

# Informatics College Pokhara



informatics  
college pokhara

**Application Development**

**CS6004NI**

**Course Work 1**

**Submitted By: Divya Bhattari**  
**London Met ID:** Enter ID Here

**Submitted To:** Ishwor Sapkota  
Module Leader

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

<b>Application architecture &amp; description of the classes ad methods sued</b>	average work with very limited explanation of the classes and methods used
<b>Flow chart, algoriathms and data sctructures used</b>	average work with very limited explanation and missing diagramatic representation.
<b>Reflective essay</b>	Average work with un clear learnings, experience or findings.

### C. Programming Style

<b>Clarity of code,Popper Naming convention &amp; comments</b>	Code is poorly written and lacks comments
<b>System Usability</b>	very poorly developed application

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

### Overall Comment:

Code should be self explainable with less comments. Need some proper naming of the 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



**Module Code & Module Title**

**CS6004NP Application Development**

**Assessment Weightage & Type**

**30% Individual Coursework**

**Year and Semester**

**2018-19 Autumn**

**Name: Divya Bhattarai**

**College ID: NP04CP4S180007**

**University ID: 17031942**

## Contents

<b>1. Introduction</b>	1
1.1 Current Scenario	2
1.2 Proposed System	2
<b>2. User Manual</b>	3
2.1 Login	3
2.2 MainWindow (Student Form)	6
2.3 DisplayReport	13
2.4 Chart	20
<b>3. System Architecture</b>	22
3.1 Architecture Diagram	22
3.2 Class Diagram	23
3.2.1 Individual Diagram	23
3.3 Flowchart of Weekly Report	26
3.4 Algorithm of Weekly Report	27
<b>4. Sorting Algorithm</b>	28
<b>5. Journals</b>	31
<b>6. Reflection</b>	33
<b>7. Conclusion</b>	34
Bibliography	35
<b>Appendix</b>	36

## Lists of Figures

Figure 1: Login Form.....	3
Figure 2: Empty Username or Password cant Login.....	4
Figure 3: Incorrect Login .....	4
Figure 4: Exit Login Form.....	5
Figure 5: Successful Login.....	5
Figure 6: MainWindow (Student Form) .....	6
Figure 7: Filling the data of Student .....	7
Figure 8: Data saved successfully .....	7
Figure 9: Xml file generated.....	7
Figure 10: Importing the data of Student in grid .....	8
Figure 11: Clearing the filled data .....	8
Figure 12: Filling, Saving and Importing the data again .....	8
Figure 13: Adding data generated in xml .....	9
Figure 14: Clearing the filled data again .....	10
Figure 15: Empty value can't Saved .....	10
Figure 16: Empty value can't Import .....	10
Figure 17: Incorrect value can't Saved (Int value in String).....	11
Figure 18: Incorrect value can't Saved (String value in Int).....	11
Figure 19: Incorrect value can't Import (String value in Int).....	11
Figure 20: Incorrect value can't Import (Int value in String).....	12
Figure 21: Exiting Main Window.....	12
Figure 22: Clicking the Report Button appear DisplayReport Window .....	12
Figure 23: DisplayReport Window.....	13
Figure 24: All Data retrieved in grid.....	14
Figure 25: Data sorted by Name .....	14
Figure 26: Data sorted by Date .....	14
Figure 27: Weekly Report .....	15
Figure 28: After clicking Import CSV button (choose option) .....	15
Figure 29: Selecting the .csv file (data.csv) .....	15
Figure 30: data.CSV File (in Excel) .....	16
Figure 31: Adding (Importing) the .csv data in same grid.....	16
Figure 32: Adding the .csv data in same student.xml file .....	17

Figure 33: Adding the .csv data and retrieving it again .....	18
Figure 34: Adding the .csv data and sorted by name.....	18
Figure 35: Adding the .csv data and sorted by date.....	18
Figure 36: Adding the .csv data and showing weekly report (number increment).....	19
Figure 37: Exiting DisplayReport Window.....	19
Figure 38: Clicking the Chart Button appear Chart Window.....	19
Figure 39: Chart Window .....	20
Figure 40: Exiting Chart Window.....	21
Figure 41: Files Generated and Created in D: Drive .....	21
Figure 42: Data.csv file (Importing File in D: Drive).....	21
Figure 43: Architecture Diagram .....	22
Figure 44: Class Diagram Generated in Visual Studio 2019 (SIS).....	23
Figure 45: Login (Individual Diagram) .....	23
Figure 46: MainWindow (Individual Diagram) .....	24
Figure 47: DisplayReport (Individual Diagram) .....	24
Figure 48: Chart (Individual Diagram) .....	25
Figure 49: Flowchart of Weekly Report.....	26

## 1. Introduction

In the course of the Application Development, there is a coursework to develop (design and implement) the System in Visual Studio with C# Language (WPF) of “Student Information System” for a Company. This system is an offline Desktop Application. In this report, there is all about “Student Information System”.

In this System, the application is to keep track of the student’s details, program enrol and registration date. This application allow the user to input the student personal detail including registration date so that a system can generate a weekly enrolment report of the student. And, the System include details like Name, address, contact no, email, program enrol and registration date. This system helps the user to keep track of the student’s details, program enrol and registration date. It is easy to use because it has a simple user interface.

It more necessary in this days because today all the work are transformed on the computerized system, which helps the user to work rapidly and easily. In this system, there is also login system which can open by a user who used the system. An only authorized person can use this application.

There are many functions which are implemented in this System like; to import a record from a text file (e.g. in .CSV format for bulk data input), to generate and display two different reports, listing the students detail like id, name, program enrol and registration date: (a) one sorted by student first name and (b) the other sorted by registration date, to display weekly tabular report showing total number of students enrolled so far in each program offered by the company, to display chart showing total number of student on each program (computing, multimedia, networking etc) and to Save and retrieve the student enrol status with the student details. Other available features are well explained in other sections of the report.



### 1.1 Current Scenario

There are numerous system who keep record of Student data in old traditional system which is Paper-Based System (record in register) which are unsafe and risky to maintain it for long period of time. Using the traditional method, the daily report and weekly report are also difficult to generate. In addition to that, there are some digital system but they are lacking the features which are needed to keep track of the student's details, program enrol and registration date.

### 1.2 Proposed System

The proposed system is digitized system which is specially designed to overcome problem mentioned above. The main objective of the developed system is to keep track of student's details, program enrol and registration date. The system ensures security with the presence of login section. Entry of data and display of data have been made easy with the presence of easy user interface.

## 2. User Manual

There are screenshot below which will illustrate a user how to operate the system. In this section, there is detail discussion about to run the program and detail discussion about different Windows and their button functions. To run this program it is necessary Visual Studio or the system which able to open or a compatible version which it supports. Below are the steps to run the system.

### 2.1 Login

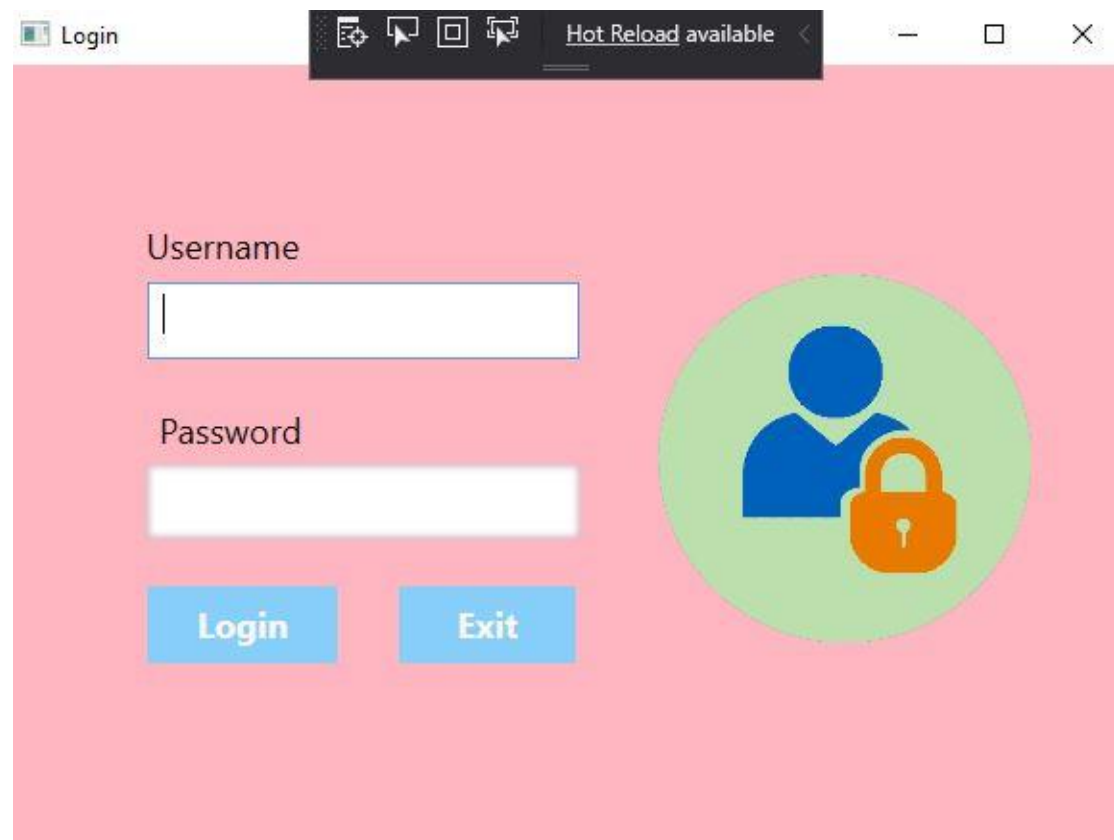


Figure 1: Login Form

The above screenshot shows the Login form of Student System, which is the very first operation of the System. Here, we should enter the correct username and password. Then, we must enter the Login button. After the successful Login, new window will appear that is MainWindow (Student Form). Here, the system cannot login the empty value or incorrect username or password. There is another button called Exit which will close the window of Login form as per our need.

