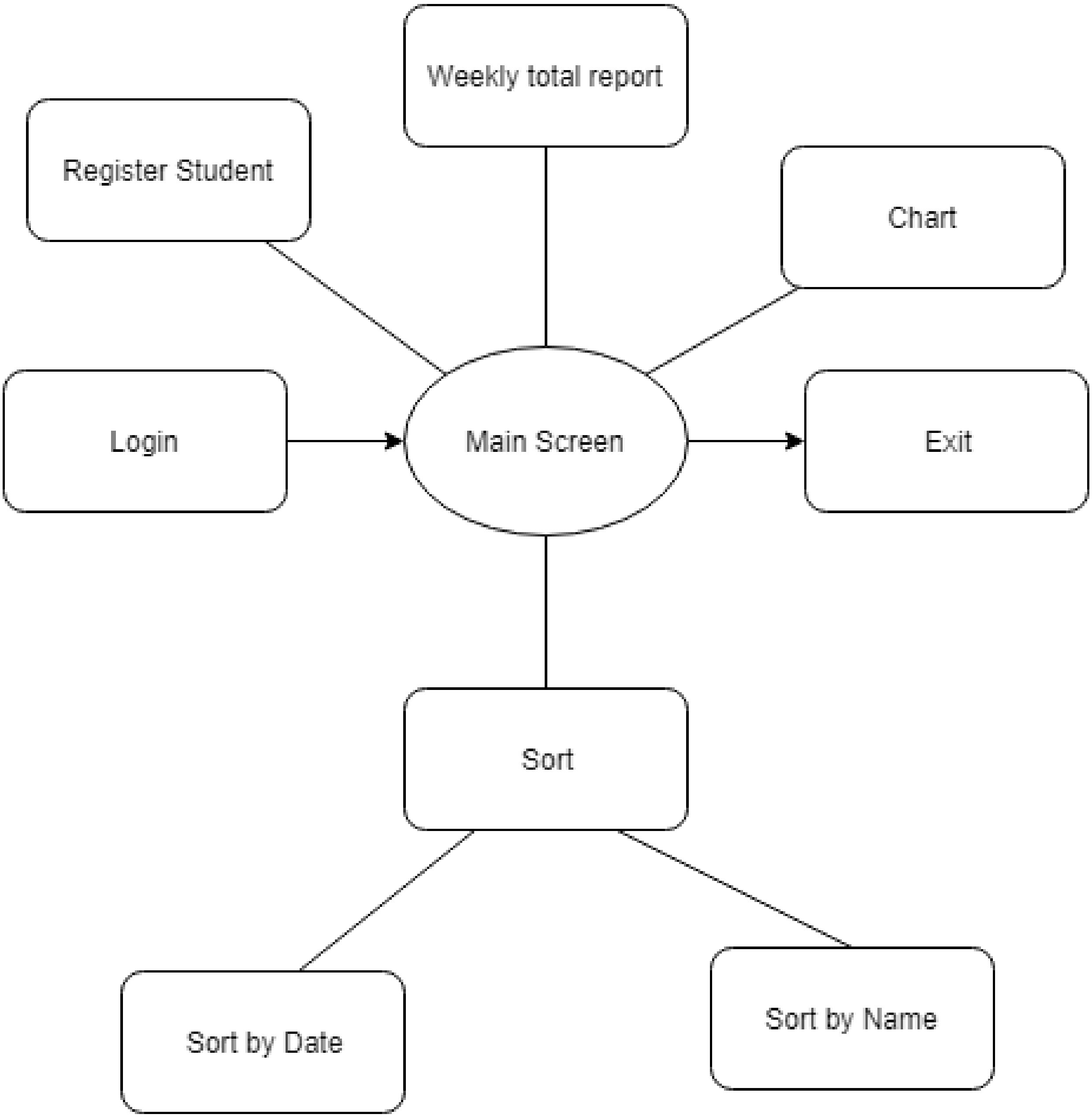


Figure 19: Architecture Diagram

4.2 Class Diagram

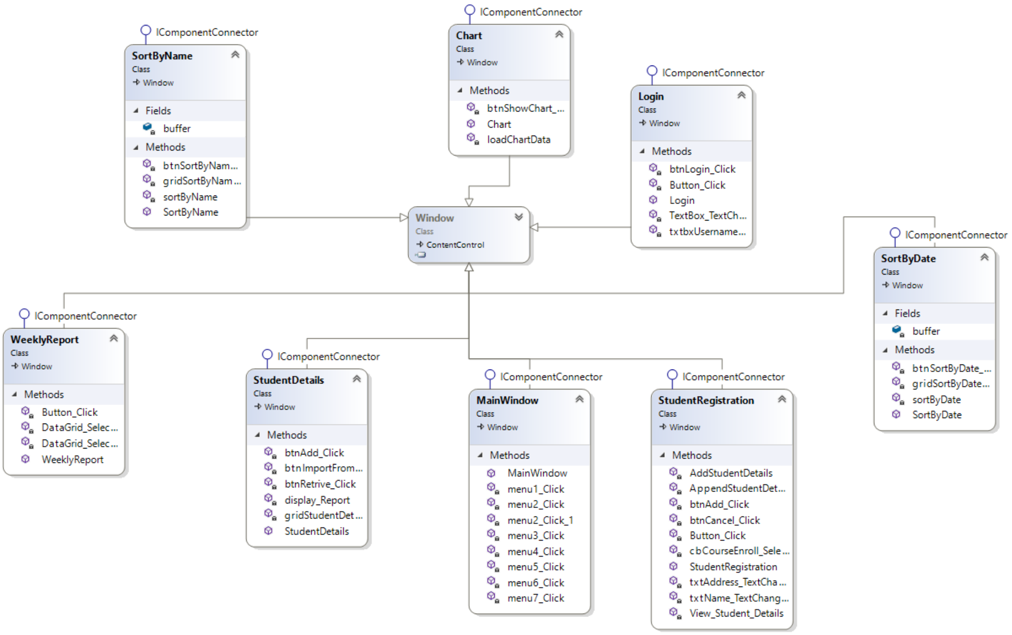


Figure 20: Class Diagram

Individual Diagrams

1. Login

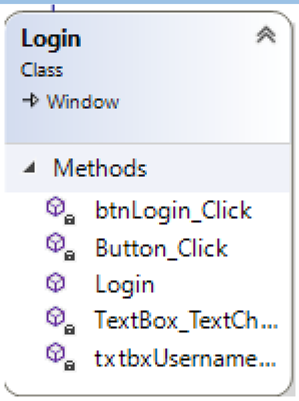
Methods	Description	Diagram
btnLogin_Click	Checks if the entered username and password before entering the program.	 Login Class Window Methods btnLogin_Click Button_Click Login TextBox_TextCh... txtbxUsername...
Button_Click	Terminates the program.	
Login	Initialize component.	

Table 1: Login description

2. MainWindow

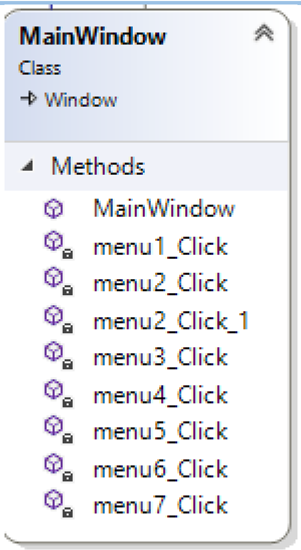
Methods	Description	Diagram
menu1_click	Show Register Student window.	 MainWindow Class Window Methods MainWindow menu1_Click menu2_Click menu2_Click_1 menu3_Click menu4_Click menu5_Click menu6_Click menu7_Click
Menu2_click	Show WeeklyReport window.	
Menu4_click	Show menu item for Sort by registration date and Sort by student name windows.	
Menu5_click	Show Sort by registration date window.	
Menu6_click	Show Sot by student name window.	
Menu7_click	Show Chart window.	

Table 2: MainWIndow description

3. StudentDetails

