Marking Scheme

Informatics College Pokhara

informatics
college pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: **Saugat Timisina**
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	missing controls in the interface
Manual data entry or import from csv	not properly saved or imported data
Data Validation	Only basic 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.

Marking Scheme

Application architecture & description of the classes ad methods sued	very poorly explained.
Flow chart, algoriathms and data sctructures used	average work with very limited explanation and missing diagramatic representation.
Reflective essay	Very poorly written

C. Programming Style

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

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

Overall Comment:

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

OK

LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CC6004NP Application Development

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2019-20 Autumn

Name: Saugat Timilsina

College ID: NP04CP4A170057

University ID: 17030528

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Contents

List of Figure

List of Table

1. Introduction

This system designed is Student Information System to keep and manage records of Student in any institution. This system can be used to store, manage records of students in any institution or also be used to study and research on the specified domain by other scholars on the similar field. This information system has all the features and functions that is required to keep record of regular daily activities. Users/Admin of this system can enrol student by entering name, address, email, contact and programme to be enrolled in. Moreover, it can also present weekly tabular report, total graphical report and all data stored sorted by Name or Date.

2. System Manual

This manual can be used to operate this Student Information System. This system can be used by anyone by following specified set of actions in this manual.

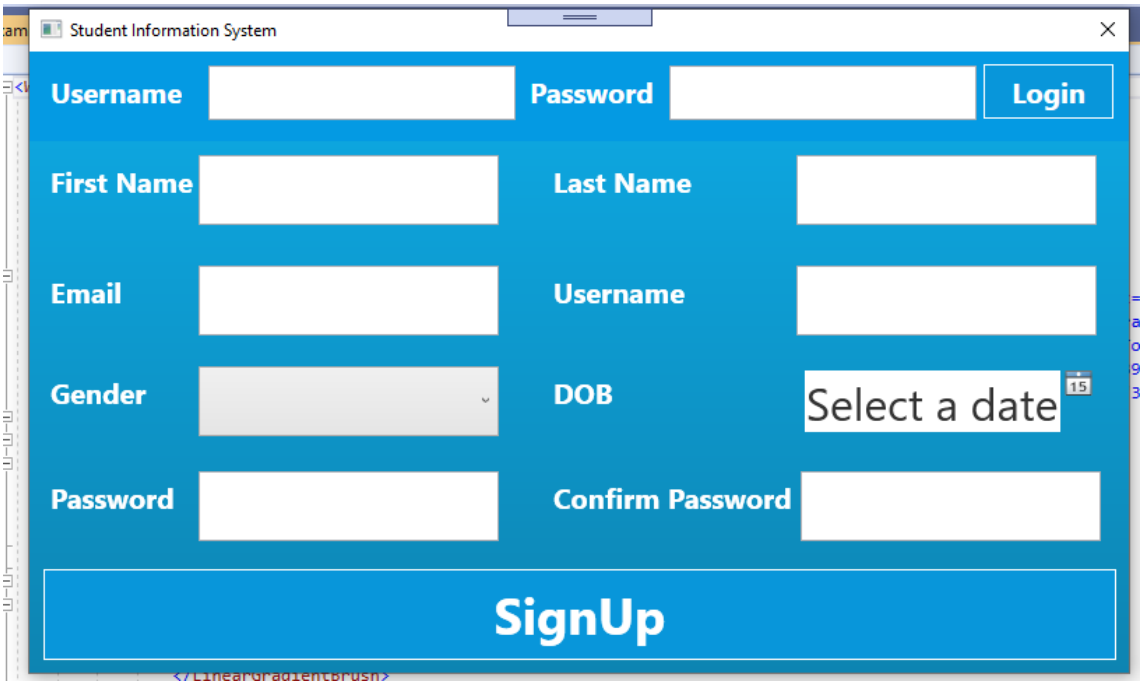


Figure 1: Main Login/Sign-Up Window

Above figure is the main/landing page of our Student Information System. Use have to enter their username and password in order to proceed to the main dashboard where users can be enrolled, searched or to view various reports. The username for this panel is 'admin' and password is 'pass'. User won't be able to proceed to another panel without logging in with these credentials.

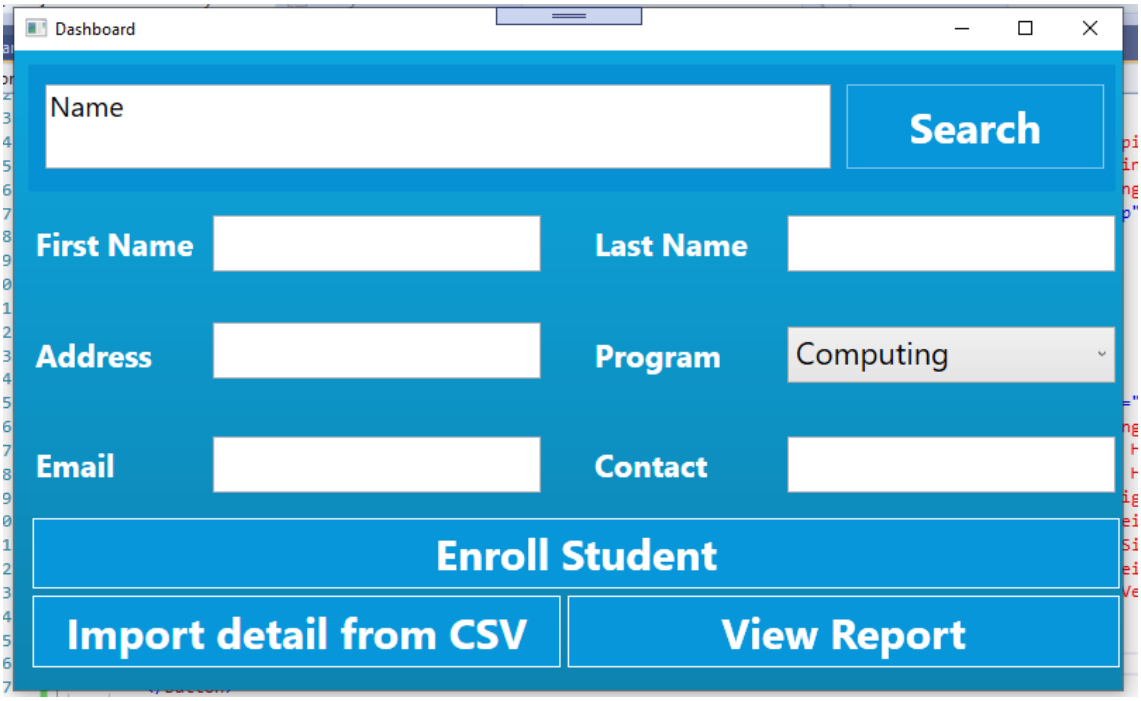


Figure 2: Dashboard after Login

Above Figure 2 is the main dashboard of our Student Information System. After entering the login credentials, the user will be redirected to this panel and user will be able to enroll, search students. Users will also be able to import bulk data using CSV and view Report right from this panel.

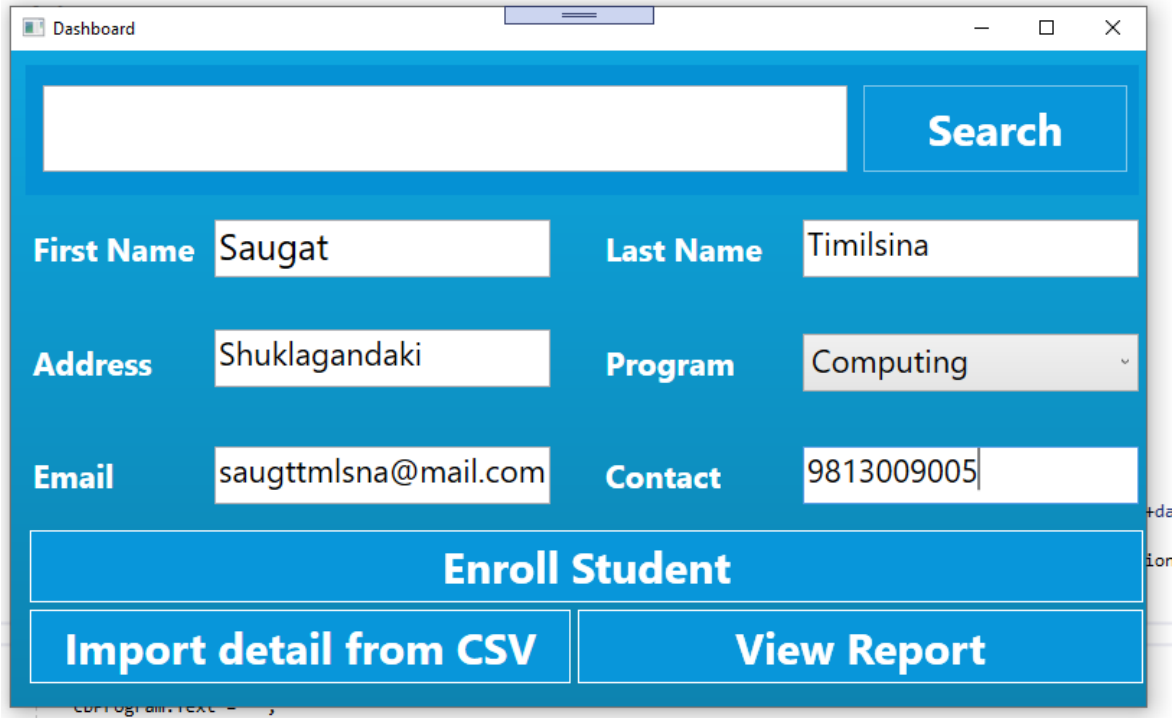


Figure 3: Filling value for enrolling student

Above Figure 3 is a process for enrolling students. Users can enroll students by entering their First Name, Last Name, Address, Program, Email & Contact Number. Users should first fill these details and click the enroll button for enrolling students.