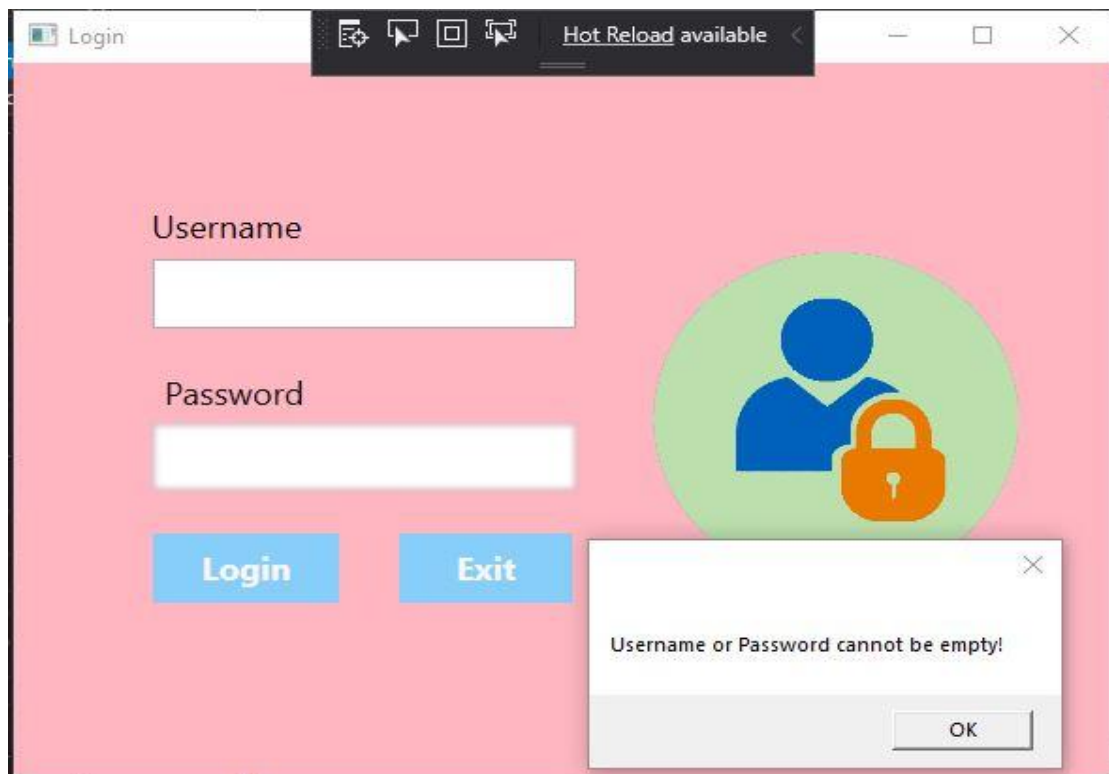


Figure 2: Empty Username or Password cant Login

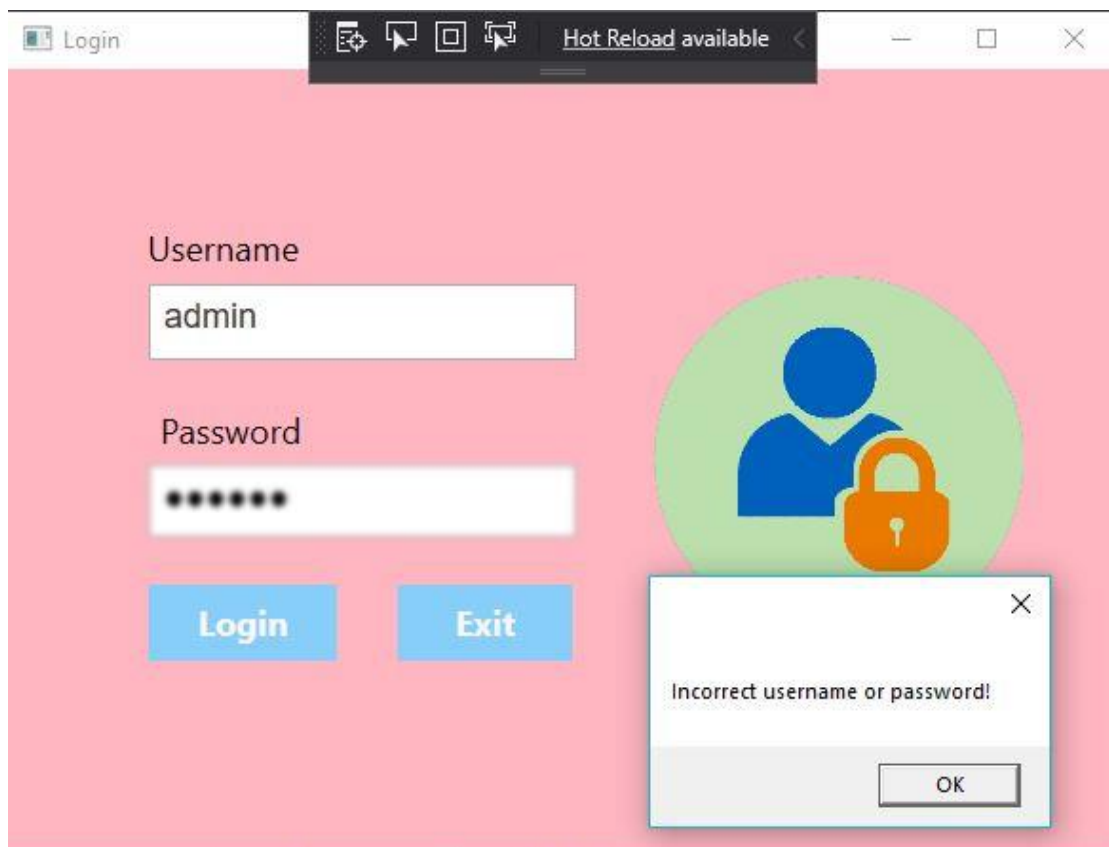


Figure 3: Incorrect Login

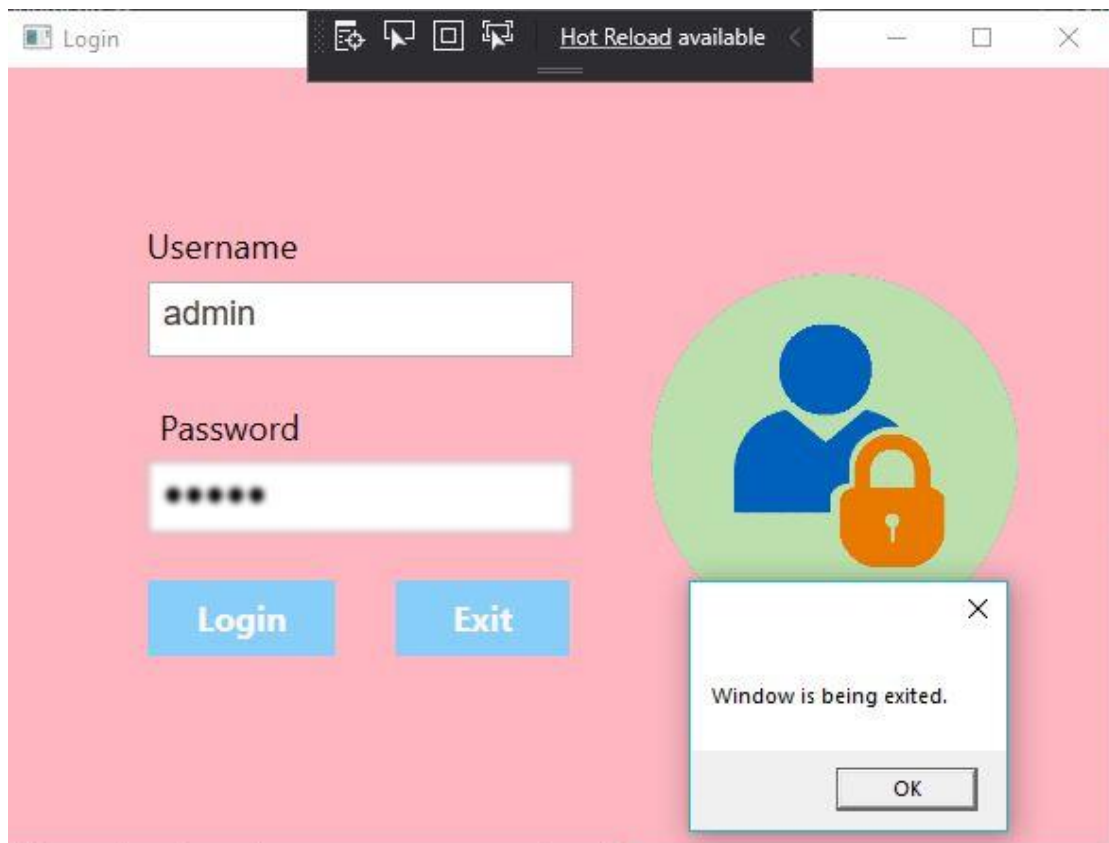


Figure 4: Exit Login Form



Figure 5: Successful Login

## 2.2 MainWindow (Student Form)

ID	Name	Address	Contact	Email	Course Enrol	Registration Date
----	------	---------	---------	-------	--------------	-------------------

Figure 6: MainWindow (Student Form)

After the login, this MainWindow (Student Form) will appear. Now, we must fill the form in text boxes like; Id No, Name, Address, Contact, Email, Course Enrol, Registration. After filling the form we must click on Save button, then it will create a folder in D: Drive in our Computer (or it will create in any folder where we have given path). In that D: Drive, Student.xml file will be generated. After generating the file, we must click on Import button. Import button will show the data in grid form in Window that we have filled. If, we want to fill the form of another Student again, we can Clear the data by clicking Clear button, that we have fill before and can fill again, and Save and Import again as well. We cannot save or import the empty value (data) in this System and we cannot write Integer value in place of String or String value in place of Integer which cannot be saved and import. There is Button called Report, that will open another window i.e. Display Report. There are other buttons like Clear and Exit as per the need we can clear the data and exit the window.

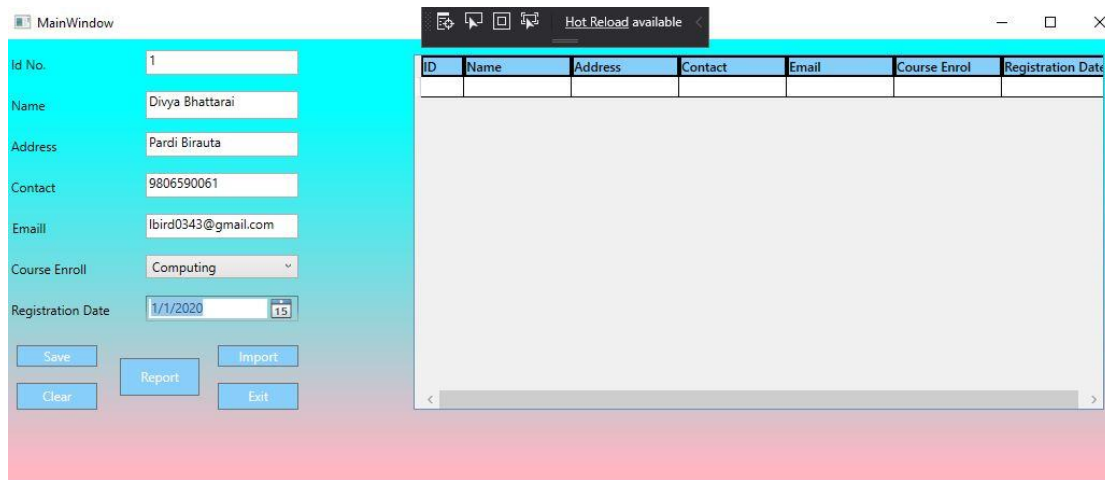


Figure 7: Filling the data of Student

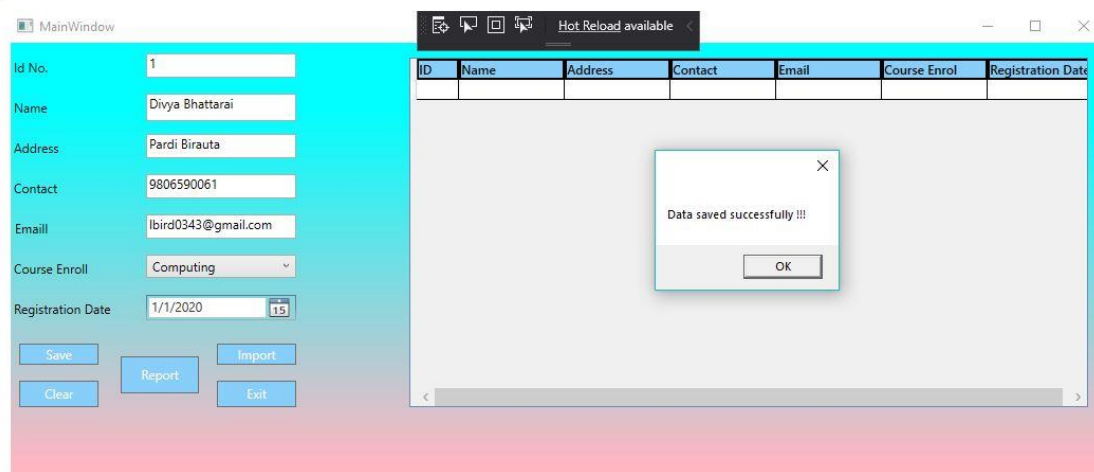


Figure 8: Data saved successfully

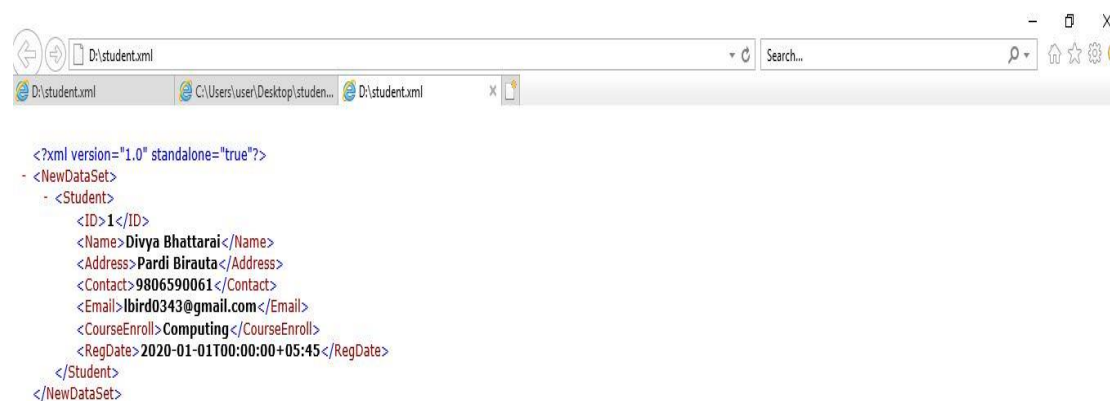


Figure 9: Xml file generated

MainWindow

Hot Reload available

Id No. 1

Name Divya Bhattarai

Address Pardi Biraute

Contact 9806590061

Email lbird0343@gmail.com

Course Enroll Computing

Registration Date 1/1/2020

Save Clear Report Import Exit

ID	Name	Address	Contact	Email	Course Enroll	Registration Date
1	Divya Bhattarai	Pardi Biraute	9806590061	lbird0343@gmail.com	Computing	1/1/2020

Figure 10: Importing the data of Student in grid

MainWindow

Hot Reload available

Id No.