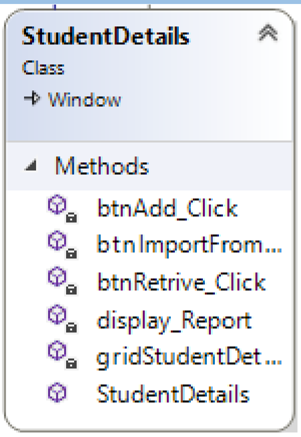
Methods	Description	Diagram
btnAdd_Click	Show Add student window	 StudentDetails Class Window Methods btnAdd_Click btnImportFrom... btnRetrive_Click display_Report gridStudentDet... StudentDetails
Display_Report	Reads xml file, creates new data set, create new table, assigns values to table, and displays data in data grid.	
btnImportFromCSV_Click	Opens files, reads CSV file, generates CSV file to xml and merges with previous XML file.	
btnRetrive_Click	When clicked on this button Display_Report method is called.	

Table 3: StudentDetails description

4. StudentRegistration

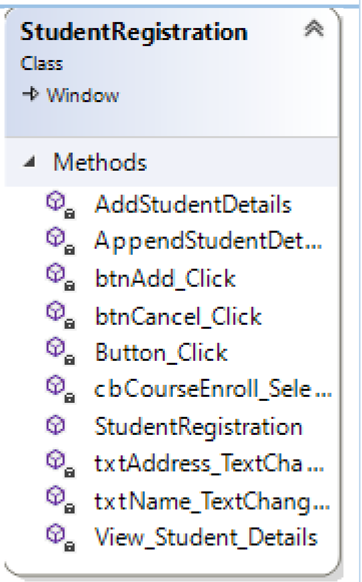
Methods	Description	Diagram
AddStudentDetails	Adds DataSet values to DataTable.	 StudentRegistration Class → Window ◢ Methods AddStudentDetails AppendStudentDet... btnAdd_Click btnCancel_Click Button_Click cbCourseEnroll_Sele... StudentRegistration txtAddress_TextCha... txtName_TextChang... View_Student_Details
AppendStudentDetails	Adds new value to the table.	
View_Student_Details	Reads XML file, assigns value to DataSet	
btnAdd_Click	Adds new student data to the table.	

