Marking Scheme

# Informatics College Pokhara

informatics
college pokhara

## Application Development

## CS6004NI

## Course Work 1

**Submitted By:** **Niranjan Gurung**
**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** | 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** | 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** | Very poorly written |

**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 componen and require to add comments on required area.

OK

LONDON METROPOLITAN UNIVERSITY


# Application Development

# CS6004NP

## Coursework 1

**Submitted by:**

**Student Name:** Niranjan Gurung

**Student ID**: 17030728

**Group**: C1

**Submitted to:**

Application Development

## Abstract

This is an individual coursework of the module “Application Development” for “Student Information System”. This project have to be develop by using Visual studio platform and C# programing language. The coursework was release in the week 5 and it is supposed to be submit in the week 11. The system is design to track all the records of the students.

With the great contribution of “Mr. Ishowr Sapkota”, the coursework was completed within the period.

# Table of Contents

## Table of Figures

# 1. Introduction

In this modern world, technologies are becoming more and more edge cutting. One of them are data handling. We do not have to handle data with traditional way like paper, notebook, drawer, etc. anymore. Therefore, to handle the data of enrolled students in any educational organization this type of digitalized window form is very essential.

Inside this student information system, we gets functions like to import a record from a text file (in .CSV format for bulk data input) and manual inputs details of the students. Details includes Registration Id, Full Name, Address, Contact Number, Course Enrol and Registration Date. Similarly, to generate and displaying two different reports of the students sorted by student's full name and sorted by the registration date. Also, a tabular report and a chart showing total n umber of student on each program. Finally, using the system we can save and retrieve the student enrol status with the student details.

## 1.1 Current Scenario

Nowadays, like all of the Student organization like colleges, schools and Universities uses this type of system to handle their student's data. In our traditional way, there is high chance of data loss and errors. To diagnose those problems and to make the data handling easy system like this very essential. Peoples are researching and wanting to take this type of system in next level.

## 1.2 Description of the System

This system can be use in student's organization to overcome the traditional data handling methods. This system provides data security with the login features and creates the report on single click. This saves the precious time of the user. This system is user friendly and easy to use. No extra knowledge and practise is needed.

# 2. User Manual

The full instruction to run the program along with proper screenshots are as follows:

a) **First, run the Program.**

Click the "**Student_Information_System.sln**" file inside the folder name "course-work-1-Niranjan-26".

**b) Login**

After running the system, first we see login window.