Name

Address

Contact

Email

Course Enroll

Registration Date 1/1/2020

Save Clear Report Import Exit

ID	Name	Address	Contact	Email	Course Enroll	Registration Date
1	Divya Bhattarai	Pardi Biraute	9806590061	lbird0343@gmail.com	Computing	1/1/2020

Figure 11: Clearing the filled data

MainWindow

Hot Reload available

Id No. 4

Name Sneha BK

Address Dulegauda

Contact 9846076987

Email snehabk468@gmail.com

Course Enroll Computing

Registration Date 12/29/2019

Save Clear Report Import Exit

ID	Name	Address	Contact	Email	Course Enroll	Registration Date
1	Divya Bhattarai	Pardi Biraute	9806590061	lbird0343@gmail.com	Computing	1/1/2020
2	Asmita GC	Parsyang	9856075689	asmita123@gmail.com	Networking	12/31/2019
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.com	Multimedia	12/30/2019
4	Sneha BK	Dulegauda	9846076987	snehabk468@gmail.com	Computing	12/29/2019

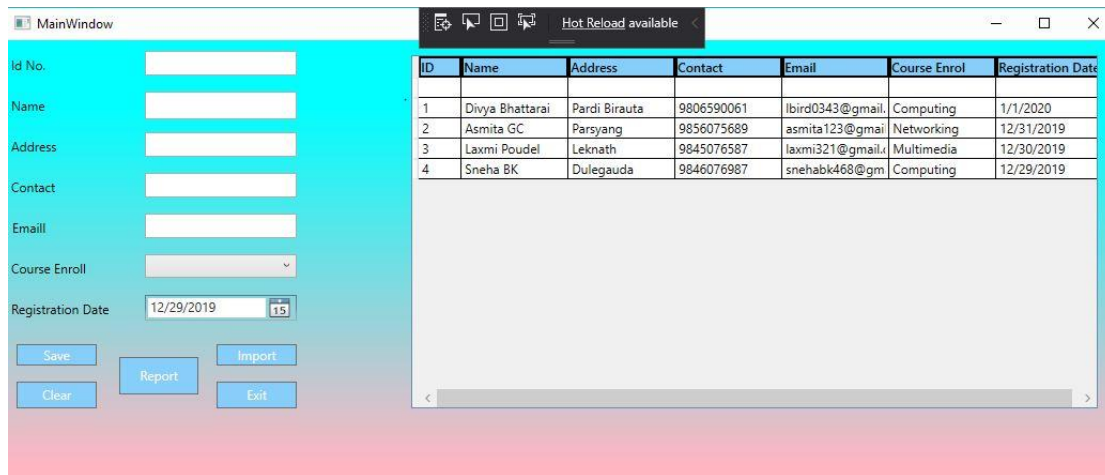
Figure 12: Filling, Saving and Importing the data again





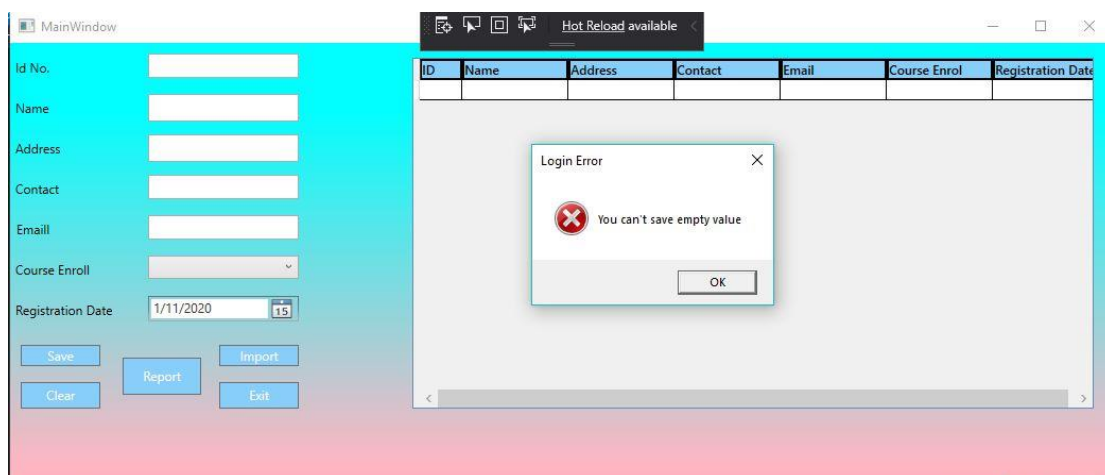
Figure 13: Adding data generated in xml





ID	Name	Address	Contact	Email	Course Enroll	Registration Date
1	Divya Bhattarai	Pardi Biraute	9806590061	lbird0343@gmail.	Computing	1/1/2020
2	Asmita GC	Parsyang	9856075689	asmita123@gmai	Networking	12/31/2019
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.u	Multimedia	12/30/2019
4	Sneha BK	Dulegauta	9846076987	snehabk468@gm	Computing	12/29/2019

Figure 14: Clearing the filled data again

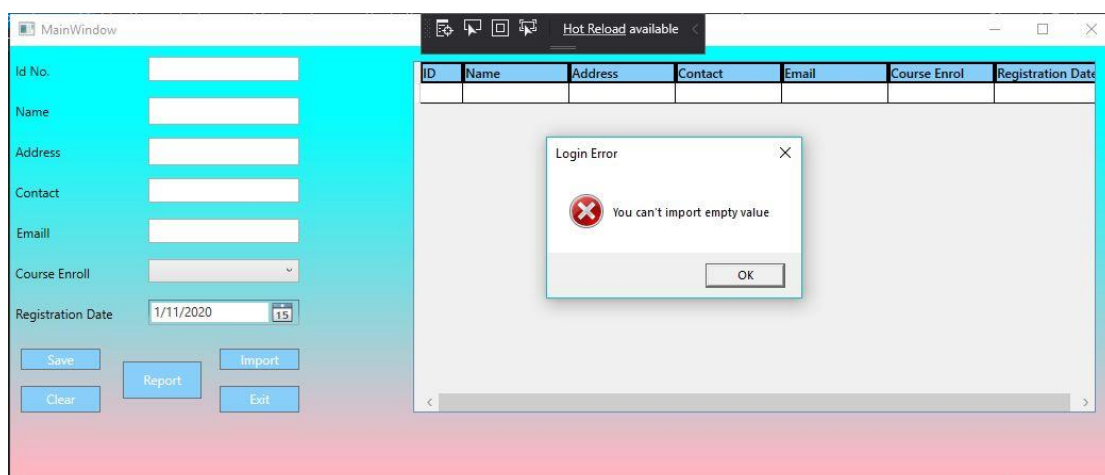


Login Error

You can't save empty value

OK

Figure 15: Empty value can't Saved



Login Error

You can't import empty value

OK

Figure 16: Empty value can't Import

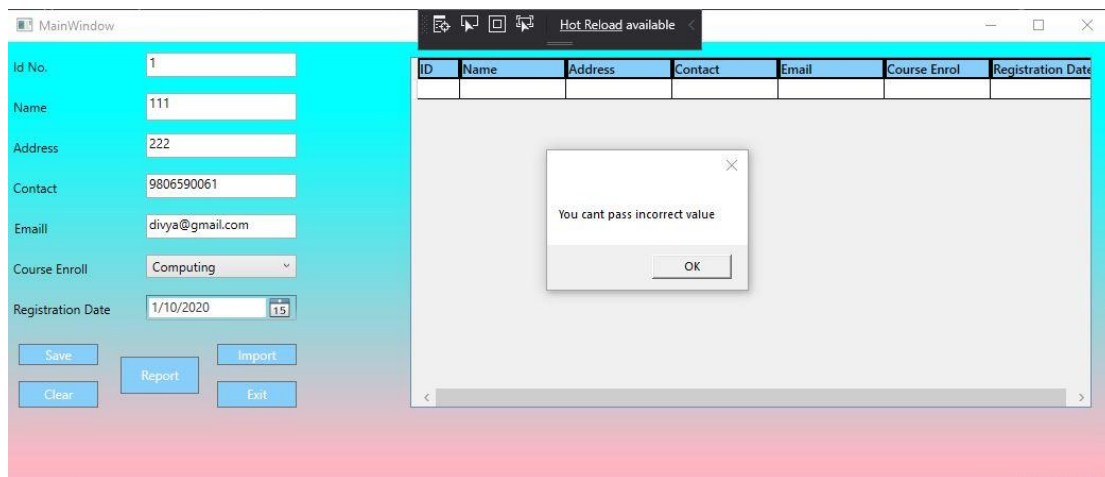


Figure 17: Incorrect value can't Saved (Int value in String)

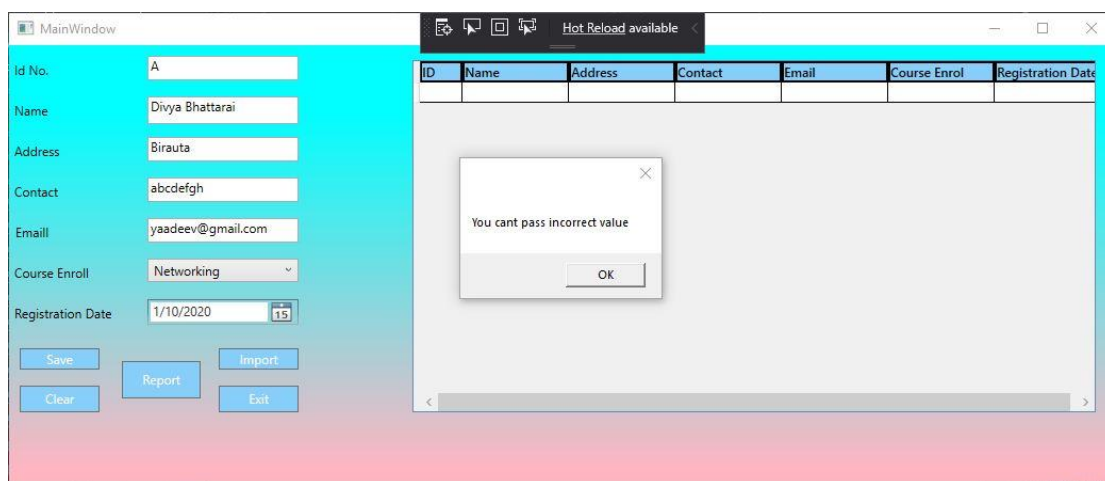


Figure 18: Incorrect value can't Saved (String value in Int)

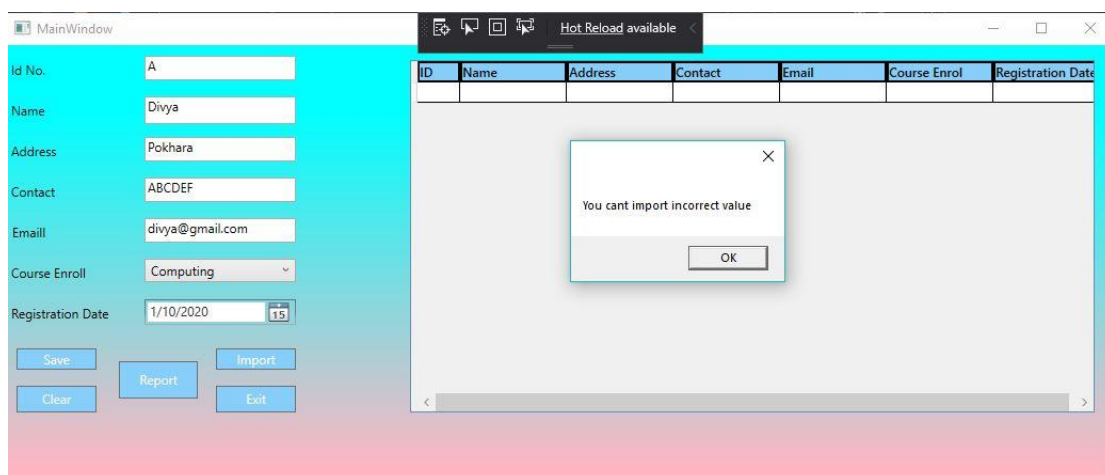


Figure 19: Incorrect value can't Import (String value in Int)

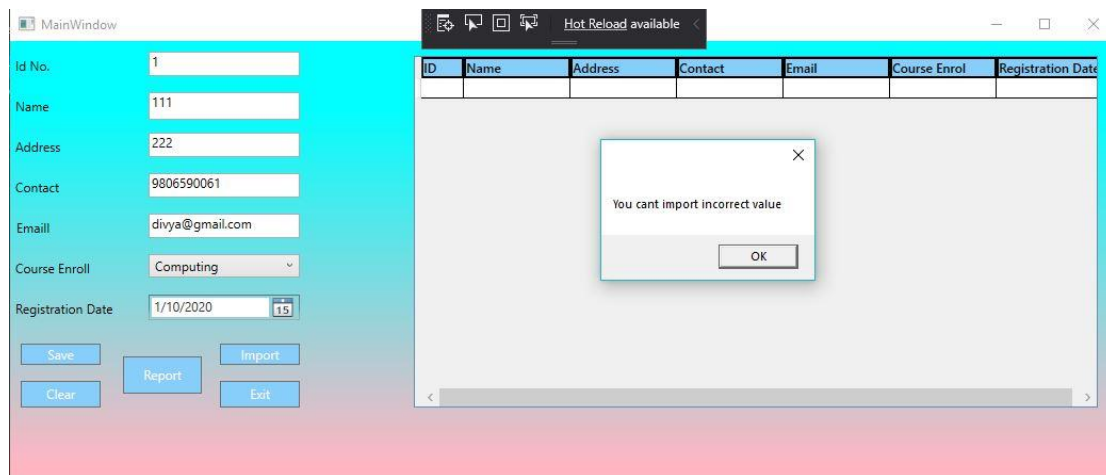


Figure 20: Incorrect value can't Import (Int value in String)