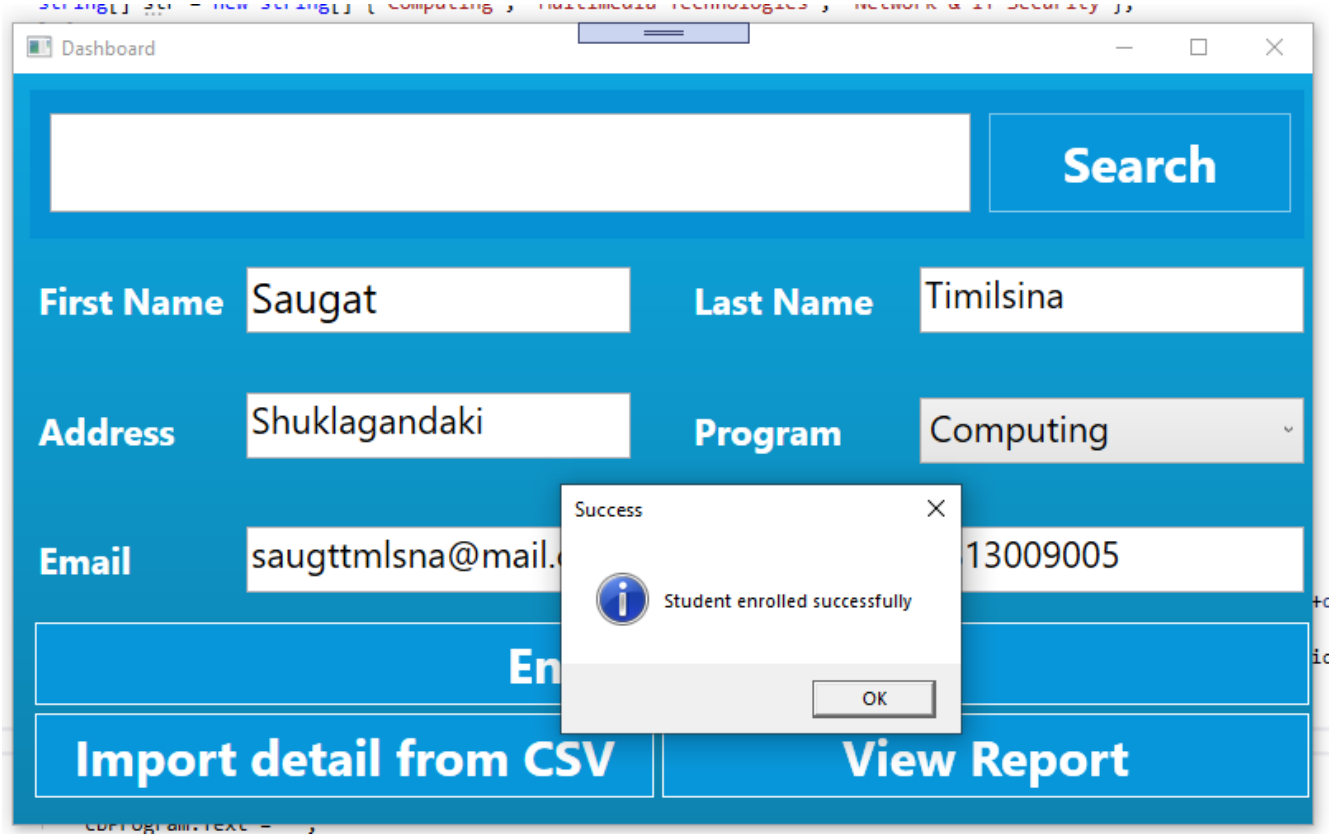


Figure 4: Enrolling Student Successful message

Above Figure 4 is a picture showing dialog box for student being enrolled successfully. This dialog box appears after the user clicks enroll button after filling student details.

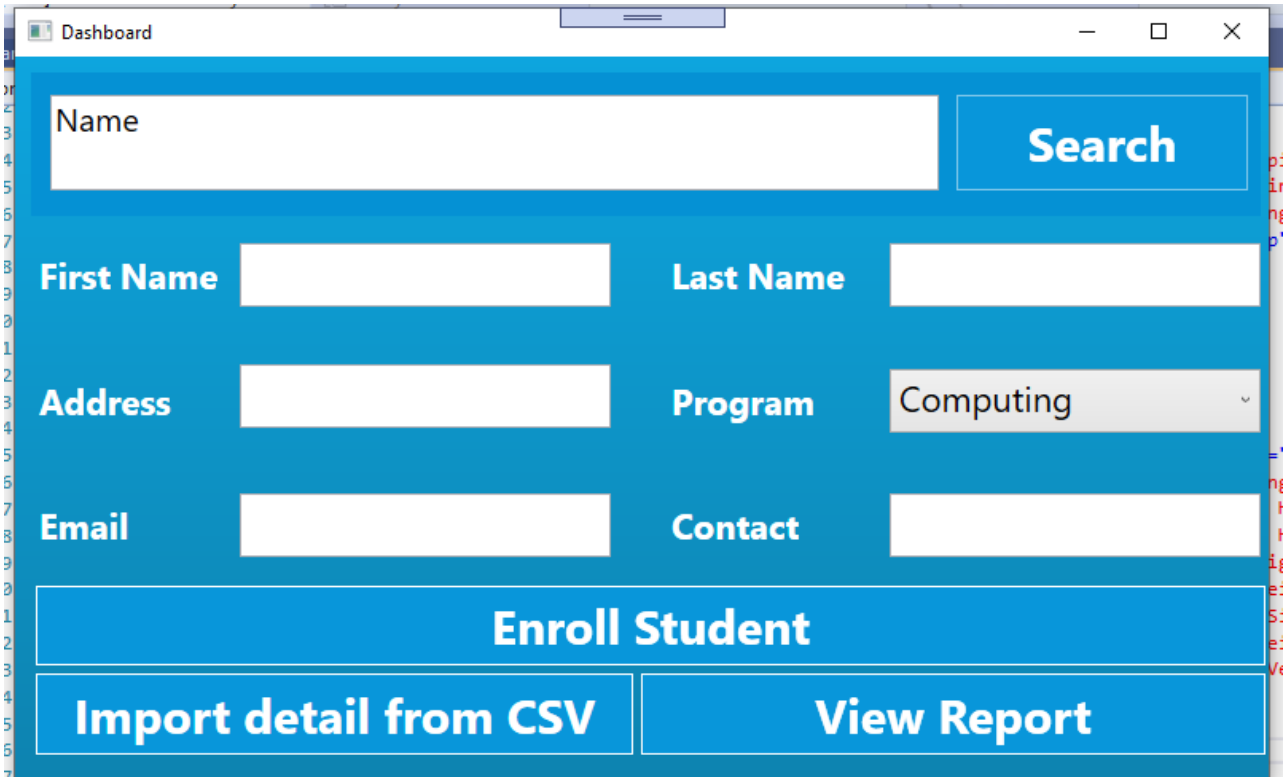


Figure 5: After Enrollment

Above Figure 5 shows how the dashboard looks line after the student is enrolled successfully. There won't be any changes to the dashboard as like the user has just logged-in.

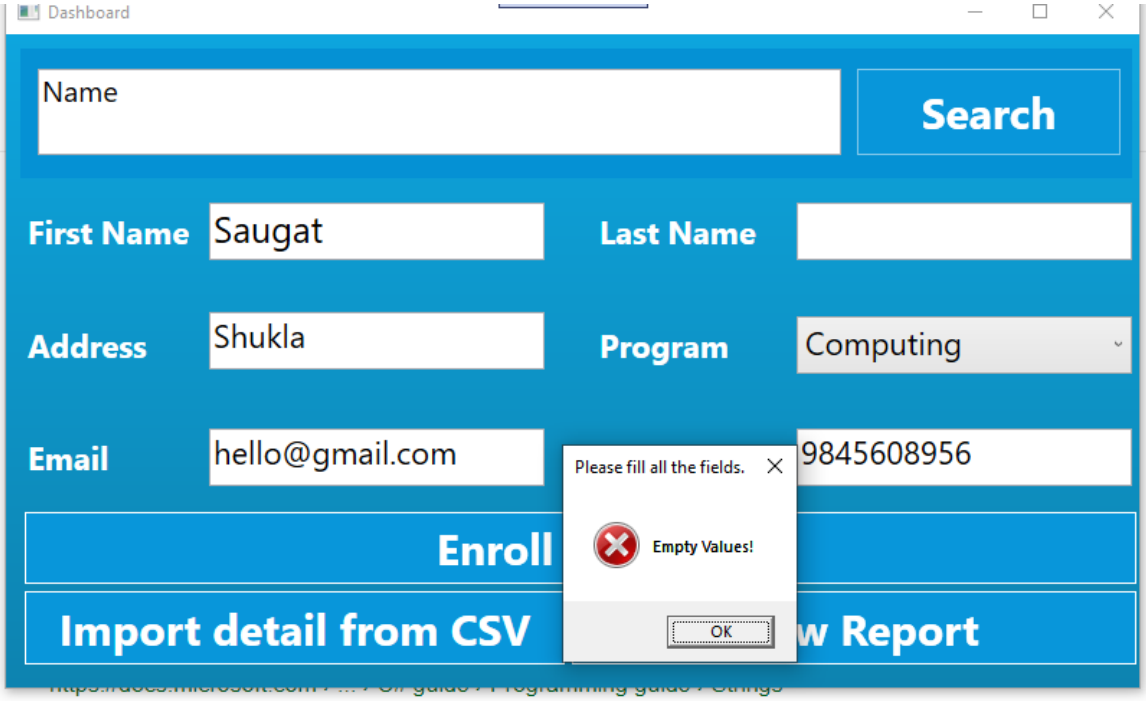


Figure 6: Error displaying Empty value in student information field

If the user tries to enroll a student with blank fields, program will display an error message saying "Empty Values! Please fill all the fields".