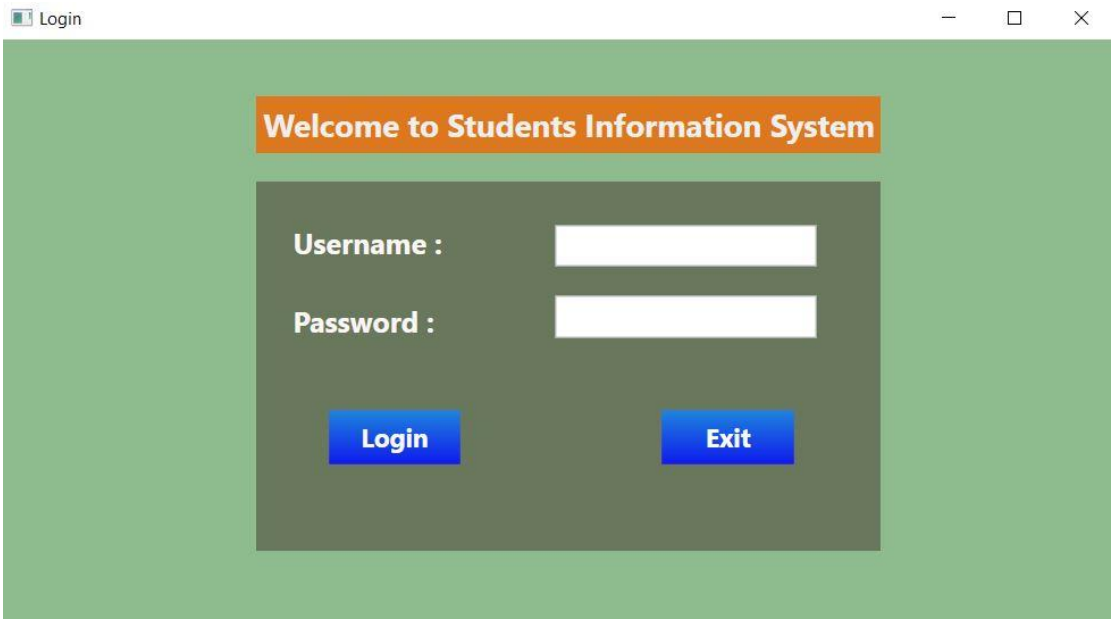



*Figure 1: Login Form*

- The Username and Password for this system is "admin".
- If the fields are empty it shows message and if the user can exit the program if they wants too.