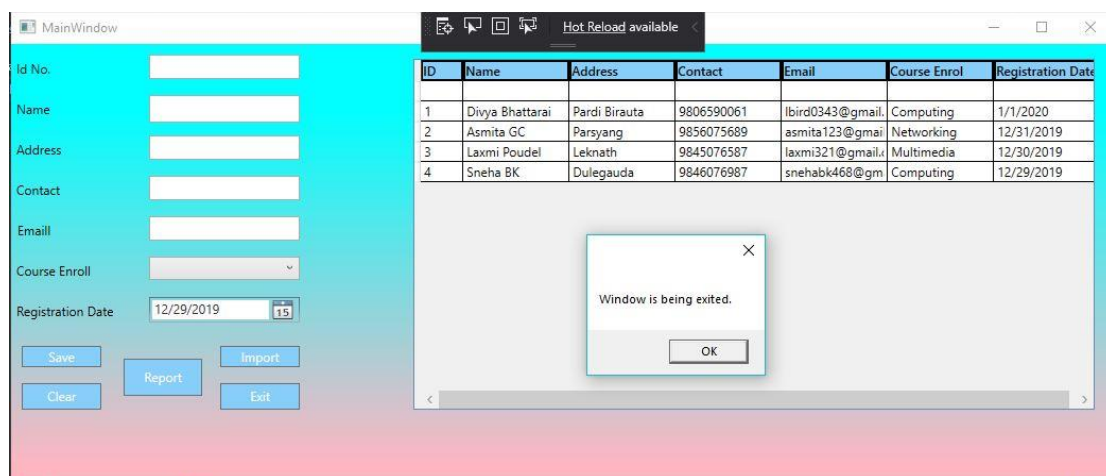


Figure 21: Exiting Main Window

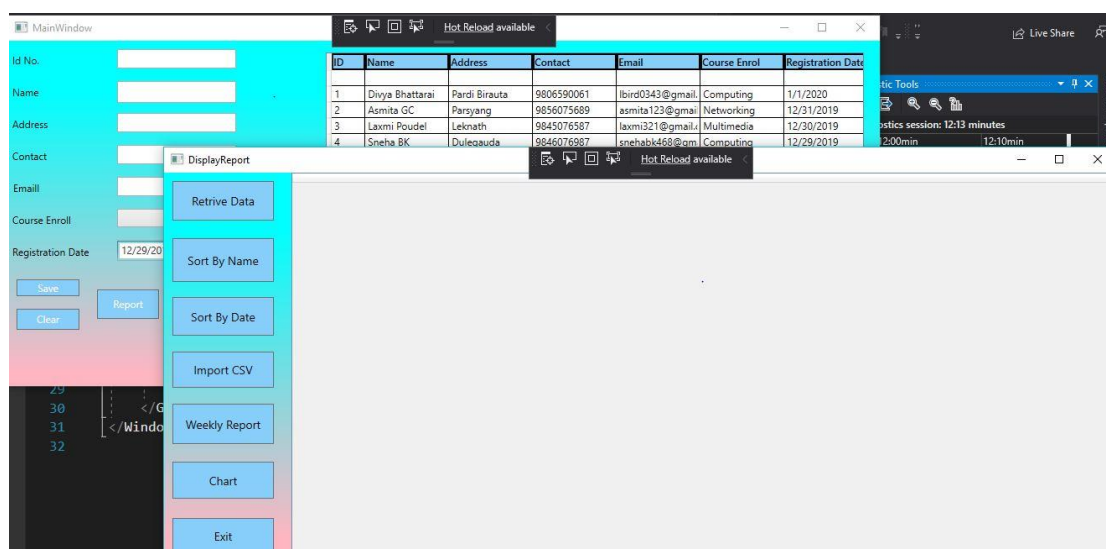


Figure 22: Clicking the Report Button appear DisplayReport Window

## 2.3 DisplayReport

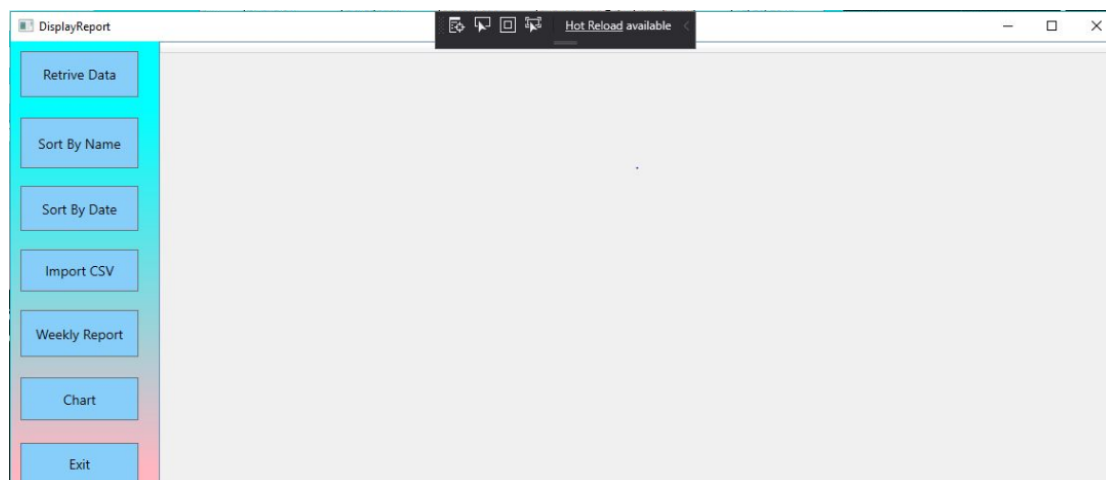
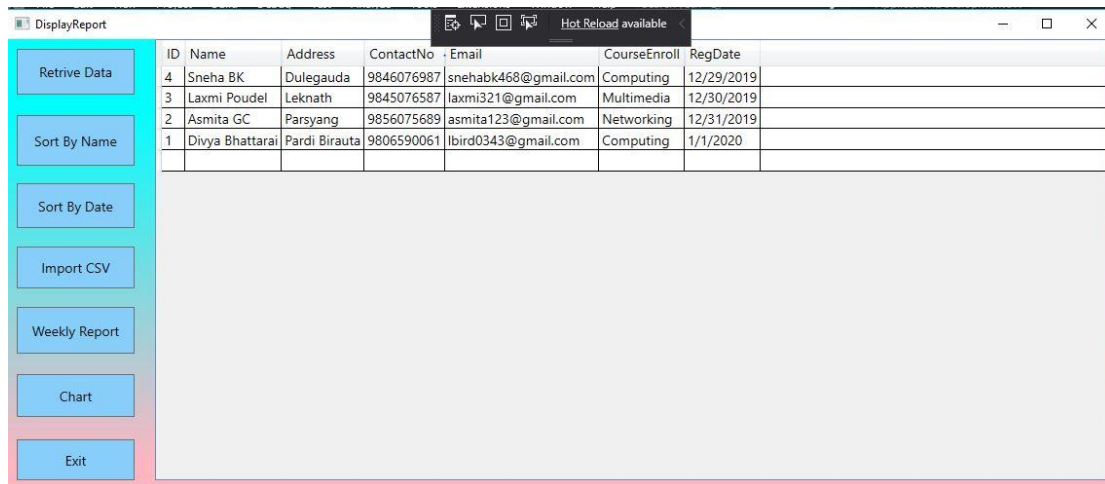


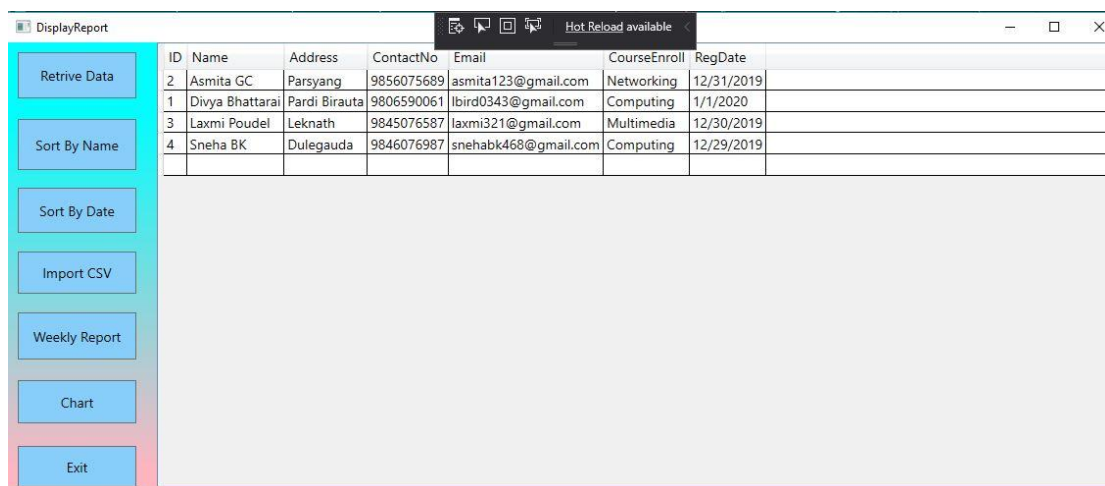
Figure 23: DisplayReport Window

When DisplayReport Window is appear, there are different Buttons which have different function of each. When we click Retrive Data button, all the data of student that we have fill already before in Main Window, will show here in grid view on Display Report Window. Sort By Name button will sorted the data by ascending order to descending in the form of first name which is listed in grid view. Sort By Date button will sorted the data by date order as per registration date which listed in grid view. The button called Weekly Report will show total number of student on each program like computing, multimedia and networking. There is one button i.e. Import CSV, which can upload the files that we have save in our computer in .csv extension, in which it contain the student information like ID, Name, Address, Contact, Email, Course Enrol and RegDate. This csv file will add in the same grid of DisplayReport Window. This csv file will also add in Retrive Data, Sort By Name, Sort By Date, Weekly Report Button and show it on a grid view (Tabular form). Csv file will also add on Student.xml file in D: Drive of our computer. The button Exit will close the Window of DisplayReport, if the user want to close the running system. There is Button called Chart, that will open in another window i.e. Chart.



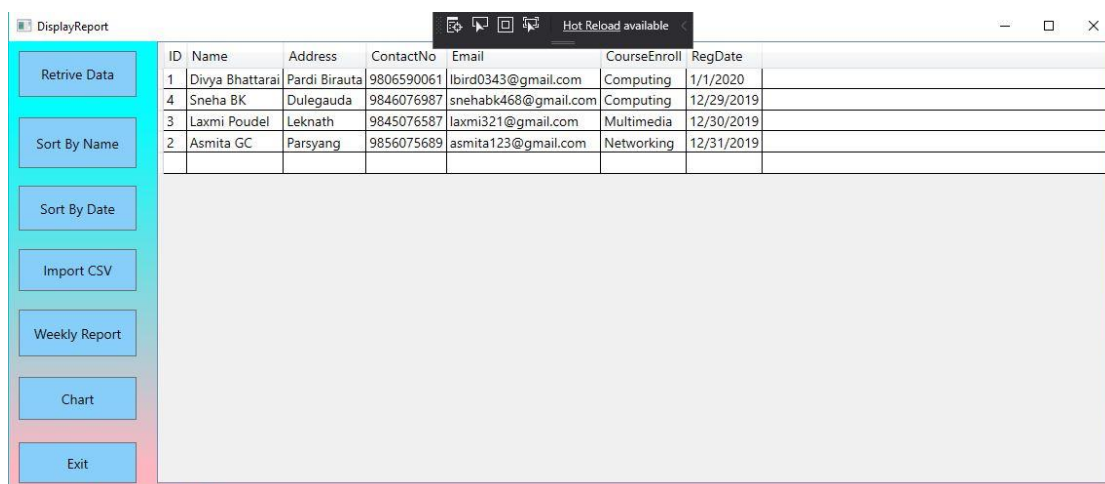
ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
4	Sneha BK	Dulegauda	9846076987	snehabk468@gmail.com	Computing	12/29/2019
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.com	Multimedia	12/30/2019
2	Asmita GC	Parsyang	9856075689	asmita123@gmail.com	Networking	12/31/2019
1	Divya Bhattarai	Pardi Birauta	9806590061	lbird0343@gmail.com	Computing	1/1/2020

Figure 24: All Data retrieved in grid



ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
2	Asmita GC	Parsyang	9856075689	asmita123@gmail.com	Networking	12/31/2019
1	Divya Bhattarai	Pardi Birauta	9806590061	lbird0343@gmail.com	Computing	1/1/2020
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.com	Multimedia	12/30/2019
4	Sneha BK	Dulegauda	9846076987	snehabk468@gmail.com	Computing	12/29/2019

Figure 25: Data sorted by Name



ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
1	Divya Bhattarai	Pardi Birauta	9806590061	lbird0343@gmail.com	Computing	1/1/2020
4	Sneha BK	Dulegauda	9846076987	snehabk468@gmail.com	Computing	12/29/2019
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.com	Multimedia	12/30/2019
2	Asmita GC	Parsyang	9856075689	asmita123@gmail.com	Networking	12/31/2019

Figure 26: Data sorted by Date

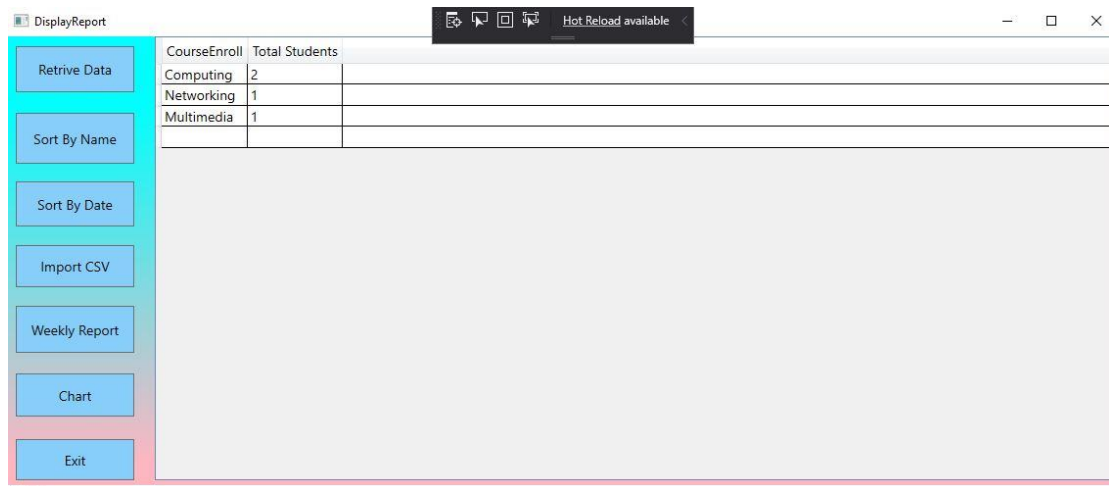


Figure 27: Weekly Report

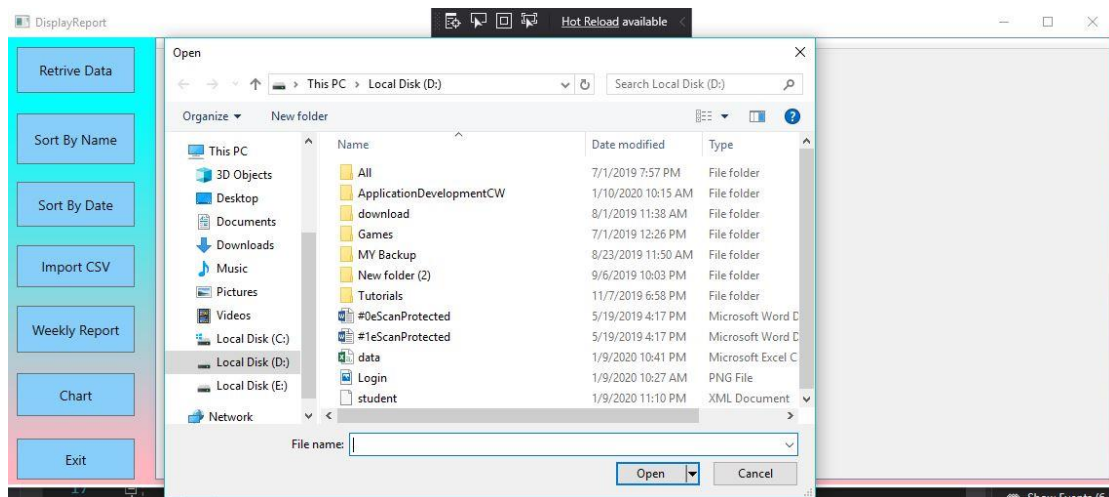


Figure 28: After clicking Import CSV button (choose option)