Table 4: StudentRegistration description

5. WeeklyReport

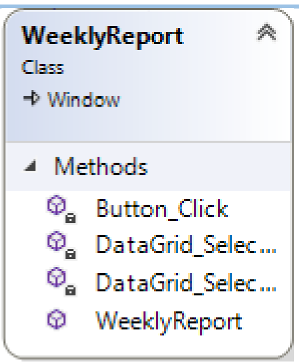
Methods	Description	Diagram
Button_Click	Reads student report xml file, create new dataset, creates new table, counts the total number of students in each course, assigns those based on table rows.	 WeeklyReport Class → Window ◢ Methods Button_Click DataGrid_Selec... DataGrid_Selec... WeeklyReport

Table 5: WeeklyReport description

6. SortByDate

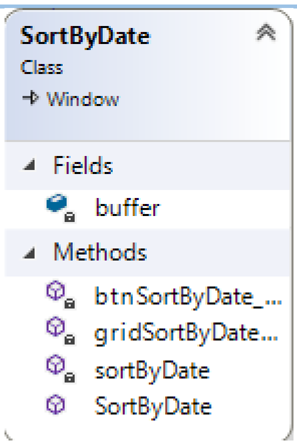
Methods	Description	Diagram
sortByDate	Reads XML file, creates DataSet and DataTable, loops the file to count the number of rows, assigns values to the rows, displays data in data grid.	 SortByDate Class → Window Fields buffer Methods btnSortByDate_... gridSortByDate... sortByDate SortByDate
btnSortByDate_Click	Sorts the table with ascending date values.	
SoryByDate	Calls sortByDate method	

Table 6: SortByDate description

7. SortByName

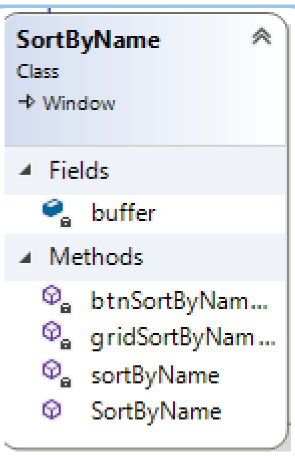
Method	Description	Diagram
sortByName	Reads XML file, creates DataSet and DataTable, loops the file to count the number of rows, assigns values to the rows, displays data in data grid.	 SortByName Class → Window Fields buffer Methods btnSortByNam... gridSortByNam... sortByName SortByName
btnSortByName_Click	Sorts the table with ascending alphabetic values.	
SortByName	Calls sortByName method	

Table 7: SortByName description

8. Chart

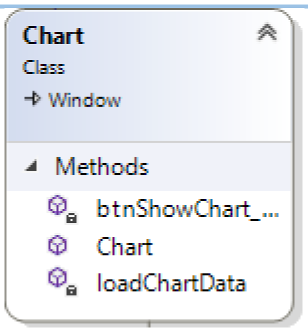
Method	Description	Diagram
loadChartData	Reads student report xml file, counts the total number of students in each course, assigns those numbers as key values	 Chart Class → Window Methods btnShowChart_... Chart loadChartData
Chart	Calls loadChartData	