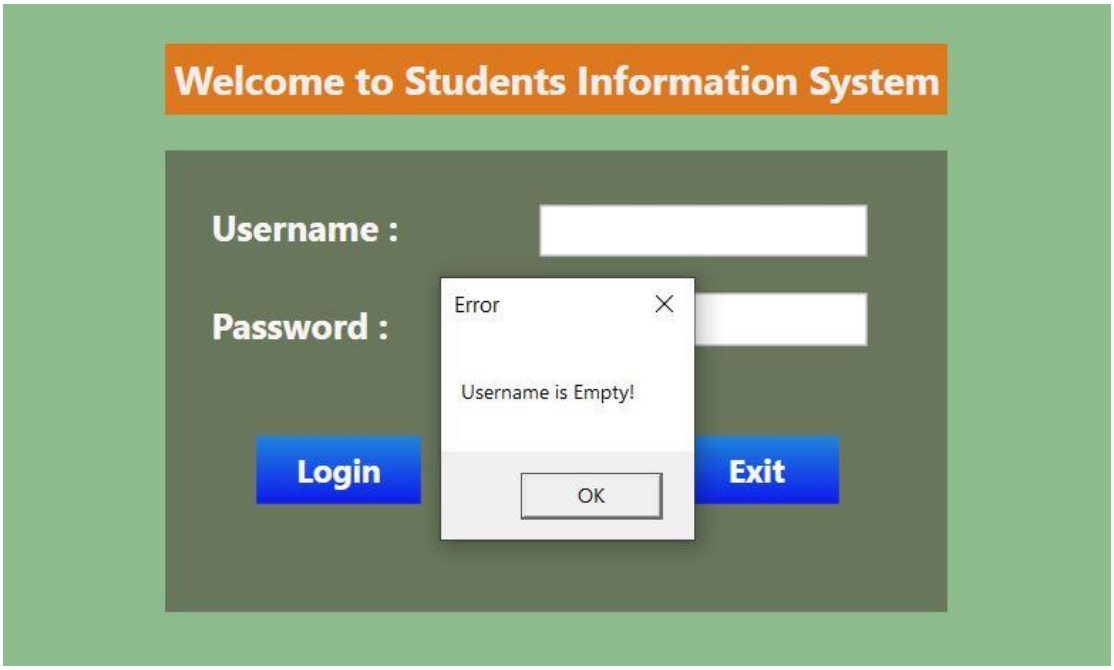



*Figure 2: Username is Empty*

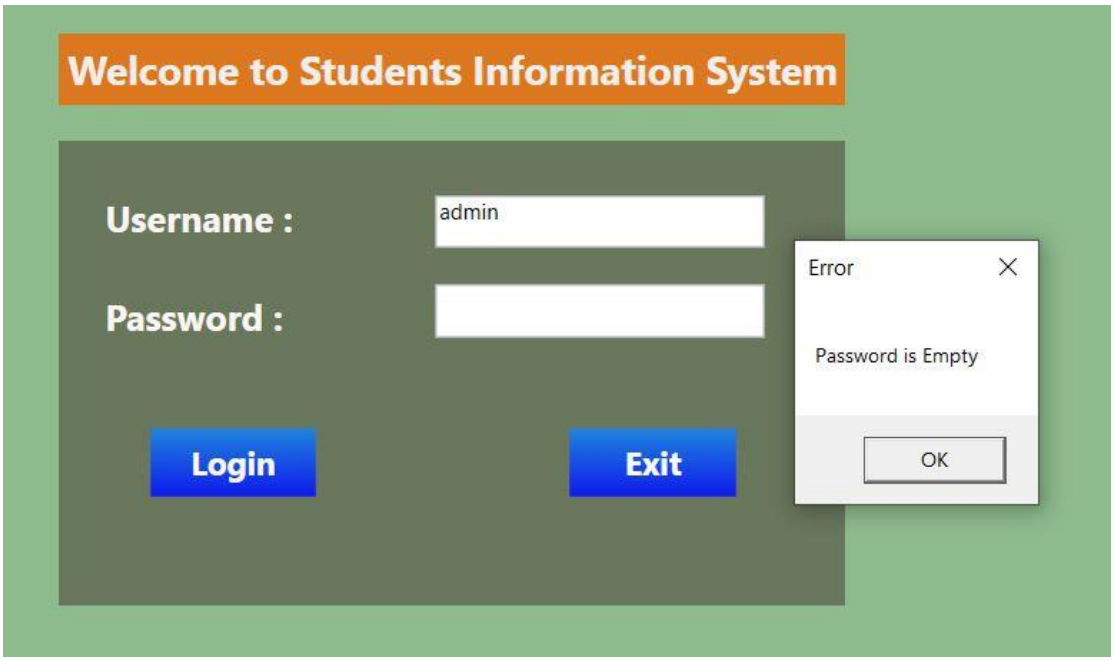



*Figure 3: Password is empty.*

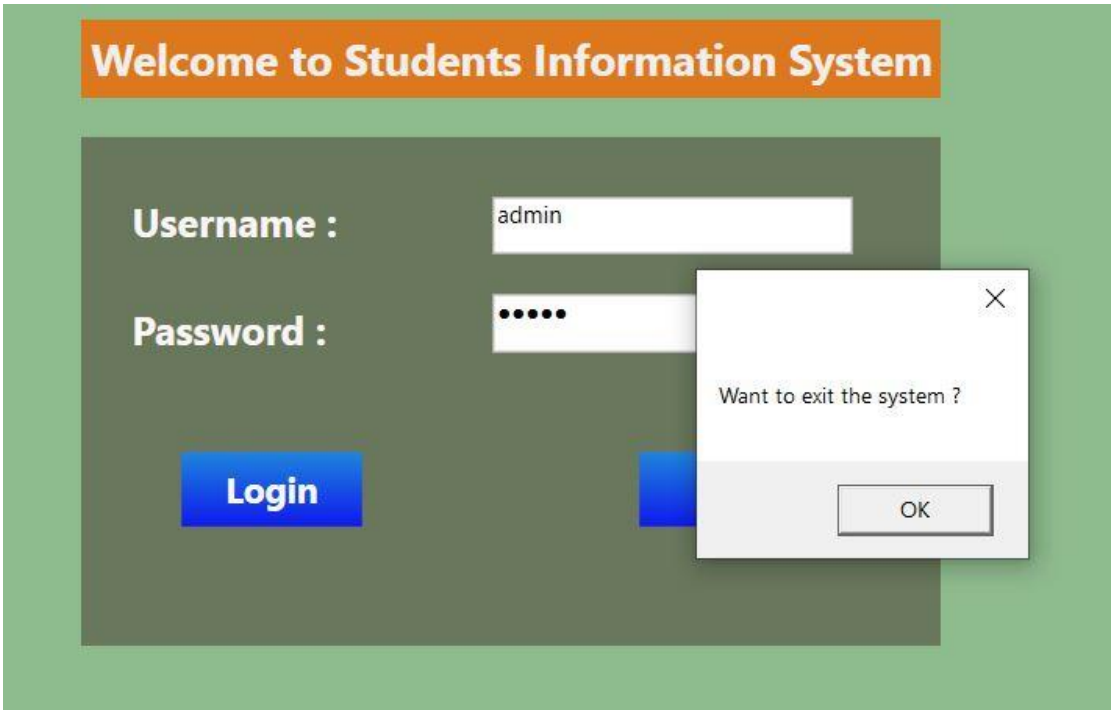



*Figure 4: exit from the Login window*