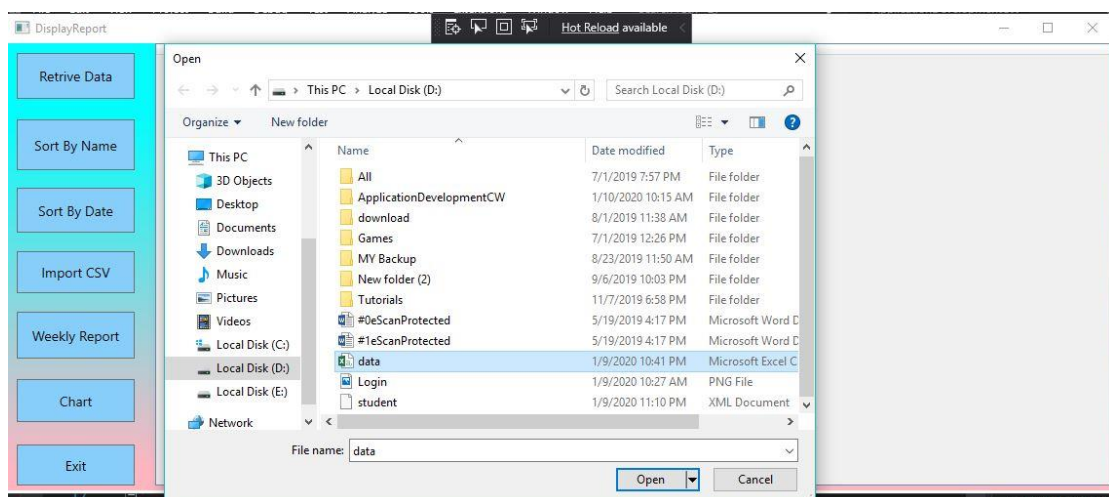


Figure 29: Selecting the .csv file (data.csv)



AutoSave data - Excel Sign in

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard Font Alignment Number Styles Cells Editing

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

ID	Name	Address	Contact	Email	CourseEnroll	RegDate
20	Jiwan Kumar Bhattarai	Shangrila Margh	9846076704	jiwankumar143@gmail.com	Computing	1/1/2020

Figure 30: data.CSV File (in Excel)

DisplayReport Hot Reload available

Retrive Data

Sort By Name

Sort By Date

Import CSV

Weekly Report

Chart

Exit

ID	Name	Address	Contact	Email	CourseEnroll	RegDate
4	Sneha BK	Dulegauda	9846076987	snehabk468@gmail.com	Computing	2019-12-29T00:00:00+05:45
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.com	Multimedia	2019-12-30T00:00:00+05:45
2	Asmita GC	Parsyang	9856075689	asmita123@gmail.com	Networking	2019-12-31T00:00:00+05:45
1	Divya Bhattarai	Pardi Birauta	9806590061	lbird0343@gmail.com	Computing	2020-01-01T00:00:00+05:45
20	Jiwan Kumar Bhattarai	Shangrila Margh	9846076704	jiwankumar143@gmail.com	Computing	1/1/2020

Figure 31: Adding (Importing) the .csv data in same grid



Figure 32: Adding the .csv data in same student.xml file



ID	Name	Address	Contact	Email	CourseEnroll	RegDate
4	Sneha BK	Dulegauda	9846076987	snehabk468@gmail.com	Computing	2019-12-29T00:00:00+05:45
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.com	Multimedia	2019-12-30T00:00:00+05:45
2	Asmita GC	Parsyang	9856075689	asmita123@gmail.com	Networking	2019-12-31T00:00:00+05:45
1	Divya Bhattarai	Pardi Biraute	9806590061	lbird0343@gmail.com	Computing	2020-01-01T00:00:00+05:45
20	Jiwan Kumar Bhattarai	Shangrila Margh	9846076704	jiwankumar143@gmail.com	Computing	1/1/2020

Figure 33: Adding the .csv data and retrieving it again

ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
2	Asmita GC	Parsyang	9856075689	asmita123@gmail.com	Networking	12/31/2019
1	Divya Bhattarai	Pardi Biraute	9806590061	lbird0343@gmail.com	Computing	1/1/2020
20	Jiwan Kumar Bhattarai	Shangrila Margh	9846076704	jiwankumar143@gmail.com	Computing	1/1/2020
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.com	Multimedia	12/30/2019
4	Sneha BK	Dulegauda	9846076987	snehabk468@gmail.com	Computing	12/29/2019

Figure 34: Adding the .csv data and sorted by name

ID	Name	Address	ContactNo	Email	CourseEnroll	RegDate
1	Divya Bhattarai	Pardi Biraute	9806590061	lbird0343@gmail.com	Computing	1/1/2020
20	Jiwan Kumar Bhattarai	Shangrila Margh	9846076704	jiwankumar143@gmail.com	Computing	1/1/2020
4	Sneha BK	Dulegauda	9846076987	snehabk468@gmail.com	Computing	12/29/2019
3	Laxmi Poudel	Leknath	9845076587	laxmi321@gmail.com	Multimedia	12/30/2019
2	Asmita GC	Parsyang	9856075689	asmita123@gmail.com	Networking	12/31/2019

Figure 35: Adding the .csv data and sorted by date

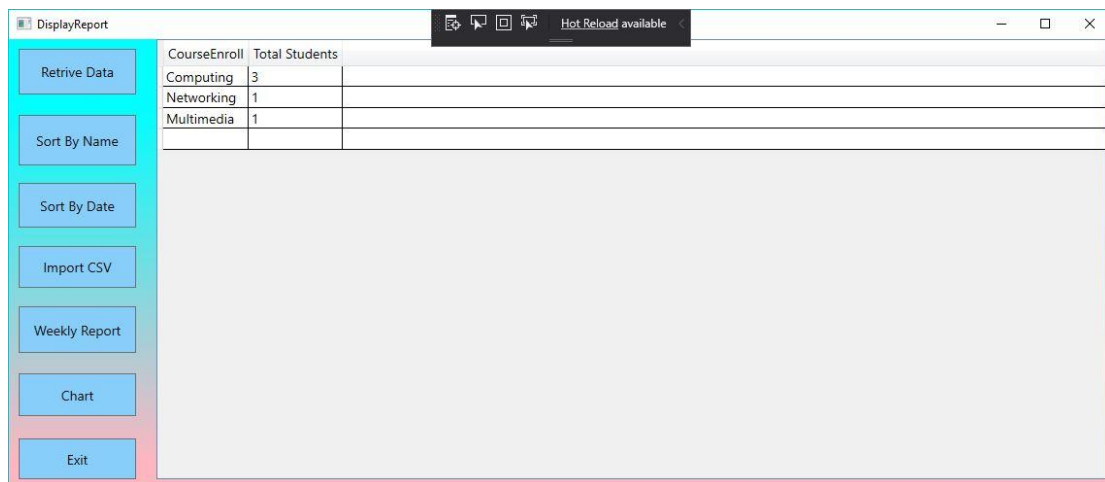


Figure 36: Adding the .csv data and showing weekly report (number increment)