Table 8: Chart Description

4.3 Flowchart of Report

4.3.1 Student Registration

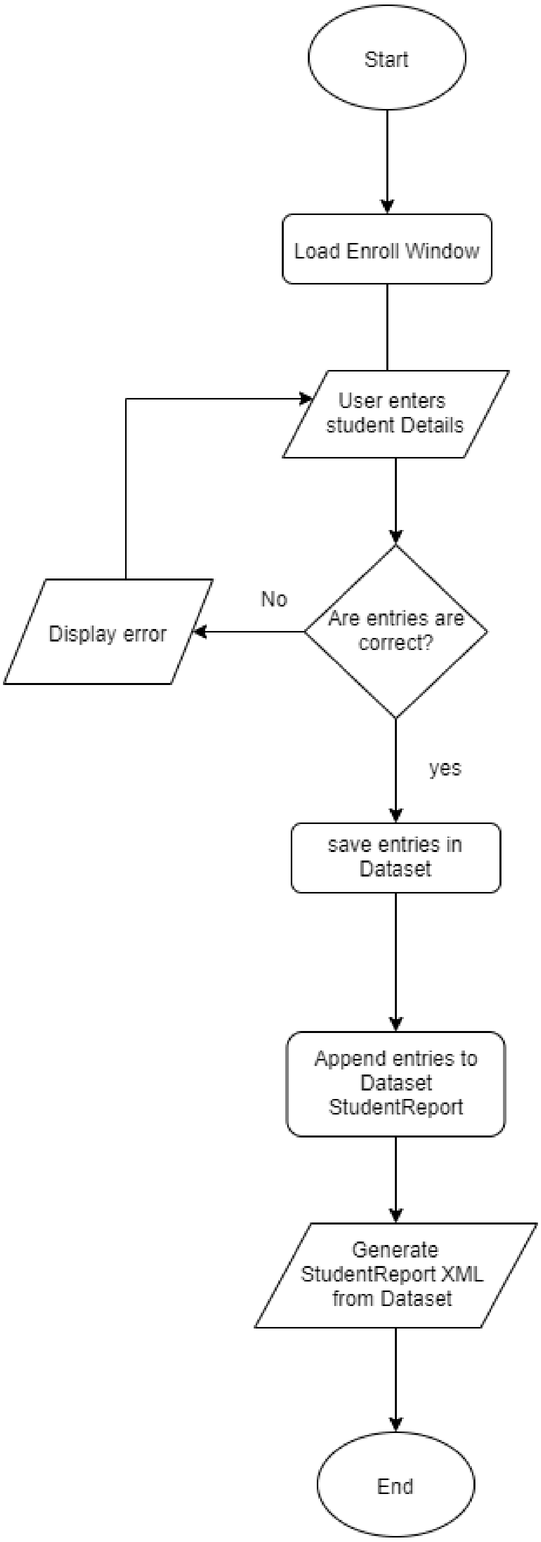

Start
Load Enroll Window
User enters student Details
Are entries are correct?
No
Display error
yes
save entries in Dataset
Append entries to Dataset StudentReport
Generate StudentReport XML from Dataset
End

Figure 21: Student Registration flowchart

4.3.2 Importing CSV file

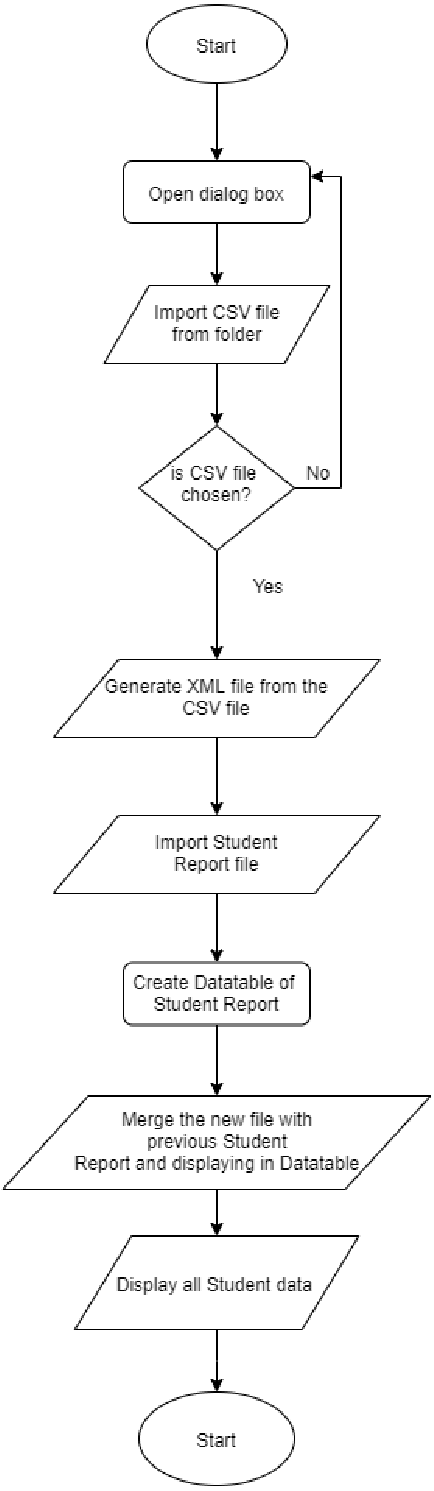

Start
Open dialog box
Import CSV file from folder
is CSV file chosen?
No
Yes
Generate XML file from the CSV file
Import Student Report file
Create Datatable of Student Report
Merge the new file with previous Student Report and displaying in Datatable
Display all Student data
Start

Figure 22: Import CSV flowchart

5. Sorting Algorithm

Bubble sort is a simple sorting algorithm used for sorting data. In this sorting algorithm each pair of adjacent elements is compared and the elements are swapped if they are not in correct order. This algorithm is used in large data sets as its average and worst case complexity are of $O(n^2)$ where n is the number of items in an array (Tutorialspoint, 2020).