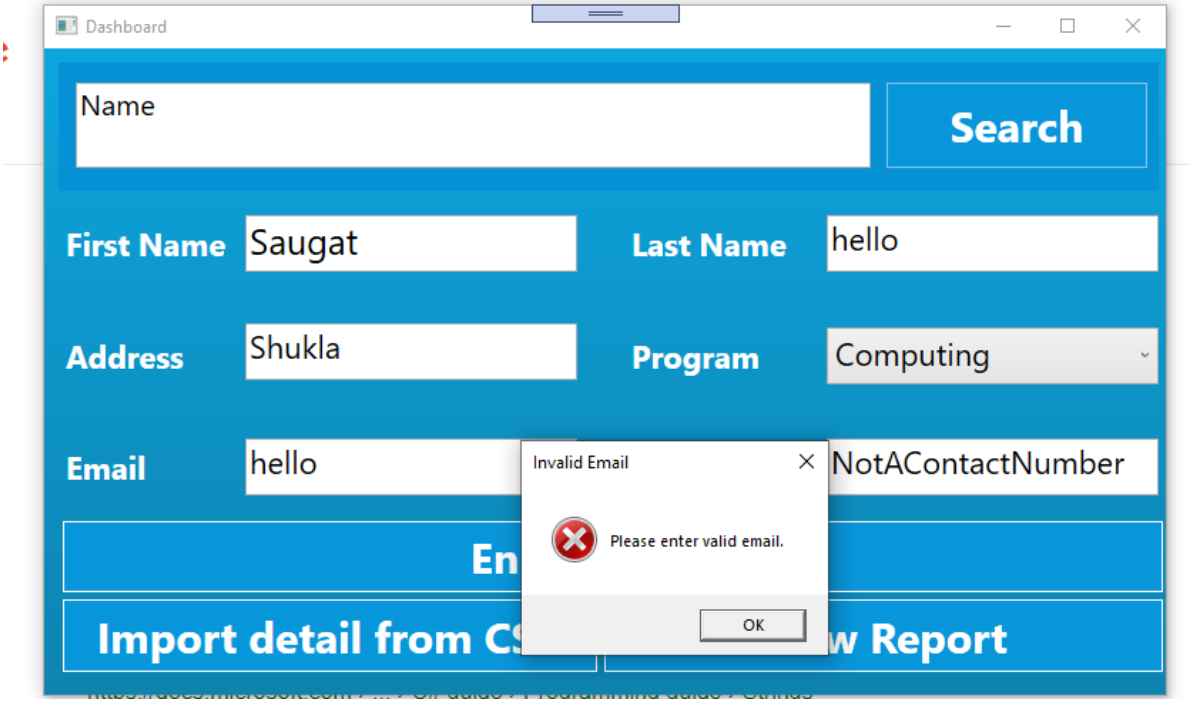


Figure 7: Error displaying invalid email

If the user tries to enroll student with an invalid email, this program will display an error message saying "Invalid Email! Please enter valid email."

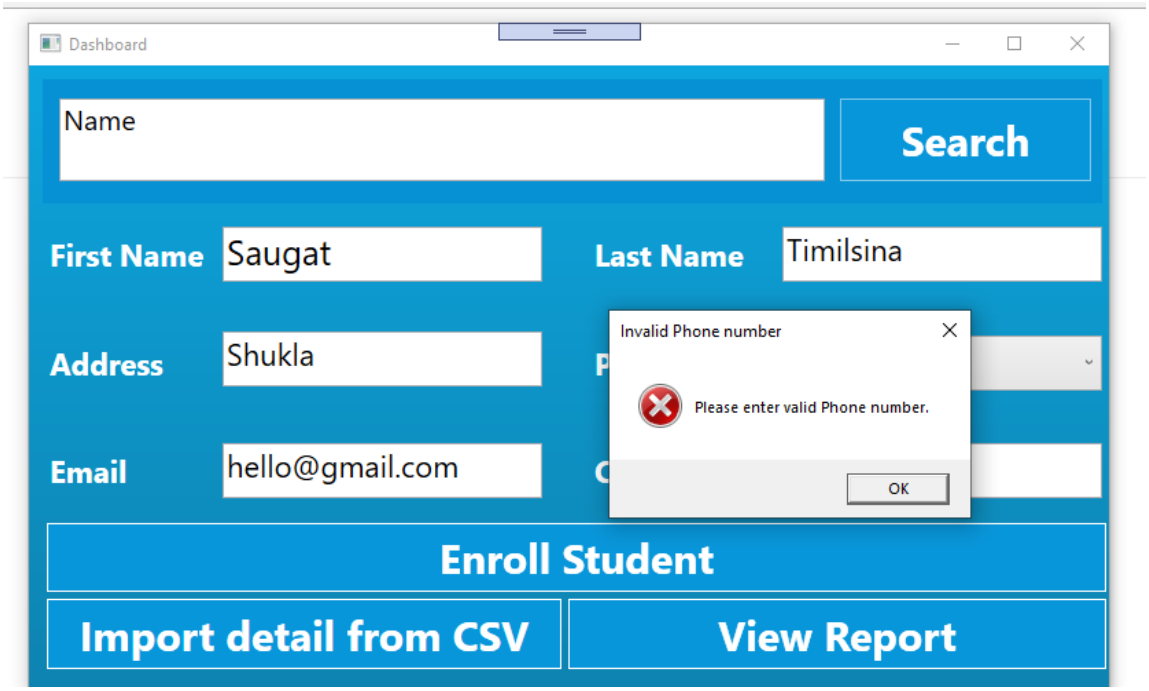


Figure 8: Error displaying invalid Phone number

If the user tries to enroll a student with invalid phone number, this program will display an error message saying "Invalid phone number! Please enter valid Phone number."

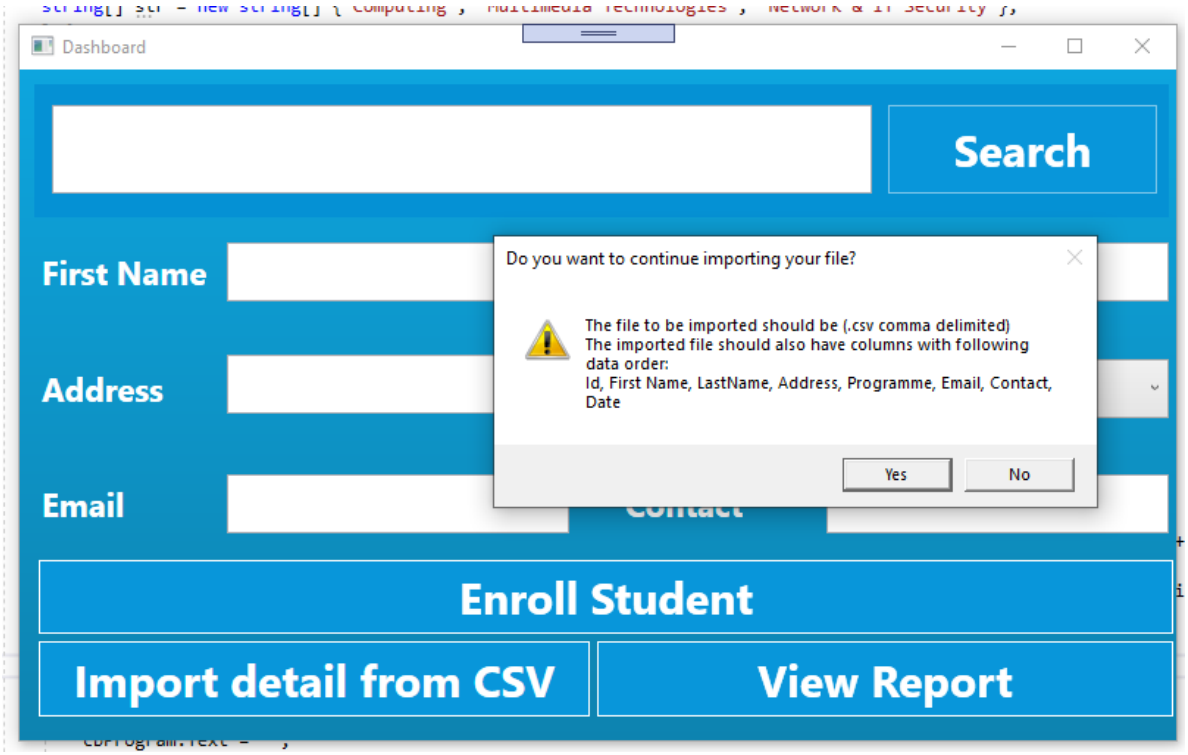


Figure 9: Instructions to Import CSV file

Above figure shows the first stage of importing CSV file. When the user clicks Import detail from CSV button, an instructional dialog box is presented with the formatting of data and file type. If the user clicks No button, the process will be cancelled.

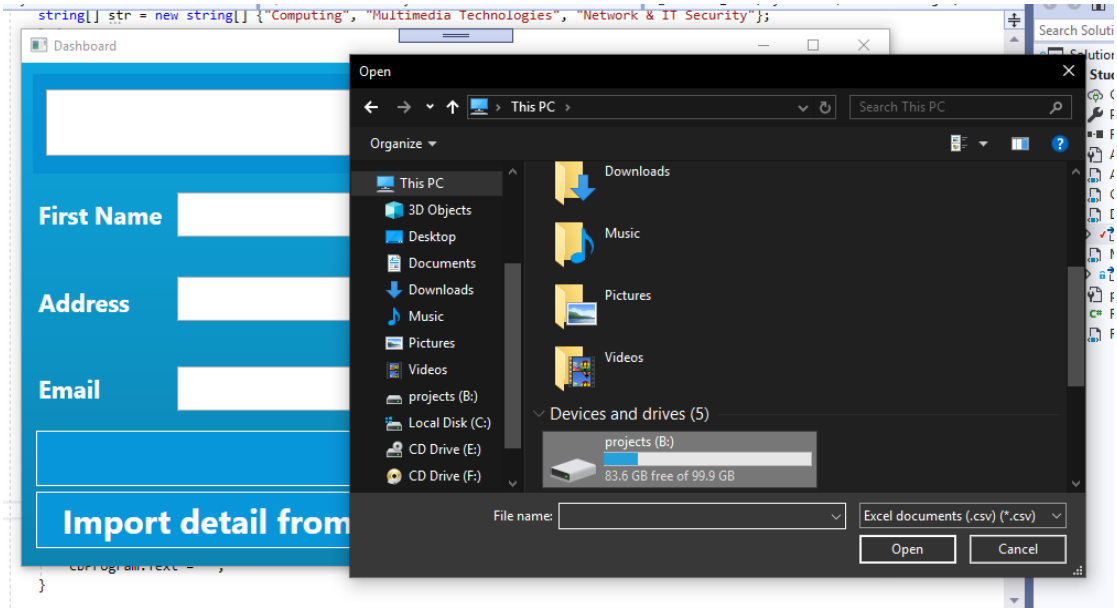


Figure 10: File Chooser Dialog box for importing CSV file

Above figure is a process to import CSV file. After the user clicks Yes to the instructional dialog box, User is presented with a file chooser dialog box where user will be able to choose only CSV file type. When the user clicks open button after choosing the CSV file, that file chooser dialog box will close and program will start importing data from CSV file.