### c) Main Window

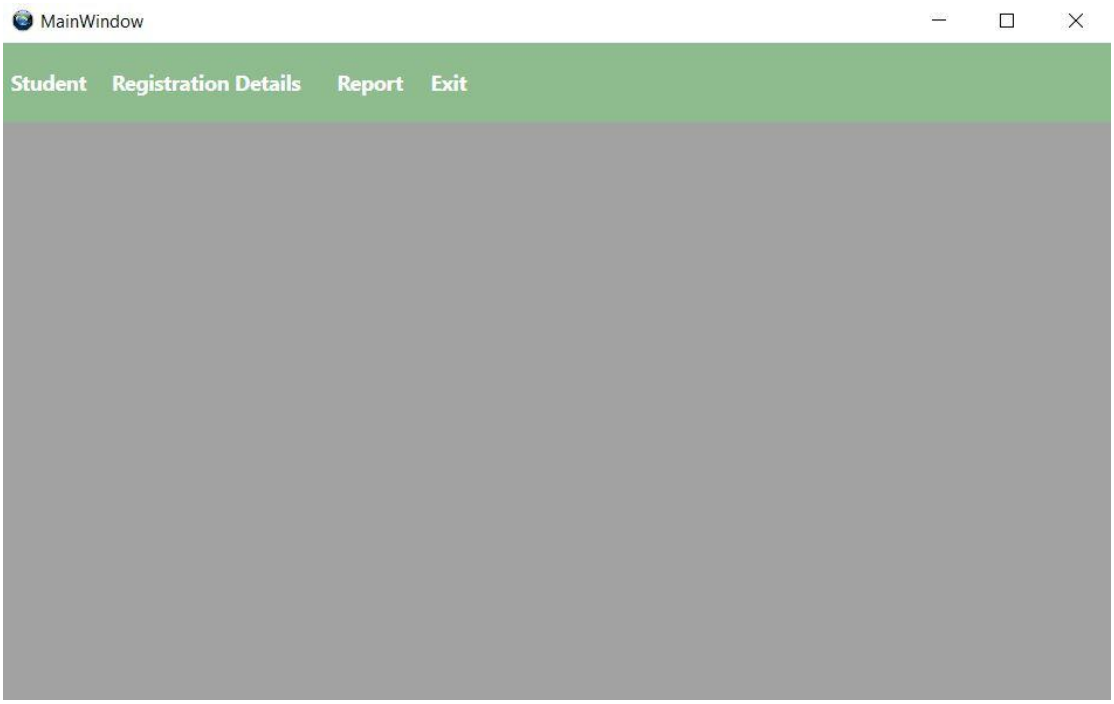



*Figure 5: Main Window*

After successful login, Main window will appears. There are four option on the menu to click. They are:

- **Studen**t

  Here, we have two options to add students and import the student's bulk data in .CSV.