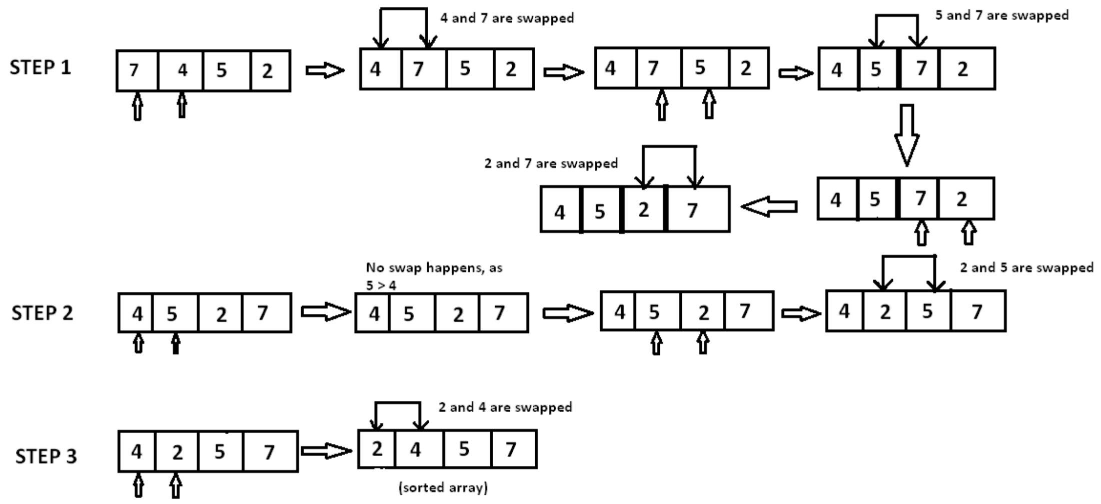


Figure 23: Bubble Sort

Step 1: In first step 7 is compared with 4. Since 4<7, 7 is shifted ahead of 4. Since all the other elements are of a lesser value than 7, 7 is shifted to the end of the array.

Step 2: 4 is then compared with 5. Since 5>4 and both 4 and 5 are in ascending order, these elements are not swapped. However, when 5 is compared with 2, 5>2 and these elements are in descending order. Therefore, the position of 5 and 2 are swapped.

Step 3: The element 4 is compared 4 is compared with 2. Since 2<4 and the elements are in descending order, 4 and 2 are swapped.

The sorted array is A = {2, 4, 5, 7} (Hackerearth, 2020).

6. Reflection

The proposed system was developed using Visual Studio 2019 as a WPF app (.Net framework) project with C#. The system is made specific for modern day management process with intuitive user interface. The workflow within the system resembles the workflow of traditional student management system but in a digital environment.

The system allows authorized personnel to add and import student data. The system provides ID automatically to a new entry. The name, address, contact number, email address, course enrolled and registration date of student are recorded and saved in the system. The end user can retrieve data and sort the data table by name or registration date. Furthermore, the end user can view the chart based on weekly total report.

C# and Visual studio were new to me this year, so getting started with the development process was a bit difficult. But, I progressed with the help of my module leader and journals. My module leader helped us with the basics of WPF form and visual studio workflow. The process of displaying the chart was challenging but not difficult. Import CSV file was also new to me. Overall I was able to develop a working student management system.

7. Conclusion

The coursework was finally completed with all needed requirements. The system was developed with a window based interface in Visual Studio 2019 using C# programming language. Only user form user administration can access the system as the system is protected with a login screen. The end users can then manipulate the system with the provided functionalities. I found the process of managing the record of students digitally to be much easier and time saving. I hope we get to see more digital mediums for recording and managing not only student details but other data from different sources.

References

Dipin Budhrani, V. M. (2018). Student Information Management System. 8-10.

Forrester, V. V. (2019). School Management information systems: Challenges to educational decision-making in the big data era. *International Journal on Integrating Technology in Education (IJITE)*, 1-11.

Hackerearth. (2020, January 2). Retrieved from hackerearth.com: https://www.hackerearth.com/practice/algorithms/sorting/bubble-sort/tutorial/

Hassan, I. A. (2018). Design and Implement a NovelStudent Information ManagementSystem. *International Journal of Computer Science and Mobile computing*, 20-31.

Tutorialspoint. (2020, January 1). Retrieved from Tutorialspoint.com: https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm

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;

namespace StudentManagementSystem2._0 {
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Window {
        public Login() {
            InitializeComponent();
        }

        private void btnLogin_Click(object sender, RoutedEventArgs e) {
            if (txtbxUsername.Text != "admin") // checks if the username isn't 
matching the required username
            {
                MessageBox.Show("Username is incorrect!", "Alert");
                txtbxUsername.Clear();
            }
            else if (txtbxPassword.Password != "admin") // checks if the 
password isn't matching the required password
            {
                MessageBox.Show("Password is incorrect", "Alert");
                txtbxPassword.Clear();
            }
            else // if username and password are matched MainWindow is opened
            {

                MainWindow hm = new MainWindow();
                hm.Show();
                this.Close(); // after loggin in it temrinates the login window

            }

        }


        private void txtbxUsername_TextChanged_1(object sender, 
TextChangedEventArgs e) {