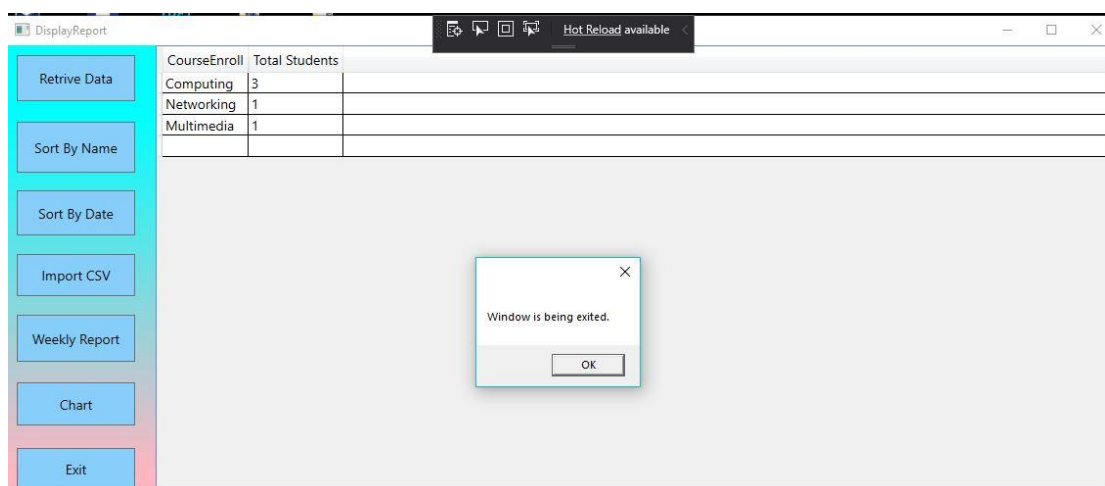


Figure 37: Exiting DisplayReport Window

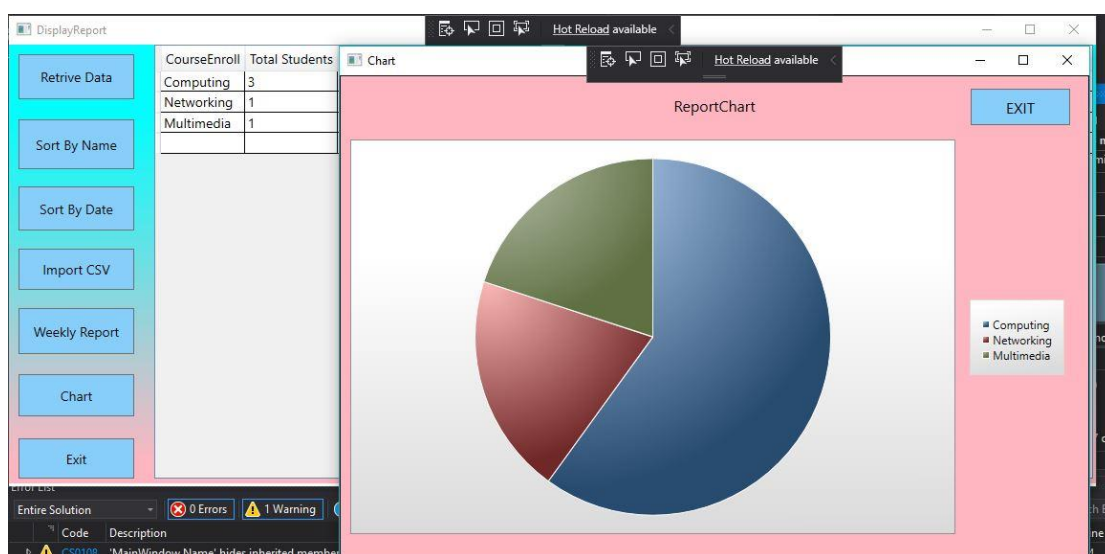


Figure 38: Clicking the Chart Button appear Chart Window

## 2.4 Chart

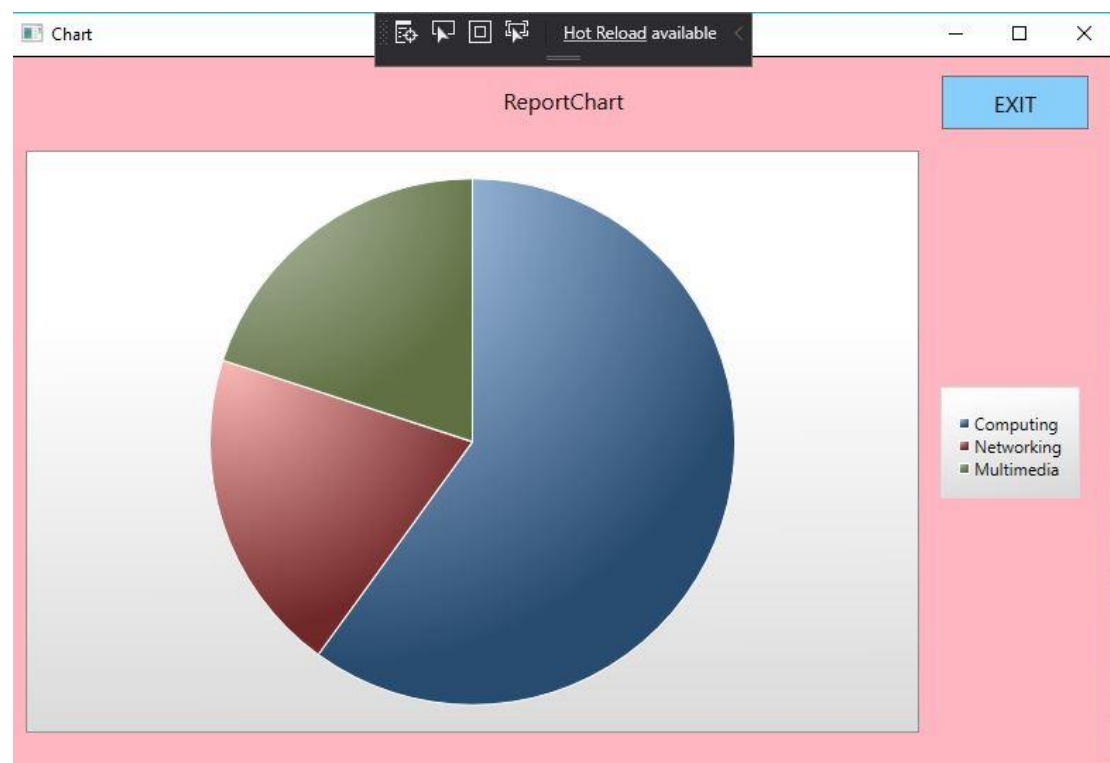


Figure 39: Chart Window

Clicking the Chart button in DisplayReport Window, these Chart Window will appear. Here, we can see a chart is generating in the Window. This chart shows the total number of student on each program like computing, multimedia and networking as per the Student details, which is generated in Pie-chart. There is only one button in this Chart Window i.e. Exit which will close the Chart Window as per the need of user. After exit we can see the files that are generated in D: Drive (where we have given path) in our computer in different extensions that we have maintain data details of Students during this running System.

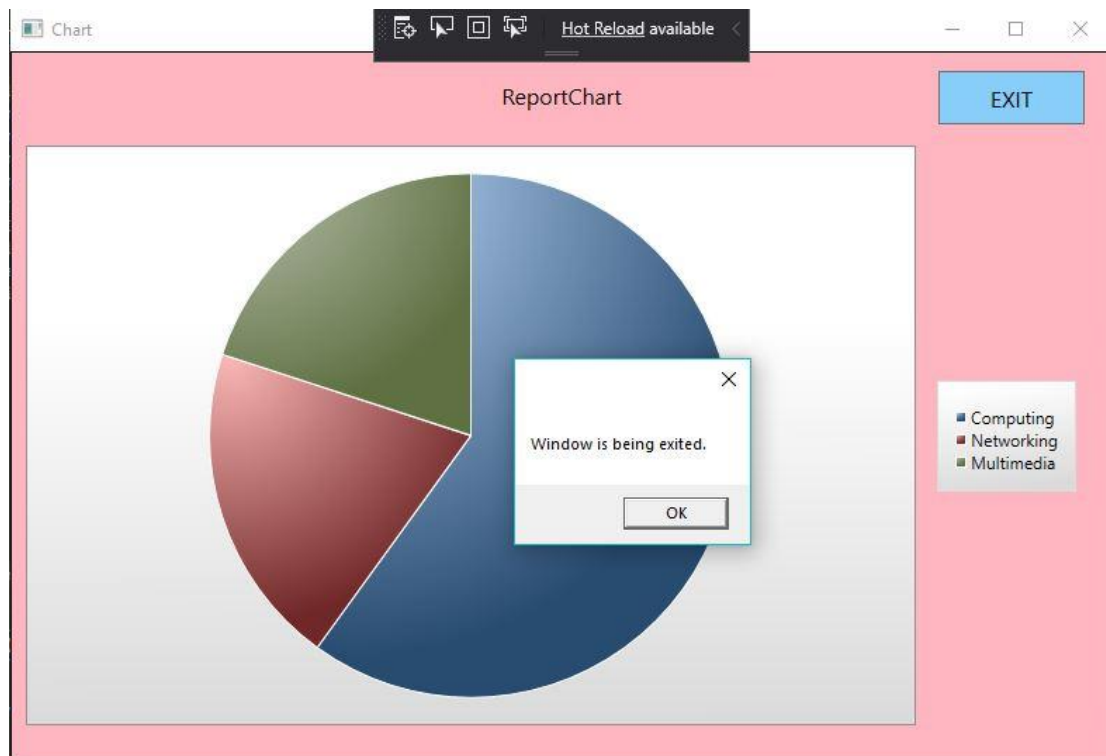


Figure 40: Exiting Chart Window

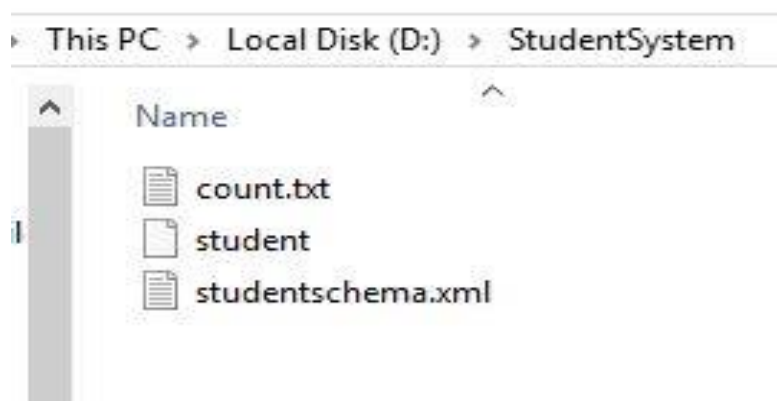


Figure 41: Files Generated and Created in D: Drive

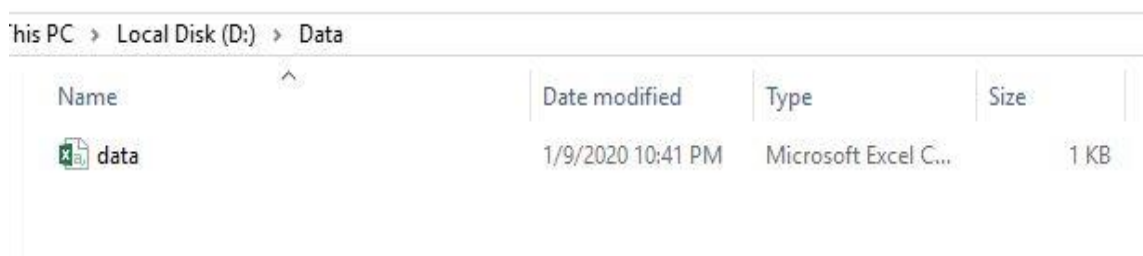


Figure 42: Data.csv file (Importing File in D: Drive)

### 3. System Architecture

#### 3.1 Architecture Diagram

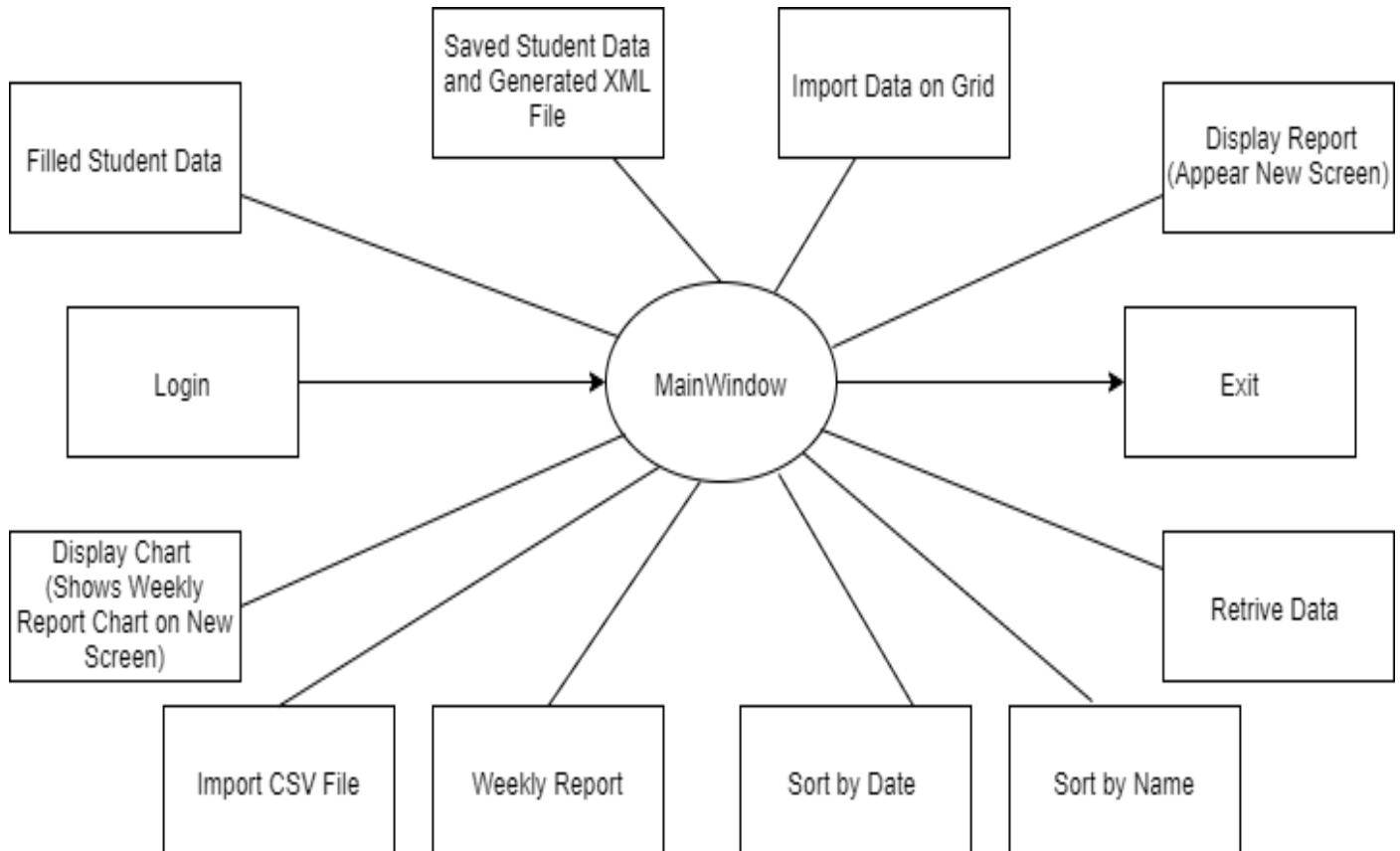


Figure 43: Architecture Diagram

The above figure represents the architecture of the developed system. At first, user need to Login to the system, for which the user need to input the correct credentials. After logging into the system with correct credentials, the system will display the MainWindow (Student Form) which is the main panel of the Student Information System. There are different functions in this system which we can clearly seen in above diagram. As per the user need they can run this system as they want by filling the data and clicking on different buttons. After running this system, at last we must Exit and the system will be closed.

### 3.2 Class Diagram

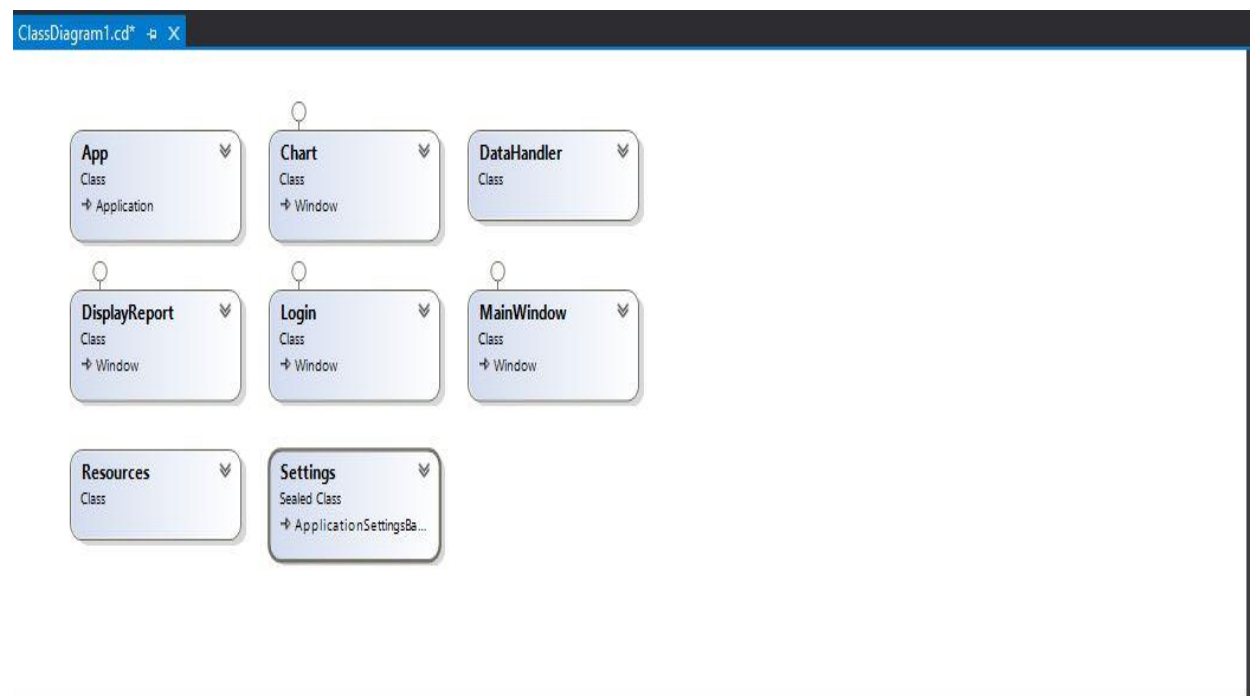


Figure 44: Class Diagram Generated in Visual Studio 2019 (SIS)