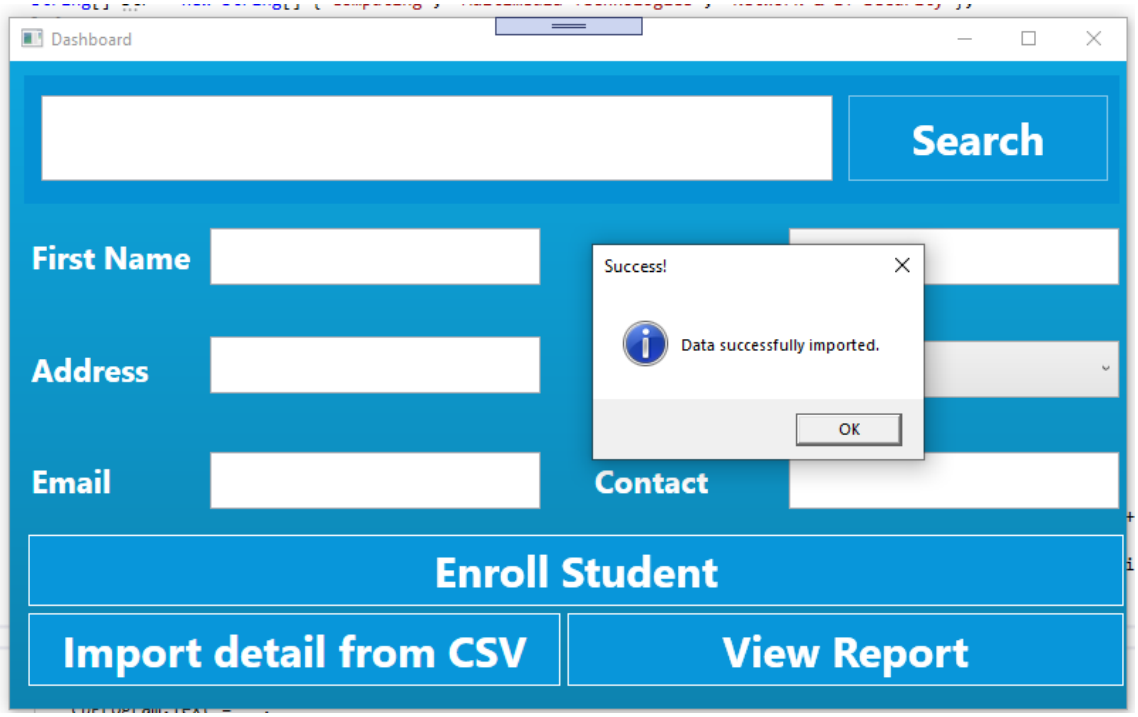


Figure 11: Import Successful

In the above figure, after user chooses files and program begins importing data, it will take a while depending upon the number of rows to completely import all data. After the data is successfully imported, program will show a successful instructional dialog box.

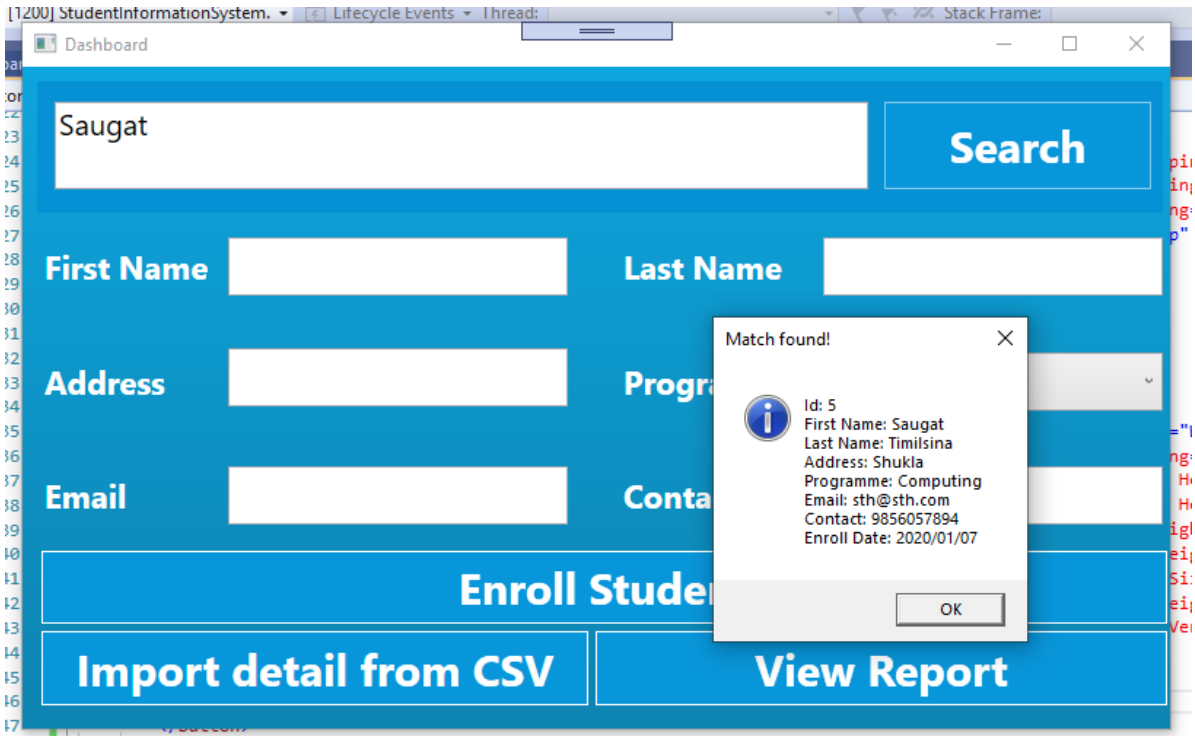


Figure 12: Searching value that exists

Above Figure shows an instructional dialog box with all of the student details. This dialog will appear only if a user enters first name of the student that has been enrolled.

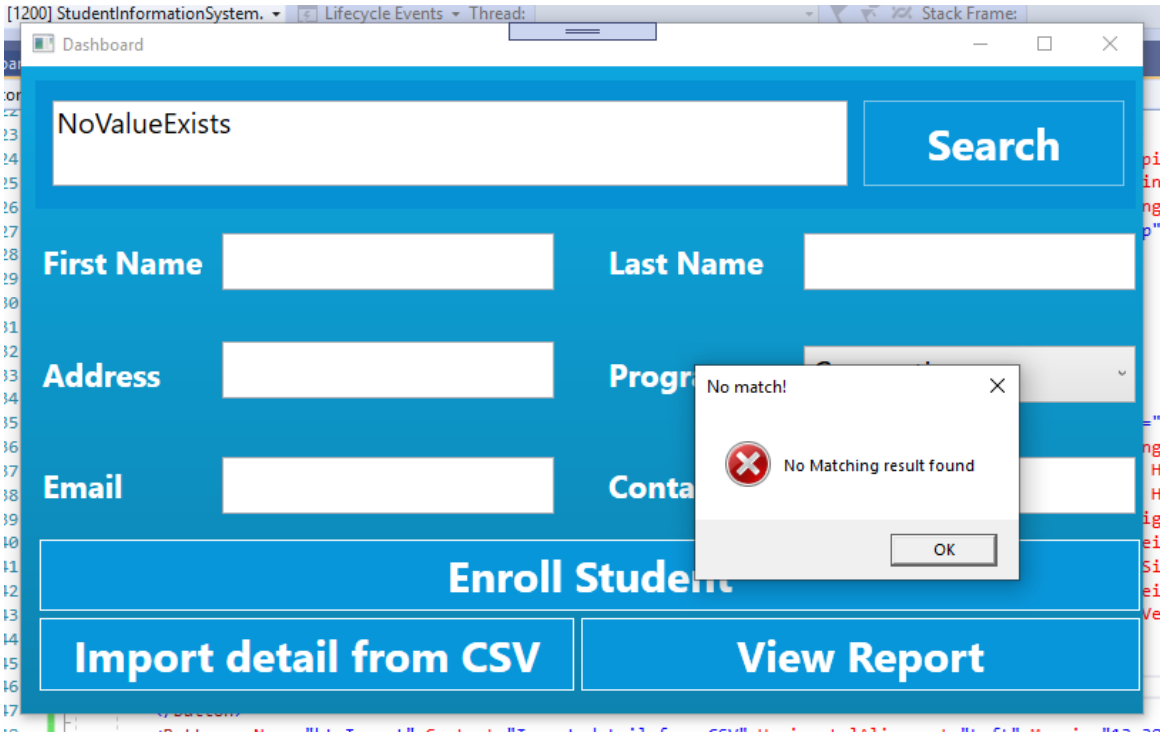


Figure 13: Searching value that doesn't exist

Above Figure shows an error dialog box displaying "No Matching result found". It is displayed when there is no matching first name of the saved data.

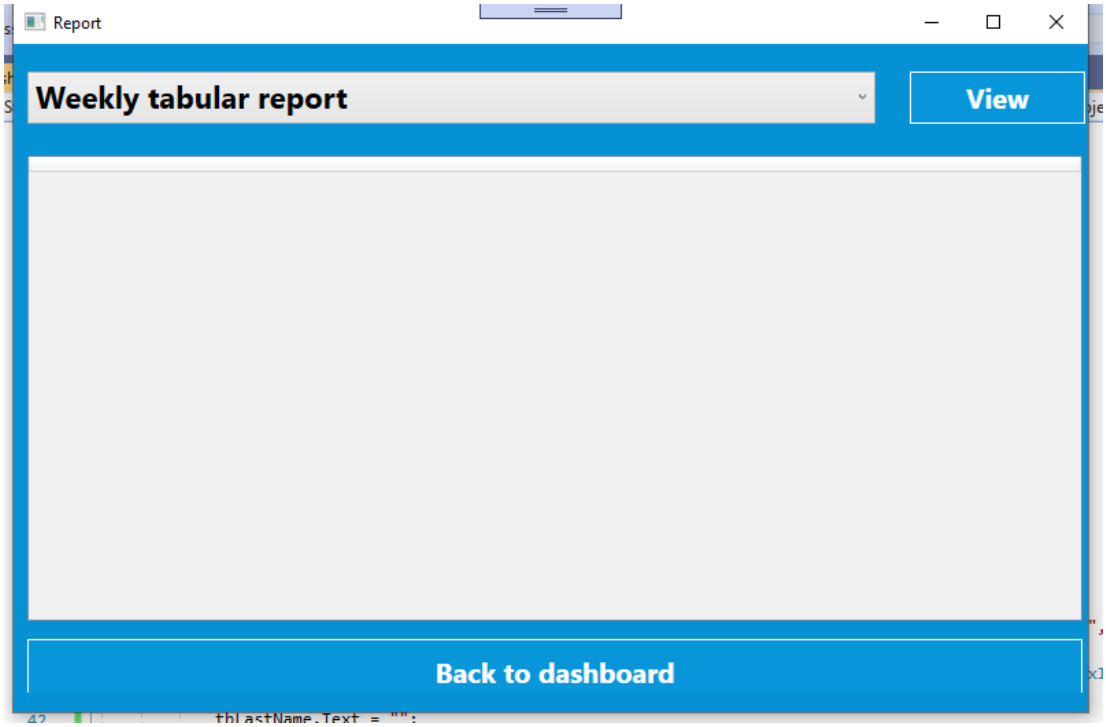


Figure 14: Report Panel

Above figure shows Report Panel where users can view various reports. Users will be redirected to this panel when they click "View Report" button.