        }
        private void Button_Click(object sender, RoutedEventArgs e) {
            System.Windows.Application.Current.Shutdown();  // on hitting 
cancel button the system is terminated
        }

        private void TextBox_TextChanged(object sender, TextChangedEventArgs e) 
{

        }
    }
}
```

MainWindow.xaml.cs

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

namespace StudentManagementSystem2._0 {
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window {
        public MainWindow() {
            InitializeComponent();
        }


        private void menu2_Click(object sender, RoutedEventArgs e) {
            WeeklyReport weeklydtls = new WeeklyReport();
            weeklydtls.Show();
        }

        private void menu1_Click(object sender, RoutedEventArgs e) {
            StudentDetails stddetails = new StudentDetails();
            stddetails.Show();
        }

        private void menu2_Click_1(object sender, RoutedEventArgs e) {

        }

        private void menu3_Click(object sender, RoutedEventArgs e) {

        }

        private void menu5_Click(object sender, RoutedEventArgs e) {
            SortByDate srtbd = new SortByDate();
            srtbd.Show();
        }

        private void menu4_Click(object sender, RoutedEventArgs e) {

        }

        private void menu6_Click(object sender, RoutedEventArgs e) {
            SortByName srtbn = new SortByName();
            srtbn.Show();
        }
        private void menu7_Click(object sender, RoutedEventArgs e) {
            Chart chartt = new Chart();
            chartt.Show();
        }
    }
}
```

StudentDetails.xaml.cs

```
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;
using DataHandler;
using Microsoft.Win32;

namespace StudentManagementSystem2._0 {
    /// <summary>
    /// Interaction logic for StudentDetails.xaml
    /// </summary>
    public partial class StudentDetails : Window {

        public StudentDetails() {
            InitializeComponent();
            //display_Report();
        }



        private void btnAdd_Click(object sender, RoutedEventArgs e) {
            StudentRegistration stdRegistration = new StudentRegistration();
            stdRegistration.Show();
        }

        private void gridStudentDetails_SelectionChanged(object sender,
SelectionChangedEventArgs e) {

        }
        private void display_Report() {
            string sampleXmlFile = @"E:\College\3rd Year\Application
Development\StudentReport.xml"; // assiging path to sting variable
sampleXmlFile
            DataSet dataset = new DataSet(); // declaring new DataSet dataset
            dataset.ReadXml(sampleXmlFile); // sampleXMlFile is read as an XML
file.

            DataTable 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("ContactNo", typeof(String));
            buffer.Columns.Add("EmailAddress", typeof(String));
            buffer.Columns.Add("CourseEnroll", typeof(String));
            buffer.Columns.Add("Date", typeof(String));
```

```
      for(int i=0;i< dataset.Tables[0].Rows.Count; i++)
      {
        string s = dataset.Tables[0].Rows[i][6].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(),
          dataset.Tables[0].Rows[i][5].ToString(),
          dtime.ToShortDateString());

      }

      DataView dataView = new DataView(buffer); // setting the itemsource
to table
      gridStudentDetails.ItemsSource = dataView;  // viewing table to
data grid
    }


    private void btnRetrive_Click(object sender, RoutedEventArgs e) {
      display_Report();
    }

    private void btnImportFromCSV_Click(object sender, RoutedEventArgs e) {


      try
      {
        var dataSet = new DataSet();  //declaring new DataSet dataSet
        dataSet.ReadXml(@"E:\College\3rd Year\Application
Development\StudentReport.xml"); // xml file is read and set as dataSet
        Microsoft.Win32.OpenFileDialog dialog = new
Microsoft.Win32.OpenFileDialog(); // for opening folder
        if (dialog.ShowDialog() == true)  // if folder is accesed
        {
          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["StudentReport"].NewRow();

              newRow["ID"] = values[0];
              newRow["Name"] = values[1];
              newRow["Address"] = values[2];
              newRow["ContactNo"] = values[3];
              newRow["EmailAddress"] = values[4];
              newRow["CourseEnroll"] = values[5];
              newRow["RegistrationDate"] = values[6];
              dataSet.Tables["StudentReport"].Rows.Add(newRow);

              dataSet.WriteXml(@"E:\College\3rd Year\Application
Development\StudentReport.xml"); // data is appeded to the xml file
```

```
                        }
                    }
                    MessageBox.Show("Student record sucessfully imported");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

    }
}
```

StudentRegistration.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;
using DataHandler;
using System.IO;

namespace StudentManagementSystem2._0 {
    /// <summary>
    /// Interaction logic for StudentRegistration.xaml
    /// </summary>
    public partial class StudentRegistration : Window {
        public StudentRegistration() {
            InitializeComponent();
        }
        private void AddStudentDetails(DataSet dataset) {

                var dt_student = dataset.Tables["Student"].NewRow();
                dt_student["Name"] = txtName.Text;
                dt_student["Address"] = txtAddress.Text;
                dt_student["ContactNo"] = txtContactNo.Text;
                dt_student["EmailAddress"] = txtEmailAddress.Text;
                dt_student["CourseEnroll"] = cbCourseEnroll.Text;
                dt_student["RegistrationDate"] =
dpRegistrationDate.SelectedDate.ToString();
                //MessageBox.Show("Date Added" +
dpRegistrationDate.SelectedDate.ToString());
                dataset.Tables["Student"].Rows.Add(dt_student);
        }
        private void AppendStudentDetails(DataSet dataset) {