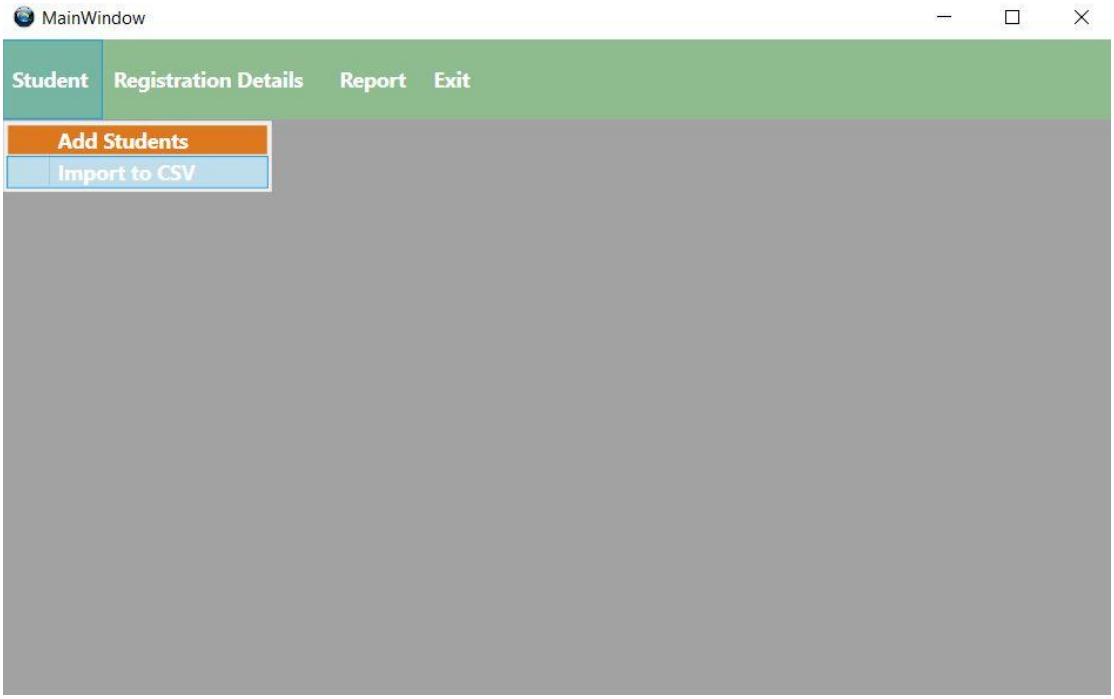



*Figure 6: Add Students option in Student Option*

After clicking this "Add students" option, we get in to this following window, which is registration form.