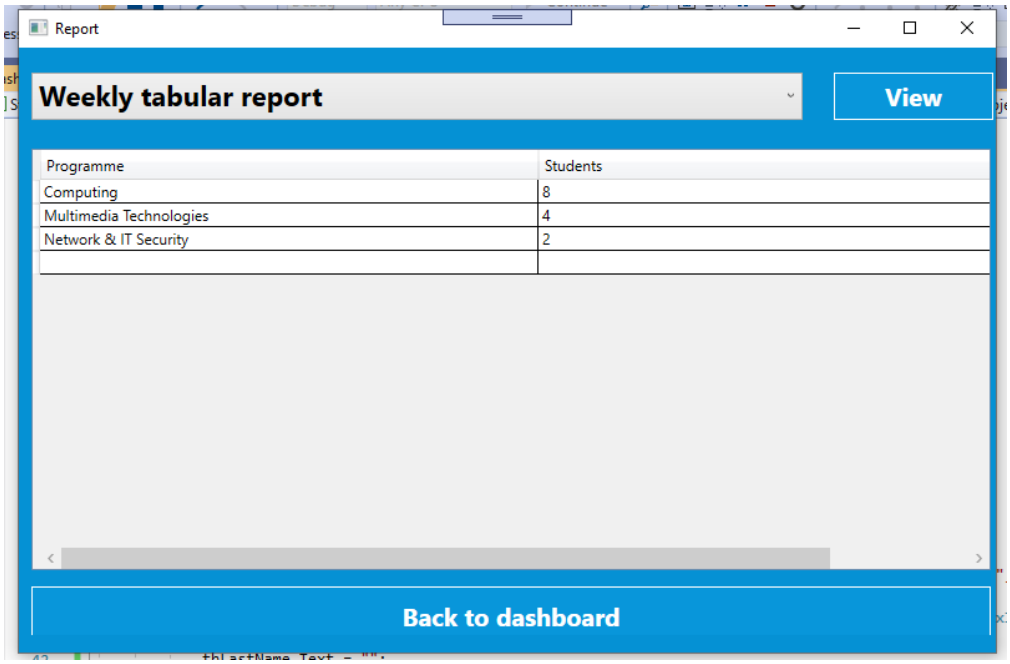


Figure 15: Weekly Report

Above figure shows a weekly tabular report of enrolled students. This report is shown when user selects "Weekly tabular report" from drop-down menu and clicks the "View" button.

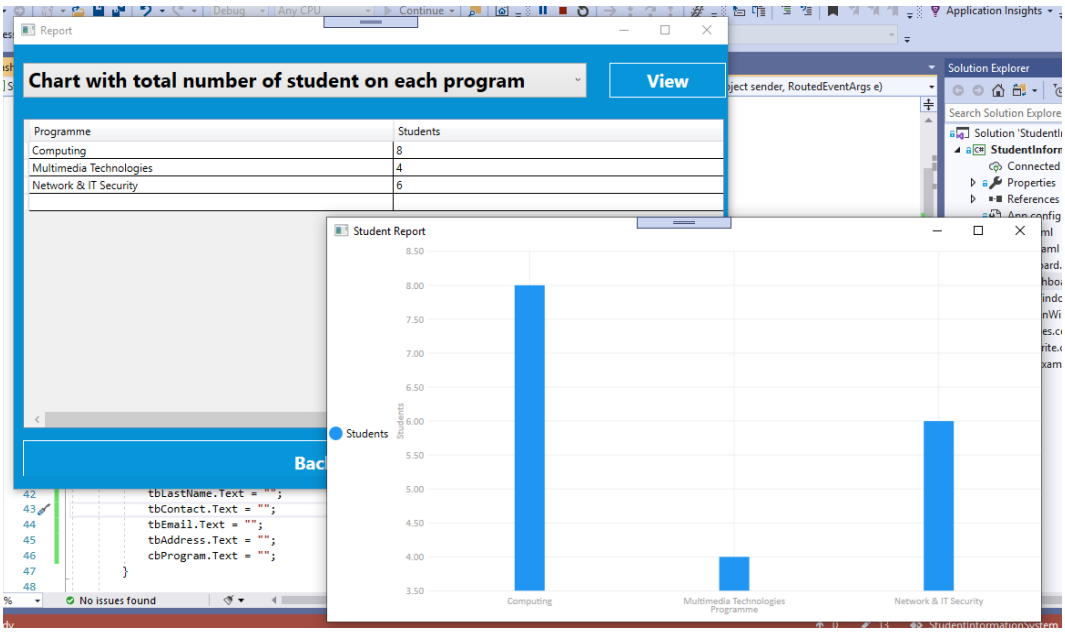
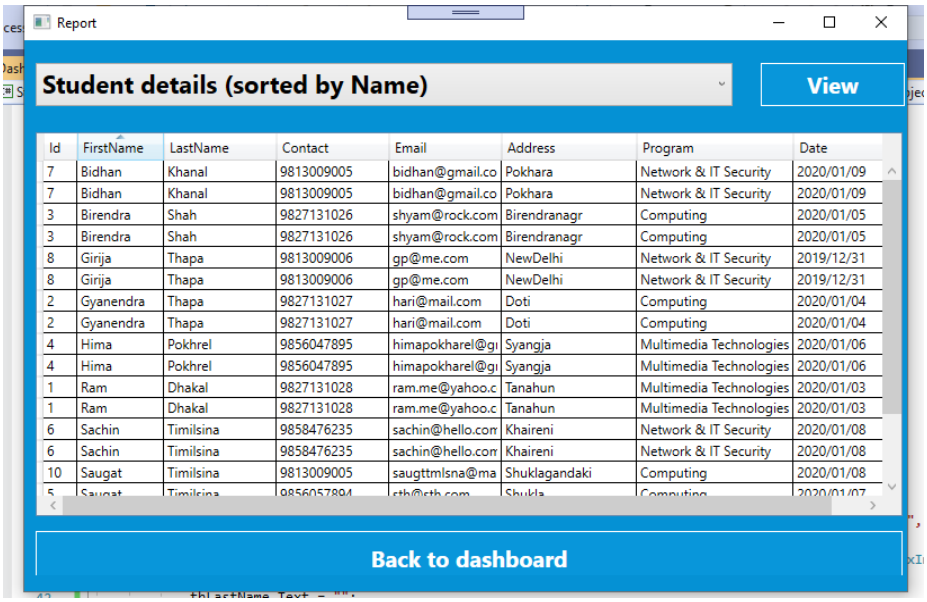


Figure 16: Graphical report of total students

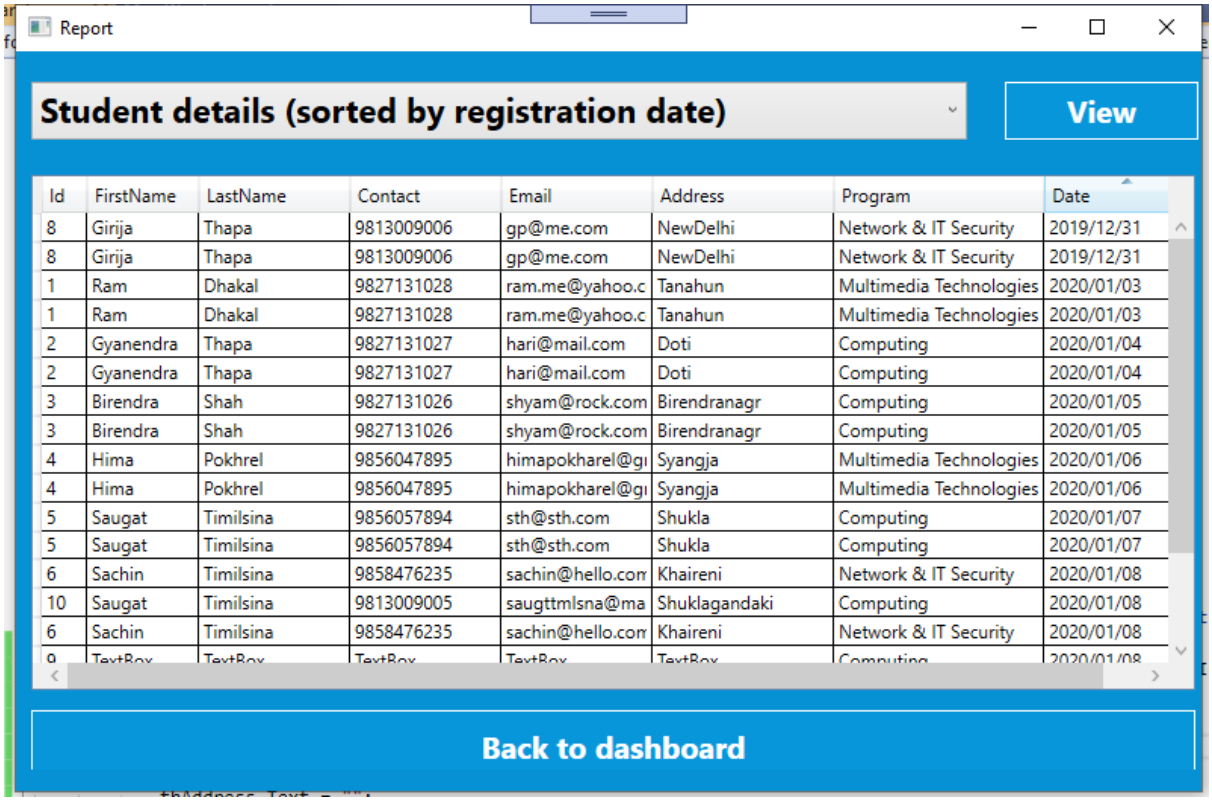
Above figure shows graphical report of total students. This report is shown when user clicks "Chat with total number of student on each program" and clicks the "View" button. In this report, user is presented with total number of students on each program and also in bar graph.