            if (File.Exists(@"E:\College\3rd Year\Application
Development\StudentReport.xml"))
            {
                dataset.Tables["StudentReport"].ReadXml(@"E:\College\3rd
Year\Application Development\StudentReport.xml");
                var dt_student = dataset.Tables["StudentReport"].NewRow();
                dt_student["Name"] = txtName.Text;
                dt_student["Address"] = txtAddress.Text;
                dt_student["ContactNo"] = txtContactNo.Text;
                dt_student["EmailAddress"] = txtEmailAddress.Text;
                dt_student["CourseEnroll"] = cbCourseEnroll.Text;
                dt_student["RegistrationDate"] =
dpRegistrationDate.SelectedDate;
                dataset.Tables["StudentReport"].Rows.Add(dt_student);
                dataset.Tables["StudentReport"].WriteXml(@"E:\College\3rd
Year\Application Development\StudentReport.xml");
            }
            else
            {
                dataset.Tables["StudentReport"].WriteXml(@"E:\College\3rd
Year\Application Development\StudentReport.xml");
                AppendStudentDetails(dataset);
            }

        }

        private void View_Student_Details() {
            if (File.Exists(@"E:\College\3rd Year\Application
Development\StudentReport.xml"))
            {
                var dataset = new DataSet();
                dataset.ReadXml(@"E:\College\3rd Year\Application
Development\StudentReport.xml");
            }
            else
            {
                MessageBox.Show("Sorry, there's data. Please fill up the form
to view data.");
            }

        }

        private void btnAdd_Click(object sender, RoutedEventArgs e) {
            var handler = new Handler();
            var dataset = handler.CreateDataSet();
            AddStudentDetails(dataset);
            AppendStudentDetails(dataset);

            dataset.Tables["Student"].WriteXml(@"E:\College\3rd
Year\Application Development\" + txtName.Text + "Data.xml");
            //Res_no_write(txtResNo.Text);
            //txtResNo.Text = Res_no_read();

            MessageBox.Show("Sudent Details saved successfully!");
            txtName.Text = "";
            txtAddress.Text = "";
            txtContactNo.Text = "";
            txtEmailAddress.Text = "";
            //cbCourseEnroll.SelectedIndex
            //dpRegistrationDate.SelectedDate
            StudentDetails stddetails = new StudentDetails();
            stddetails.Show();

        }

        private void btnCancel_Click(object sender, RoutedEventArgs e) {
            this.Close();
        }

        private void cbCourseEnroll_SelectionChanged(object sender,
SelectionChangedEventArgs e) {

        }
        private void Button_Click(object sender, RoutedEventArgs e) {

        }

        private void txtName_TextChanged(object sender, TextChangedEventArgs e)
{

        }

        private void txtAddress_TextChanged(object sender, TextChangedEventArgs
e) {

        }
    }
}
```

SortByDate.xaml.cs

```
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 StudentManagementSystem2._0 {
    public partial class SortByDate : Window {
        DataTable buffer;
        public SortByDate() {
            InitializeComponent();
            sortByDate();
        }

        private void gridSortByDate_SelectionChanged(object sender,
SelectionChangedEventArgs e) {

        }
        private void sortByDate() {
            string sampleXmlFile = @"E:\College\3rd Year\Application
Development\StudentReport.xml"; //declaring sampleXmlFile to xml file
destination
            DataSet dataset = new DataSet(); // createing new data set
            dataset.ReadXml(sampleXmlFile); // Reading the XML File

            buffer = new DataTable("dt"); //creating data table dt and
assigning to buffer
            buffer.Columns.Add("ID", typeof(String));                   //
Making new column ID with data type String
            buffer.Columns.Add("Name", typeof(String));
            buffer.Columns.Add("Address", typeof(String));
            buffer.Columns.Add("ContactNo", typeof(String));
            buffer.Columns.Add("EmailAddress", typeof(String));
            buffer.Columns.Add("CourseEnroll", typeof(String));
            buffer.Columns.Add("Date", typeof(String));

            for (int i = 0; i < dataset.Tables[0].Rows.Count; i++) // Changing
GMt format to local time zone
            {
                string s = dataset.Tables[0].Rows[i][6].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(),
                    dataset.Tables[0].Rows[i][5].ToString(),
                    dtime.ToShortDateString());