#### 3.2.1 Individual Diagram

##### a) Login

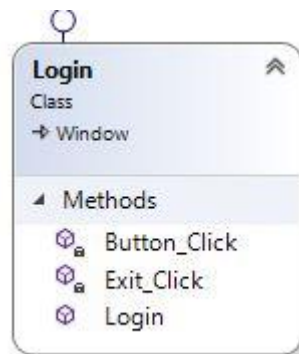


Figure 45: Login (Individual Diagram)

## b) MainWindow

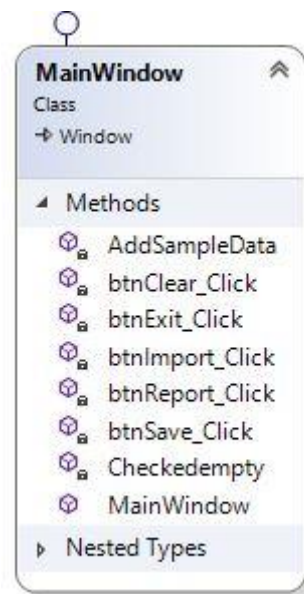


Figure 46: MainWindow (Individual Diagram)

## c) DisplayReport

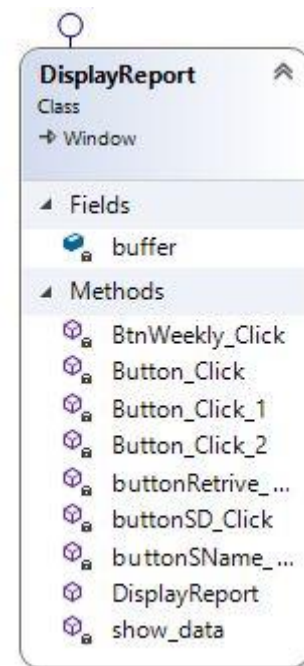


Figure 47: DisplayReport (Individual Diagram)

d) Chart

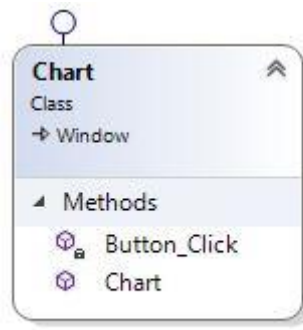


Figure 48: Chart (Individual Diagram)



### 3.3 Flowchart of Weekly Report

Flowchart for Weekly Report of Student Information System is shown below as per the CourseEnroll and Total Students:

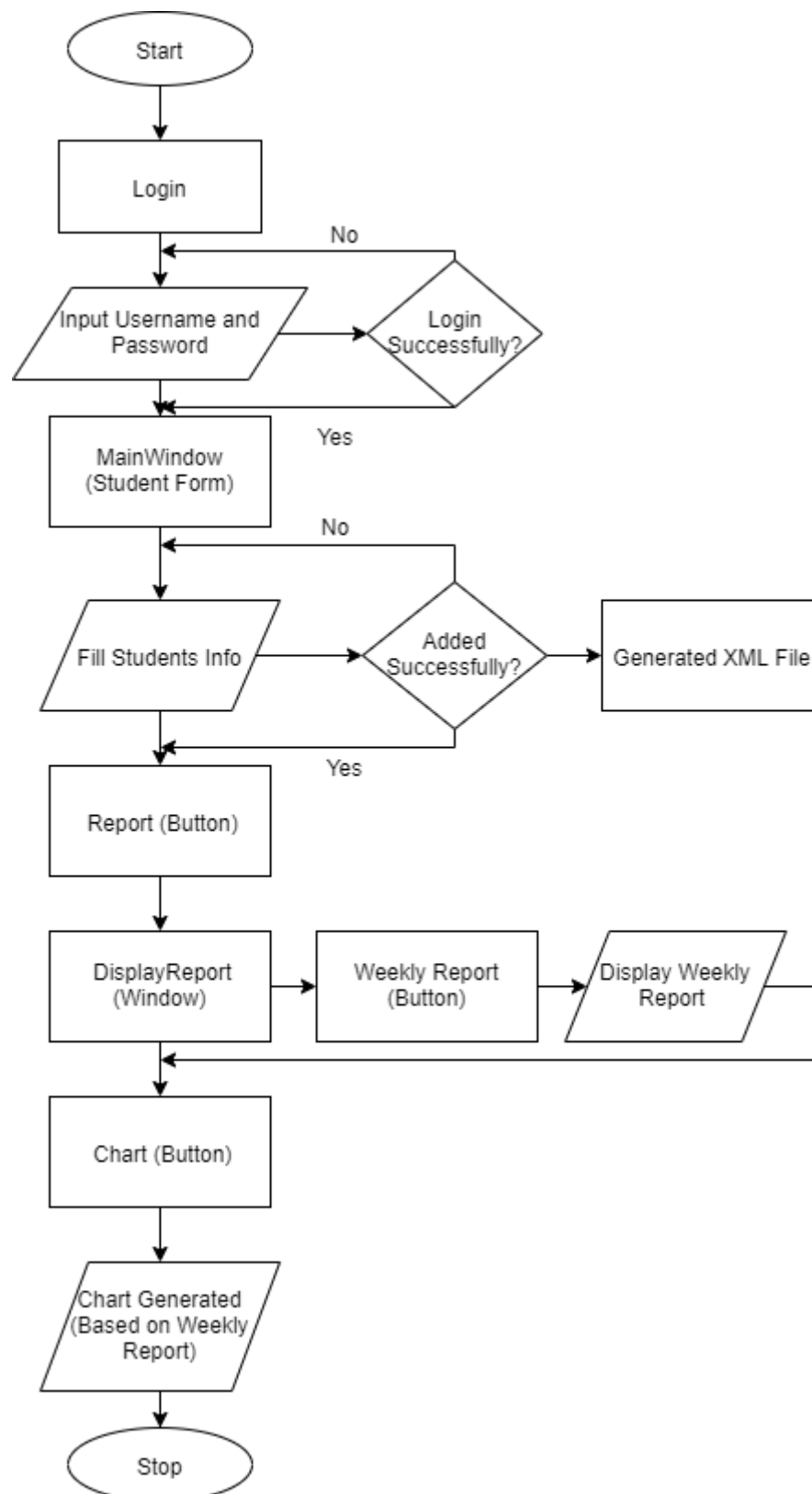


Figure 49: Flowchart of Weekly Report

### 3.4 Algorithm of Weekly Report

Algorithm for Weekly Report of Student Information System is given below in the steps as per the CourseEnroll and Total Students:

**Step 1:** Start

**Step 2:** Login Form will appear

**Step 3:** Input Username and Password

**Step 4:** Login will Success after the valid Username and Password

**Step 5:** MainWindow (Student Form) will appear

**Step 6:** We must fill the Student information

**Step 7:** Student Data will added successfully

**Step 8:** XML file will generated (in D: Drive or where we have given path)

**Step 9:** We must click on Report Button

**Step 10:** DisplayReport Window will appear (or after this directly can go to Chart Button)

**Step 11:** We must click on Weekly Report

**Step 12:** Weekly Report data will shown on grid view as per StudentEnroll and total Student.

**Step 13:** Click on Chart button

**Step 14:** Chart will generated in next window based Weekly Report data

**Step 15:** End

## 4. Sorting Algorithm

The Sorting Algorithm used in Student Information System is Bubble Sorting Algorithm.

A bubble sort, also called a sinking sort or exchange sort, is a sorting algorithm that compares adjacent pairs and swaps them if necessary, causing the items to "bubble" up toward their proper position. The process continues until no swaps are necessary. A bubble sort always brings the highest value to the top during the first pass through the list, the second highest value to the second topmost position during the second pass, and so forth for each successive pass. This pattern enables a common optimization of bubble sort: not all elements need to be processed in every pass. The number of elements processed during a particular pass may be reduced to  $n - i$ , where  $i$  is the number of passes previously completed (Chegg Study, 2003).

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



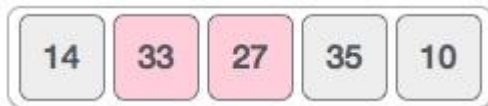
Bubble sort starts with very first two elements, comparing them to check which one is greater.



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.

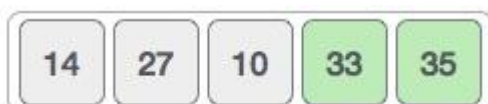


We swap these values. We find that we have reached the end of the array.

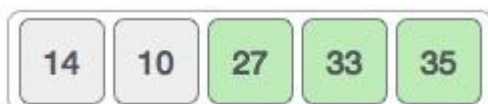
After one iteration, the array should look like this –



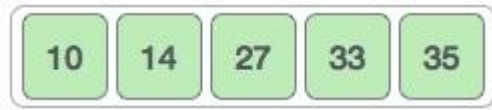
To be precise, we are now showing how an array should look like after each iteration. After the second iteration, it should look like this –



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.



(tutorialspoint, 2020)

## 5. Journals

Some other similar journals were taken as a references such as:

a) An interactive student information system

This paper describes an on-line interactive student information system designed for graduate level student information at Mississippi State University. It was designed with three objectives in mind. The first was to keep graduate students abreast of the requirements to be met for an advanced degree. The second objective was to aid the Office of Graduate Studies and the Registrar in certifying the completion of requirements for advanced degrees. Thirdly, the system was designed so that information retrieval for various colleges, departments and federal agencies could be accomplished without a time-consuming manual search of student records. A description of the software and data files is given, as well as a description of the major report forms produced by the system. In addition, the operation of the system will be briefly described (Cofer, 1976).

b) A computer-based information system to aid in undergraduate advisement and curriculum development

This paper outlines the salient features of a computer-based undergraduate student information system. The system is being designed and implemented by this author in partial fulfilment of the requirements for a Master of Science degree in Computer Science from the University of Southern Mississippi (USM). The major thrust of this paper is the description of the proposed system's capabilities rather than an elucidation of the various design strategies and theory (Greenberg, 1979).

c) Smart System Studied: New Approaches to Natural Language Understanding

This paper aims to focus on the smart system as optimized expert knowledge acquisition system as new approaches to natural language understanding system. This method can finish fine processing for any text segment instantly. The module's precision machining can adopt big production method that combines on the line first, complete coverage and accurate grasp each language point and knowledge point and original point even their respective combination. Its characteristics are teachers and students can use the text analysed method to do the fine processing of the same knowledge module, and only in Chinese or English, through the selection of keywords and terminology and knowledge modules that can be used as the menu to be selected as the way to achieve knowledge with the system. The result is the learning environment that enables human-computer collaboration system namely smart system to optimize the expert knowledge acquisition and the natural language understanding as a research field that has great significance to human beings. Its significance is that this learning environment software based on the National Excellent Courses by using the language chess with the feature of the introduction on the knowledge big production mode for the textual knowledge module finishing at Peking University (Xiaohui Zou, Shunpeng Zou, Xiaoqun Wang, 2019).

## 6. Reflection

The developed system is Student Information System which was developed using Visual Studio in C# - desktop application. This application keeps track of the student's details, program enrols and registration date. The application must allow the user to input the student personal detail including registration date so that a system can generate a weekly enrolment report of the student. System must include detail like Name, address, contact no, email, program enrol, and registration date.

Working on the Visual Studio is new experience for me. I got to know many more thing about the desktop application. The most complex part I found was to make the chart to display total number of student on each program. I also spend more time to import a record from csv file as I had to tackle with several errors was new aspect for me. In addition to that, sorting of data form, in the grid was a new thing for me. Creating a class diagram within the visual studio helps me in documentation phase.

With the growing of technology, the visual studio and its community helps beginner developer like us, to pace our development speed. Making this application flourish the knowledge on making the desktop application. Overall evaluation with the great support of friends and teachers had a great experience with the Application Development of the "Student Information System".



## 7. Conclusion

The main objective of this coursework is to develop (design and implement) the System in Visual Studio with C# Language (WPF) of “Student Information System” for a Company. This system is an offline Desktop Application.

There are many functions which are implemented in this System like; to import a record from a text file (e.g. in .CSV format for bulk data input), to generate and display two different reports, listing the students detail like id, name, program enrol and registration date: (a) one sorted by student first name and (b) the other sorted by registration date, to display weekly tabular report showing total number of students enrolled so far in each program offered by the company, to display chart showing total number of student on each program (computing, multimedia, networking etc) and to Save and retrieve the student enrol status with the student details. Other available features are well explained above in the different section of report.

For this system, I have done many research to know about the things, that I was confused about. While doing this coursework, I had a lot of problems and as a guide I took the help from reliable website, related videos and also from our module teacher. This project gave student like us the opportunity to try our new skills in practice.

I am now more adept at coding, debugging and publishing the application in the windows environment. Importing and operating csv data files were challenging at first but it became a lot easier later on. I managed to solve the problems by consulting with my tutor, friends and the Internet. The drag and drop feature of the visual studio which I liked most. It was quite difficult for me when my system is crashed it makes me difficult to rebuild it. So from this problem I know we need to back up the files which were error free and test in another file.

Lastly, I was more familiar with the Visual Studio. I am able to debug the errors. There are many facilities in the visual studio which helps to know about it and also able to use when necessary. The application development helps to improve in coding in C# language.