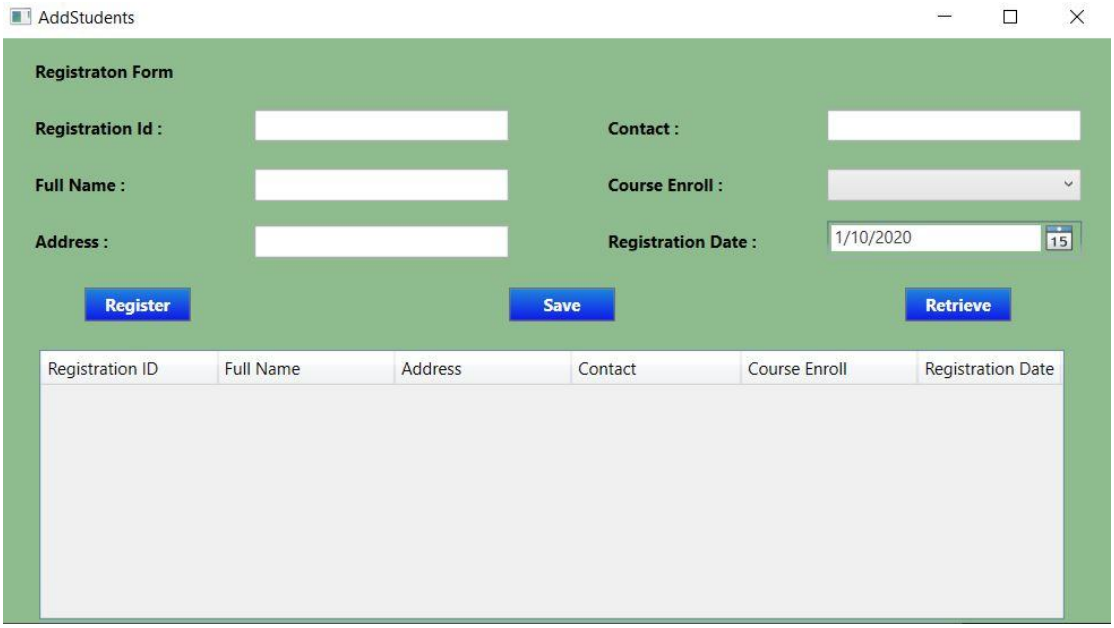



*Figure 7: Registration Form*

If a Fields is register with an empty data. It shows the message.