            }
            DataView dataView = new DataView(buffer); // setting the itemsource
to table
            gridSortByDate.ItemsSource = dataView;
        }
        // setting the itemsource to table
        // code responsible sorting in ascending order, In Date ASE, DATE
should match your variable from handler class
        // Displaying data
        private void btnSortByDate_Click(object sender, RoutedEventArgs e) {
            DataView dataView = new DataView(buffer);
            dataView.Sort = "Date ASC";
            gridSortByDate.ItemsSource = dataView;
        }
    }
}
```

SortByName.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 StudentManagementSystem2._0 {
    /// <summary>
    /// Interaction logic for SortByName.xaml
    /// </summary>
    public partial class SortByName : Window {
        DataTable buffer;
        public SortByName() {
            InitializeComponent();
            sortByName();
        }

        private void gridSortByName_SelectionChanged(object sender,
SelectionChangedEventArgs e) {

        }
        private void sortByName() {
            string sampleXmlFile = @"E:\College\3rd Year\Application
Development\StudentReport.xml";
            DataSet dataset = new DataSet();
            dataset.ReadXml(sampleXmlFile);

            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("ContactNo", typeof(String));
            buffer.Columns.Add("EmailAddress", typeof(String));
            buffer.Columns.Add("CourseEnroll", typeof(String));
            buffer.Columns.Add("Date", typeof(String));

            for (int i = 0; i < dataset.Tables[0].Rows.Count; i++)
            {
                string s = dataset.Tables[0].Rows[i][6].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(),
                    dataset.Tables[0].Rows[i][5].ToString(),
                    dtime.ToShortDateString());
            }

            DataView dataView = new DataView(buffer); // setting the itemsource 
to table
            gridSortByName .ItemsSource = dataView;
        }

        private void btnSortByName_Click(object sender, RoutedEventArgs e) {
            DataView dataView = new DataView(buffer); // setting the itemsource 
to table
            dataView.Sort = "Name ASC";
            gridSortByName.ItemsSource = dataView;

        }
    }
}
```

WeeklyReport.xaml.cs

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

namespace StudentManagementSystem2._0 {
    /// <summary>
    /// Interaction logic for WeeklyReport.xaml
    /// </summary>
    public partial class WeeklyReport : Window {

        public WeeklyReport() {
            InitializeComponent();
        }

        private void DataGrid_SelectionChanged(object sender,
SelectionChangedEventArgs e) {

        }

        private void DataGrid_SelectionChanged_1(object sender,
SelectionChangedEventArgs e) {

        }

        private void Button_Click(object sender, RoutedEventArgs e) {
            var dataset = new DataSet(); // declaring new data set
            dataset.ReadXml(@"E:\College\3rd Year\Application
Development\StudentReport.xml");  // reading report
            DataTable stdReport = dataset.Tables[0];
            int total_Com = 0;   // assigning initial values of Course to 0
            int total_Mul = 0;
            int total_Net = 0;

            DataTable dt = new DataTable("tbl");
            dt.Columns.Add("Course Enroll", typeof(String));  // creating two
columns

            dt.Columns.Add("Total Students", typeof(int));

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


                String col = stdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    total_Com++;   // incrementing values of each course based 
on user input
                }
                else if (col == "Multimedia Technology")
                {
                    total_Mul++;
                }
                else if (col == "Networking")
                {
                    total_Net++;
                }
            }

            dt.Rows.Add("Computing", total_Com);          // assigning vlaues
            dt.Rows.Add("Multimedia Technology", total_Mul);
            dt.Rows.Add("Networking", total_Net);


            gridWeeklyReport.DataContext = dt.DefaultView; // view in data grid
        }


    }
}
```

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.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 StudentManagementSystem2._0 {
    /// <summary>
    /// Interaction logic for Chart.xaml
    /// </summary>
    public partial class Chart : Window {


        public Chart() {
            InitializeComponent();
            loadChartData();

        }

        private void btnShowChart_Click() {
            throw new NotImplementedException();
        }

        private void loadChartData() {

            var dataset = new DataSet(); // declaring new data set
            dataset.ReadXml(@"E:\College\3rd Year\Application
Development\StudentReport.xml");  // reading main report
            DataTable stdReport = dataset.Tables[0];
            int total_Com = 0;   // assigning initial values of Course to
            int total_Mul = 0;
            int total_Net = 0;

            DataTable dt = new DataTable("tbl");
            dt.Columns.Add("Course Enroll", typeof(String));  // creating two
columns
            dt.Columns.Add("Total Students", typeof(int));

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


                String col = stdReport.Rows[i]["CourseEnroll"].ToString();
                if (col == "Computing")
                {
                    total_Com++;   // incrementing values of each course based
on user input
                }
                else if (col == "Multimedia Technology")
                {
                    total_Mul++;
                }
                else if (col == "Networking")
                {
                    total_Net++;
                }
            }

            dt.Rows.Add("Computing", total_Com);          // final assign
            dt.Rows.Add("Multimedia Technology", total_Mul);
            dt.Rows.Add("Networking", total_Net);

            ((ColumnSeries)mychart).ItemsSource =
                    new KeyValuePair<string, int>[]{
                    new KeyValuePair<string,int>("Computing",total_Com),
                    new KeyValuePair<string,int>("Multimedia
Technology",total_Mul),
                    new KeyValuePair<string,int>("Networking",total_Net)};
        }

        private void btnShowChart_Click(object sender, RoutedEventArgs e) {

        }
    }
}
```