Id	FirstName	LastName	Contact	Email	Address	Program	Date
7	Bidhan	Khanal	9813009005	bidhan@gmail.co	Pokhara	Network & IT Security	2020/01/09
7	Bidhan	Khanal	9813009005	bidhan@gmail.co	Pokhara	Network & IT Security	2020/01/09
3	Birendra	Shah	9827131026	shyam@rock.com	Birendranagr	Computing	2020/01/05
3	Birendra	Shah	9827131026	shyam@rock.com	Birendranagr	Computing	2020/01/05
8	Girija	Thapa	9813009006	gp@me.com	NewDelhi	Network & IT Security	2019/12/31
8	Girija	Thapa	9813009006	gp@me.com	NewDelhi	Network & IT Security	2019/12/31
2	Gyanendra	Thapa	9827131027	hari@mail.com	Doti	Computing	2020/01/04
2	Gyanendra	Thapa	9827131027	hari@mail.com	Doti	Computing	2020/01/04
4	Hima	Pokhrel	9856047895	himapokharel@gı	Syangja	Multimedia Technologies	2020/01/06
4	Hima	Pokhrel	9856047895	himapokharel@gı	Syangja	Multimedia Technologies	2020/01/06
1	Ram	Dhakal	9827131028	ram.me@yahoo.c	Tanahun	Multimedia Technologies	2020/01/03
1	Ram	Dhakal	9827131028	ram.me@yahoo.c	Tanahun	Multimedia Technologies	2020/01/03
6	Sachin	Timilsina	9858476235	sachin@hello.com	Khaireni	Network & IT Security	2020/01/08
6	Sachin	Timilsina	9858476235	sachin@hello.com	Khaireni	Network & IT Security	2020/01/08
10	Saugat	Timilsina	9813009005	saugttmlsna@ma	Shuklagandaki	Computing	2020/01/08

Figure 17: All student details (Sorted by Name)

Above figure all report of total students Sorted by First Name. This report is shown when user clicks "Student details (sorted by Name)" and clicks the "View" button. In this report, user is presented with all students sorted by their "First Name".



Id	FirstName	LastName	Contact	Email	Address	Program	Date
8	Girija	Thapa	9813009006	gp@me.com	NewDelhi	Network & IT Security	2019/12/31
8	Girija	Thapa	9813009006	gp@me.com	NewDelhi	Network & IT Security	2019/12/31
1	Ram	Dhakal	9827131028	ram.me@yahoo.c	Tanahun	Multimedia Technologies	2020/01/03
1	Ram	Dhakal	9827131028	ram.me@yahoo.c	Tanahun	Multimedia Technologies	2020/01/03
2	Gyanendra	Thapa	9827131027	hari@mail.com	Doti	Computing	2020/01/04
2	Gyanendra	Thapa	9827131027	hari@mail.com	Doti	Computing	2020/01/04
3	Birendra	Shah	9827131026	shyam@rock.com	Birendranagr	Computing	2020/01/05
3	Birendra	Shah	9827131026	shyam@rock.com	Birendranagr	Computing	2020/01/05
4	Hima	Pokhrel	9856047895	himapokharel@gı	Syangja	Multimedia Technologies	2020/01/06
4	Hima	Pokhrel	9856047895	himapokharel@gı	Syangja	Multimedia Technologies	2020/01/06
5	Saugat	Timilsina	9856057894	sth@sth.com	Shukla	Computing	2020/01/07
5	Saugat	Timilsina	9856057894	sth@sth.com	Shukla	Computing	2020/01/07
6	Sachin	Timilsina	9858476235	sachin@hello.cor	Khaireni	Network & IT Security	2020/01/08
10	Saugat	Timilsina	9813009005	saugttmlsna@ma	Shuklagandaki	Computing	2020/01/08
6	Sachin	Timilsina	9858476235	sachin@hello.cor	Khaireni	Network & IT Security	2020/01/08
9	TextBox	TextBox	TextBox	TextBox	TextBox	Computing	2020/01/08

Figure 18: All student details (Sorted by Date)

Above figure all report of total students Sorted by Date. This report is shown when user clicks "Student details (sorted by Date)" and clicks the "View" button. In this report, user is presented with all students sorted by their "Date".

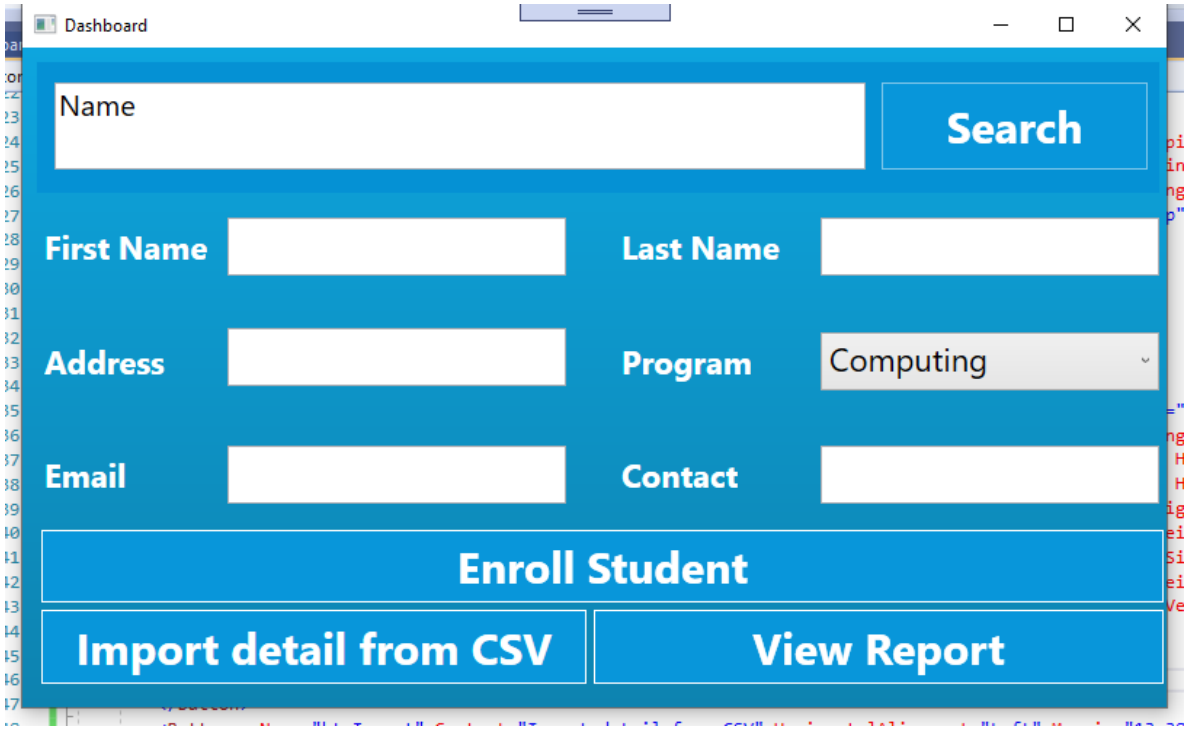


Figure 19: Back to Dashboard

Above figure shows dashboard panel of our Student Information System. This is presented to the user when s/he clicks "Back to Dashboard" button in Report panel.

3. Architecture

Class Name	Source	Remarks
Chart(Window1)	Myself	This class handles methods for showing and processing charts.
Dashboard	Myself	This class handles student enrolment, importing CSV file and redirecting users to Report panel.
MainWindow	Myself	This class is the landing page for users where they will be able to log into the Dashboard.
ReadWrite	Myself	This class contains all methods to Read from and write to file. This class also has methods to handle other various file operations.
Report	Myself	This class contains methods to handle operations regarding showing and processing report.

Table 1: Architecture of the project

While all the class in this program are of my own, I have used Live Chart library for creating and showing graphs to the user.

4. Description of Properties and Methods

a. Chart(Window1)

Properties	Descriptions
SeriesCollection	This property stores Series values.
Labels	This property stores Label values.
Formattes	This property stores value of how the chart should behave/display.

Table 2: Properties of Chart(Window1) Class

Methods	Descriptions
SetGraph	This method sets the value of points to be plotted in the graph
Window1	This is the constructor of our class.

Table 3: Methods of Chart(Window1) Class

b. Dashboard

Properties	Descriptions
readWrite	Object of ReadWrite class
toStore	Stores the user entered student information.
chosenFile	Stores the location of selected file

Table 4: Properties of Dashboard Class

Methods	Descriptions
IsValid	This method checks if a email address is valid. It returns true if it is valid and false if it isn't
IsNumeric	This method checks if a string can be converted to numerical values. It returns true if the method can be changed to numerical value and false if it can't.
btnEnroll_Click	This method is executed whenever the user clicks Enroll Button. This method gets all the value from TextBoxes and passes the value to be enrolled.
btnSearch_Click	This method is executed when the user clicks search button. After getting the value to be searched, this method checks and displays suitable result.

Table 5: Properties of Dashboard Class

c. MainWindow

Properties	Descriptions

-	-

Table 6: Properties of MainWindow Class

Methods	Descriptions
MainWindow	This is the constructor of this class and only sets value of the combobox, initializes all components.
Button_Click	This method is executed whenever the user tries to log in to the system. It pulls values from the textboxes and passes them onto login method
Login	This method handles the login part of our system. It checks if username is "admin" and password is "pass" without inverted commas and redirects users to Dashboard if it is. If the credentials aren't correct, it displays a errpr message
SignUp	This method display a error message box when user clicks SignUp Button.

Table 7: Methods of MainWindow Class

d. ReadWrite

Properties	Descriptions
FILE_NAME	Constant property and stores the filename to store and pull data from.
managedRecords	This is a multi-dimensional array. It is used to store organized data imported from CSV file and dat file.