## Bibliography

Chegg Study, 2003. *Chegg Study*. [Online]

Available at: <http://www.chegg.com/homework-help/definitions/bubble-sort-3>

[Accessed 05 Jan 2020].

Cofer, J. E., 1976. An interactive student information system. *ACM Digital Library*, April.

Greenberg, S. H., 1979. A computer-based information system to aid in undergraduate advisement and curriculum development. *ACM Digital Library*, April.

tutorialspoint, 2020. *tutorialspoint*. [Online]

Available at:

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

[Accessed 05 Jan 2020].

Xiaohui Zou, Shunpeng Zou, Xiaoqun Wang, 2019. Smart System Studied: New Approaches to Natural Language Understanding. *ACM Digital Library*, July.

## Appendix

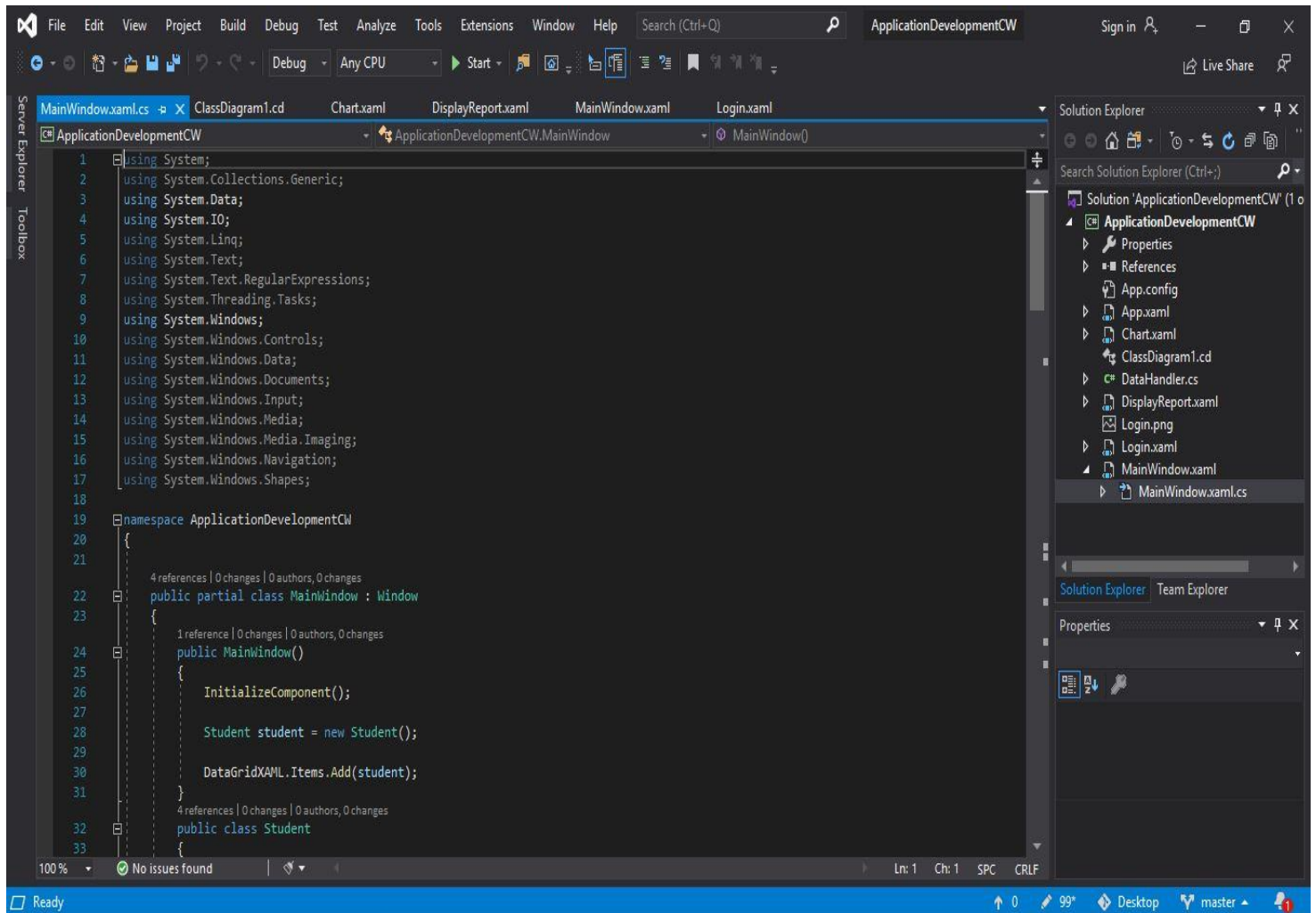


Figure: Showing Student Information System Code in Visual Studio

**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 ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>

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

```
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    string username = textBox1.Text;
    string password = textBox2.Password;

    if (username == "")
    {
        MessageBox.Show("Username cannot be empty!");
    }
    else if (password == "")
    {
        MessageBox.Show("Password cannot be empty!");
    }
    else if (password == "admin" && username == "admin")
    {
        MessageBox.Show("Welcome! Now, Please fill the Student Form...");
        this.Hide();

        MainWindow mainWindow = new MainWindow();
        mainWindow.Show();
    }

    else
```

```
{  
    MessageBox.Show("Incorrect username or password!");  
}  
  
}  
  
private void Exit_Click(object sender, RoutedEventArgs e)  
{  
    MessageBox.Show("Window is being exited.");  
    this.Close();  
}  
}  
}
```

**MainWindow.xaml.cs**

```
using System;

using System.Collections.Generic;

using System.Data;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;


namespace ApplicationDevelopmentCW
{

    public partial class MainWindow : Window
    {

        public MainWindow()
```

```
{  
    InitializeComponent();  
  
    Student student = new Student();  
  
    DataGridXAML.Items.Add(student);  
}  
  
public class Student  
{  
    public string ID { get; set; }  
  
    public string Name { get; set; }  
  
    public string Address { get; set; }  
  
    public string Contact { get; set; }  
  
    public string CourseEnroll { get; set; }  
  
    public string RegDate { get; set; }  
    public string Email { get; internal set; }  
}  
  
private void btnSave_Click(object sender, RoutedEventArgs e)  
{
```



```
Checkedempty();

var handler = new DataHandler();

var dataSet = handler.CreateDataSet();

AddSampleData(dataSet);

MessageBox.Show("Data saved successfully !!!");

if (File.Exists(@"D:\student.xml"))
{
    dataSet.ReadXml(@"D:\student.xml");
    dataSet.WriteXml(@"D:\student.xml");
}
else
{
    dataSet.WriteXml(@"D:\student.xml");
}

}

private void AddSampleData(DataSet dataSet)
{
    var dr1 = dataSet.Tables["Student"].NewRow();
    dr1["ID"] = textId.Text;
    dr1["Name"] = textName.Text;
    dr1["Address"] = textAddress.Text;
    dr1["Contact"] = textContact.Text;
    dr1["Email"] = textEmail.Text;
```

```
        dr1["CourseEnroll"] = textCourse.Text;

        string text = textDate.Text;

        dr1["RegDate"] = text;

        dataSet.Tables["Student"].Rows.Add(dr1);

    }

    private void Checkedempty()
    {
        var ID = textId.Text;

        var Name = textName.Text;

        var Address = textAddress.Text;

        var Contact = textContact.Text;

        var Email = textEmail.Text;

        var CourseEnroll = textCourse.Text;

        string text = textDate.Text;

        var RegDate = text;

        if (ID.Equals("") || Name.Equals("") || Address.Equals("") ||
Contact.Equals("") || Email.Equals("") || CourseEnroll.Equals("") ||
RegDate.Equals(""))
        {
            MessageBox.Show("You can't pass empty value", "Login Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

```
private void btnImport_Click(object sender, RoutedEventArgs e)
{
    Checkedempty();

    Student dataStudent = new Student();

    dataStudent.ID = textId.Text;

    dataStudent.Name = textName.Text;

    dataStudent.Address = textAddress.Text;

    dataStudent.Contact = textContact.Text;

    dataStudent.Email = textEmail.Text;

    dataStudent.CourseEnroll = textCourse.Text;

    dataStudent.RegDate = textDate.Text;

    DataGridXAML.Items.Add(dataStudent);
}
```

```
private void btnClear_Click(object sender, RoutedEventArgs e)
{
    textId.Clear();

    textName.Clear();

    textAddress.Clear();

    textContact.Clear();

    textCourse.SelectedIndex = -1;

    textEmail.Clear();
}
```

```
}

private void btnExit_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Window is being exited.");
    this.Close();
}

private void btnReport_Click(object sender, RoutedEventArgs e)
{
    DisplayReport displayReport = new DisplayReport();
    displayReport.Show();
}

}

}
```

**DisplayReport.xaml.cs**

```
using Microsoft.Win32;
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 ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for DisplayReport.xaml
    /// </summary>
    public partial class DisplayReport : Window
    {
        DataTable buffer;

        public DisplayReport()
```

```
{
    InitializeComponent();
}

private void show_data()
{
    String dataXML = @"D:\student.xml";
    DataSet dataset = new DataSet();
    dataset.ReadXml(dataXML);

    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("Email", typeof(String));
    buffer.Columns.Add("CourseEnroll", typeof(String));
    buffer.Columns.Add("RegDate", 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 dataV = new DataView(buffer);
    DataGridReport.ItemsSource = dataV;
}

private void buttonRetrive_Click(object sender, RoutedEventArgs e)
{
    show_data();
}

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

private void buttonSD_Click(object sender, RoutedEventArgs e)
{
    DataView dataV = new DataView(buffer);
    dataV.Sort = "RegDate ASC";
    DataGridReport.ItemsSource = dataV;
}

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

```
var dataSet = new DataSet();
dataSet.ReadXml(@"D:\student.xml");
OpenFileDialog openFileDialog = new OpenFileDialog();
if (openFileDialog.ShowDialog() == 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 dr1 = dataSet.Tables["Student"].NewRow();
            dr1["ID"] = values[0];
            dr1["Name"] = values[1];
            dr1["Address"] = values[2];
            dr1["Contact"] = values[3];
            dr1["Email"] = values[4];
            dr1["CourseEnroll"] = values[5];
            dr1["RegDate"] = values[6];
            dataSet.Tables["Student"].Rows.Add(dr1);

            dataSet.WriteXml(@"D:\student.xml");
        }
    }

    DataGridReport.ItemsSource =
dataSet.Tables["Student"].DefaultView;
}
```



```
}
```

```
private void BtnWeekly_Click(object sender, RoutedEventArgs e)
{
    var dataset = new DataSet(); // declaring new data set
    dataset.ReadXml(@"D:\student.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("CourseEnroll", 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 == "Networking")
```

```
        {
            total_Mul++;
        }
        else if (col == "Multimedia")
        {
            total_Net++;
        }
    }

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

    DataGridReport.ItemsSource = dt.DefaultView;
}

private void Button_Click_1(object sender, RoutedEventArgs e)
{
    Chart chart = new Chart();
    chart.Show();
}

private void Button_Click_2(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Window is being exited.");
    this.Close();
}
}
```

**Chart.xaml.cs**

```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.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 ApplicationDevelopmentCW
{
    /// <summary>
    /// Interaction logic for Chart.xaml
    /// </summary>
    public partial class Chart : Window
    {
        public Chart()
        {
            InitializeComponent();
        }
    }
}
```

```
var dataset = new DataSet(); // declaring new data set
dataset.ReadXml(@"D:\student.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("CourseEnroll", 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 == "Networking")
    {
        total_Mul++;
    }
    else if (col == "Multimedia")
    {
        total_Net++;
    }
}
```

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

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

private void Button_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Window is being exited.");
    this.Close();
}
}
}
```

**Student.xml** (Generated in D: Drive of our Computer)

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Student>
    <ID>4</ID>
    <Name>Sneha BK</Name>
    <Address>Dulegauda</Address>
    <Contact>9846076987</Contact>
    <Email>snehabk468@gmail.com</Email>
    <CourseEnroll>Computing</CourseEnroll>
    <RegDate>2019-12-29T00:00:00+05:45</RegDate>
  </Student>
  <Student>
    <ID>3</ID>
    <Name>Laxmi Poudel</Name>
    <Address>Leknath</Address>
    <Contact>9845076587</Contact>
    <Email>laxmi321@gmail.com</Email>
    <CourseEnroll>Multimedia</CourseEnroll>
    <RegDate>2019-12-30T00:00:00+05:45</RegDate>
  </Student>
  <Student>
    <ID>2</ID>
    <Name>Asmita GC</Name>
    <Address>Parsyang</Address>
    <Contact>9856075689</Contact>
    <Email>asmita123@gmail.com</Email>
    <CourseEnroll>Networking</CourseEnroll>
    <RegDate>2019-12-31T00:00:00+05:45</RegDate>
  </Student>
```

```
<Student>
  <ID>1</ID>
  <Name>Divya Bhattarai</Name>
  <Address>Pardi Birauta</Address>
  <Contact>9806590061</Contact>
  <Email>Ibird0343@gmail.com</Email>
  <CourseEnroll>Computing</CourseEnroll>
  <RegDate>2020-01-01T00:00:00+05:45</RegDate>
</Student>
<Student>
  <ID>20</ID>
  <Name>Jiwan Kumar Bhattarai</Name>
  <Address>Shangrila Margh</Address>
  <Contact>9846076704</Contact>
  <Email>jiwankumar143@gmail.com</Email>
  <CourseEnroll>Computing</CourseEnroll>
  <RegDate>1/1/2020</RegDate>
</Student>
</NewDataSet>
```