Table 8: Properties of ReadWrite Class

Methods	Descriptions
Read	This method is called to read data from our .dat file and returns a multi-dimensional array.
getLines	This method readss all lines from our .dat file, stores those line in an array variable and returns that array
getLength	This method returns the number of records in our .dat file.
write	This method writes data passed for enrolling to the .dat file.

Table 9: Methods of ReadWrite Class

e. Report

Properties	Descriptions
returnedRecords	This property stores multi-dimensional array returned from calling ReadWrite Class's read method
computing	This property stores the number of students enrolled in computing subject.
networking	This property stores the number of students enrolled in Networking & IT Security subject
multimedia	This property stores the number of students stored in Multimedia Technologies subject.
str	This property stores options for setting the values in combo box to view report

Table 10: Properties of Report Class

Methods	Descriptions
Report	This is the constructor of our class. It initializes all components and sets the value of Combo Box to view records.

readData	This method calls mwthod of ReadWrite class and stores the returned value.
btnShowReport	This method is executed when user clicks the "View" button to view report. It calls appropriate methods according to user chosen report
weeklyTabularReport	This method is executed when user chooses to view weekly report. This method processed all data in the file and presents enrolment data in last 7 days including current day.
sortDataGrid	This method is called to sort data grid when user chooses to view the data sorted by date or name. It sorts the grid in order of column number and direction sent to it through parameters.
AllTabularReport	This method processes and arranges all the data in a grid view when user chooses to view all records

CalculateAndSetChart	This method processes the data and plots the graph when user chooses to view Graphical Report

Table 11: Methods of Report Class

5. Design

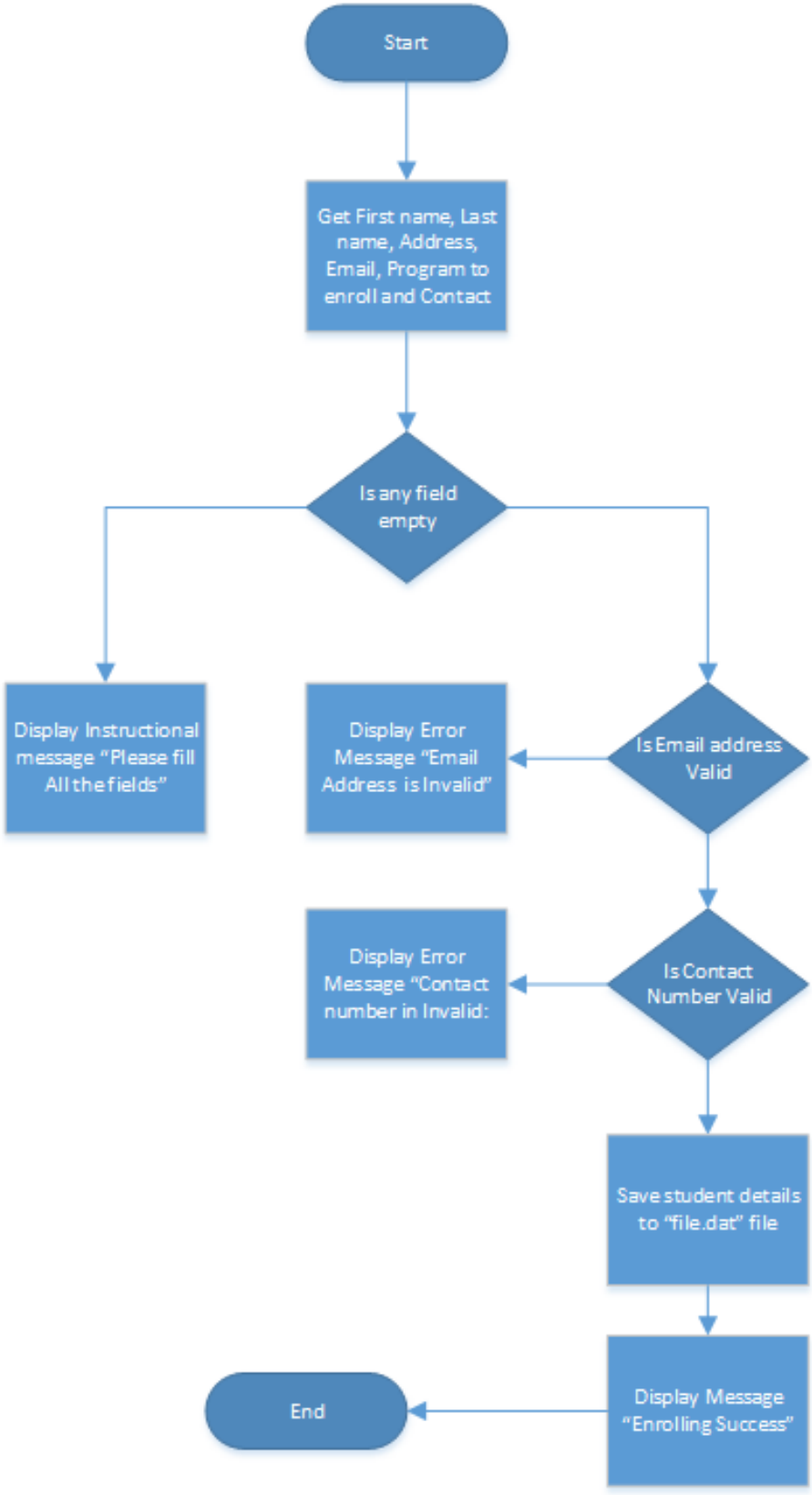


Figure 20: Flowchart for enrolling student

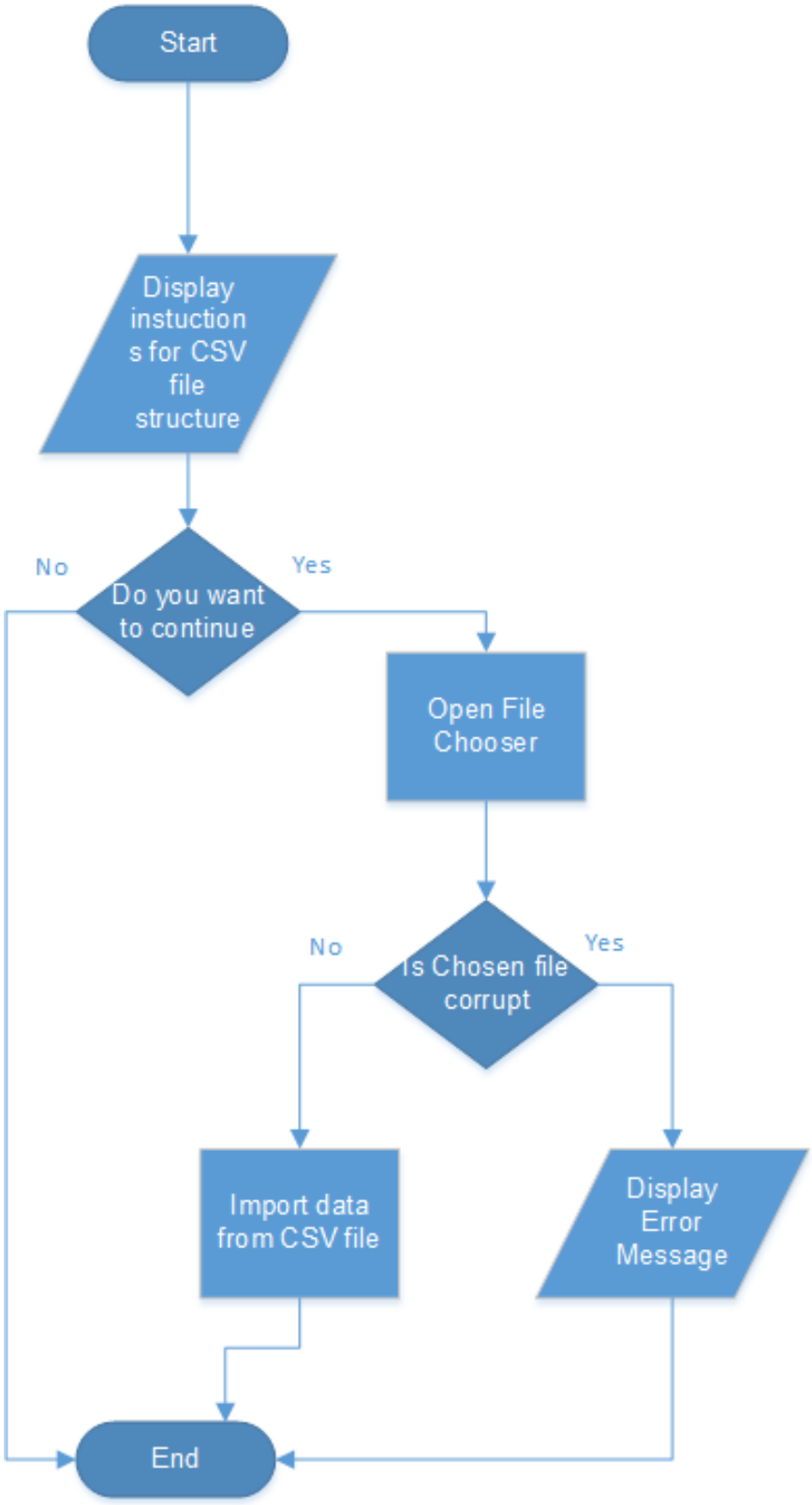


Figure 21: Algorithm for importing data from CSV file

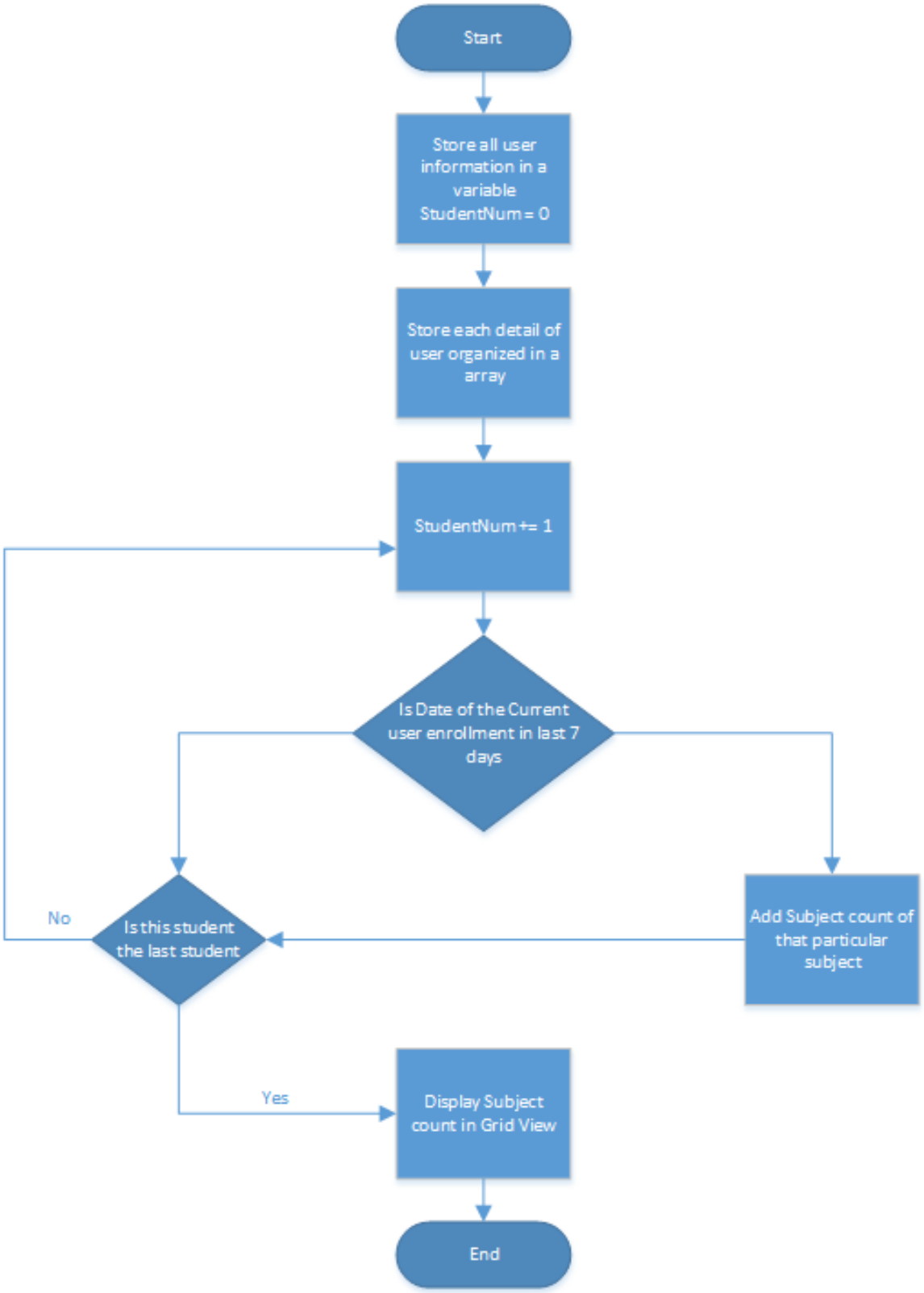


Figure 22: Flowchart for Weekly Report

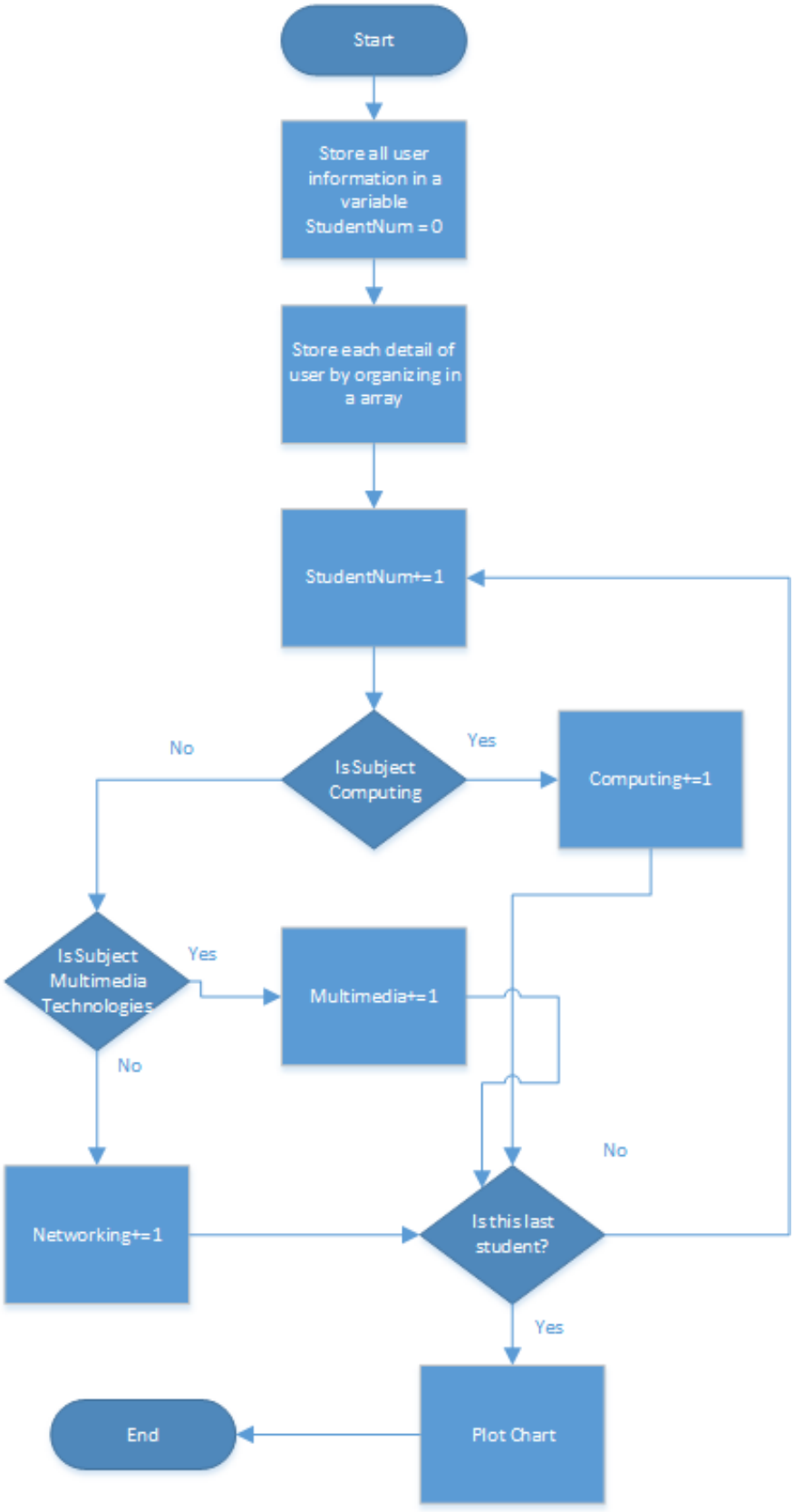


Figure 23: Flowchart for Total student Report

6. Data Structures used

Data structure is a certain way of arranging data in a computer so that it can be used effectively for future uses. In this project, I have used Array as my primary data structure because of its simplicity.

a. Array

An array is a collection of items stored at contiguous memory location which helps to store multiple items of the same type together. In an array, each element can be uniquely identified by their index in the array. Array can be classified into two types: One-dimensional, Multi-dimensional. A one-dimensional array is a list of variables with the same data type. A two-dimensional array is array of arrays having similar data types.

I have used both one-dimensional and multi-dimensional array in this project. One dimensional arrays have been used to store data for combo-box options and two-dimensional array have been used to store separated data while importing data from CSV file and checked while viewing reports. Moreover, I have also used Array Lists which are is an array-based implementation of the List interface (Toolsqa, n.d.). I used arrays on this project because arrays consume less memory than other data structures, they help tidy thing up and are less complicated that other data structures to implement.

7. Algorithms used

Algorithm is a process or set of rules to be followed in calculations or other problem-solving operations especially by a computer (Oxford Dictionary, 2015). I have used Quick Sort Algorithm for sorting in this project. I have used it by calling an in-built function of C# (WPF platform) to sort Grid view data.

a. Quick-Sort Algorithm

Quick-Sort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the giver array around the picked pivot. The key process in Quick-Sort is partition. Target of partitions is, given an array and an element x of array as pivot, put x at its correct position in sorted array and put all smaller elements (smaller than x) before x, and put all greater elements (greater than x) after x. All this should be done in linear time (Geeks for Geeks, n.d.). I used this algorithm in this project because it was built in to the C# WPF platform and was less complicated than other algorithms to implement.

8. Reflection of experience

This project was developed as a first Individual coursework of our Application Development Module. In this project, we had to complete building a Student Information System using C# as our main programming language, WPF as platform and Visual Studio as IDE. We were required to develop the full information System with GUI integration which can be navigated and used easily by a non-technical user.

As this was our first time using C#, WPF and Visual studio, it was quiet overwhelming at first. But when we started working and continued the module, it started seeming quite easy probably because of my familiarity with Java and other OOP languages. The concept was same but syntaxes were a bit different. We soon overcame the overwhelming thought of learning new language, platform and IDE after couple of week of study in Classes. The main problem I saw many of us students facing was our computer hardware lacking memory and processing power to run Visual Studio smoothly. Learning wasn't that challenging as was our effort to run visual studio on the bare minimum requirements provided by Microsoft. After a couple of days of research and jumping from various forums to forums, we started being comfortable with our laggy Visual studio experience because of other users also feeling the same situation.

After all these research and learning, my experience with Visual Studio and C# was quite fun. I came to get more knowledge on C#, Visual studio and WPF while working on this coursework. Working on creating report, graphical chart, importing data from CSV file was a new experience for more. So, this coursework has enhanced my knowledge on Computing, Application Development and I am hopeful it will also be helpful in my career ahead.

Bibliography

Geeks for Geeks, n.d. *QuickSort Algorithm.* [Online]
Available at: https://www.geeksforgeeks.org/quick-sort/
[Accessed 1 2019].

Oxford Dictionary, 2015. Algorithm. In: *Oxford Dictionary.* s.l.:Oxford.

Toolsqa, n.d. *Data Structure-Arraylist.* [Online]
Available at: https://www.toolsqa.com/java/data-structures-arraylist/
[Accessed 1 2019].