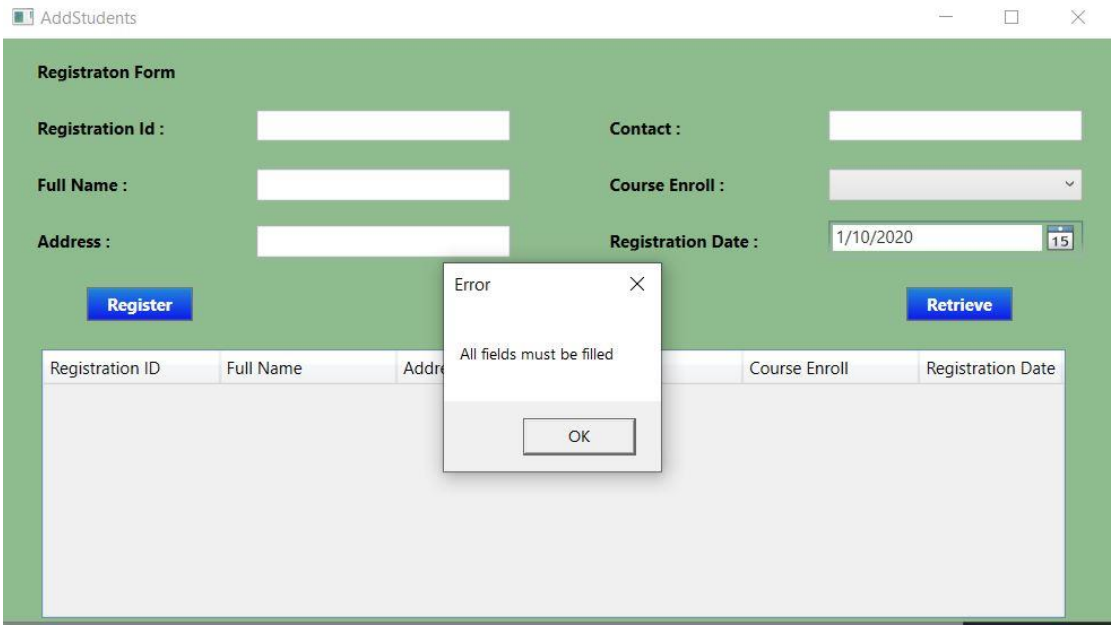



*Figure 8: message shown while there is an empty data in registration form*

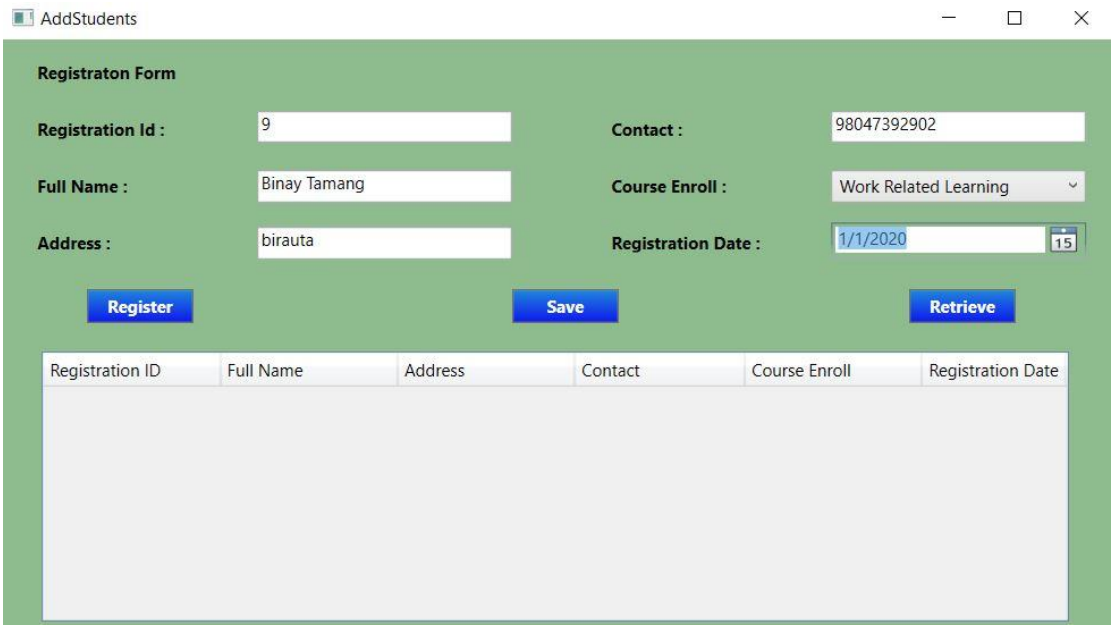



*Figure 9: Filling required data*

We should fill the registration form's required data and click the register button.

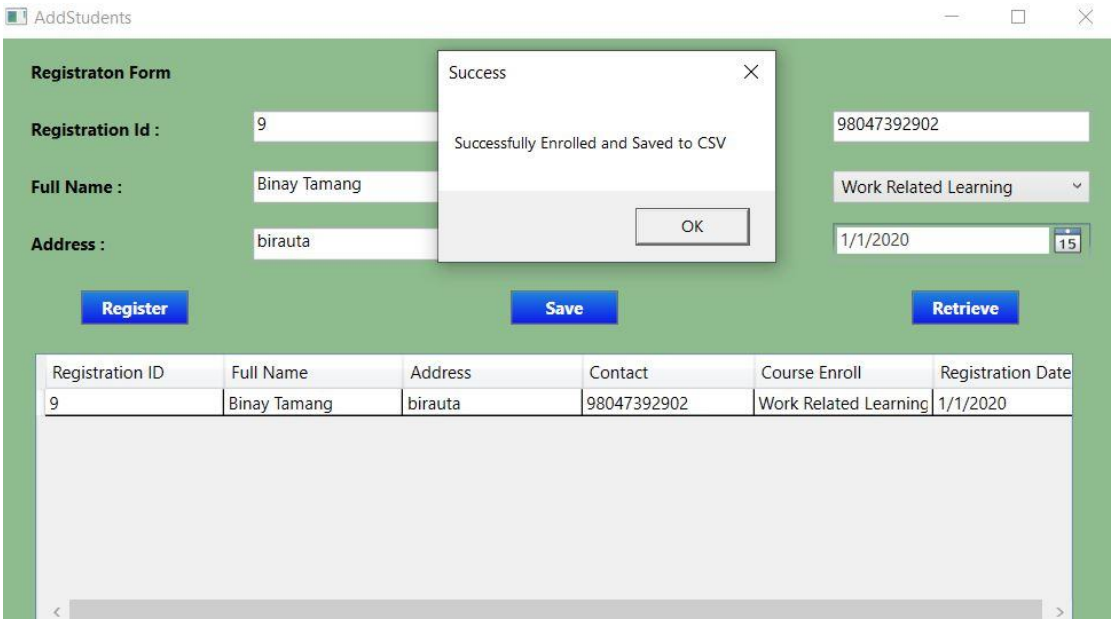



*Figure 10: clicking register button*

Then, system enrolled the students and save the data to the "studentDetails.csv" file which is located inside Debug folder. It also shows the data on the data grid below.

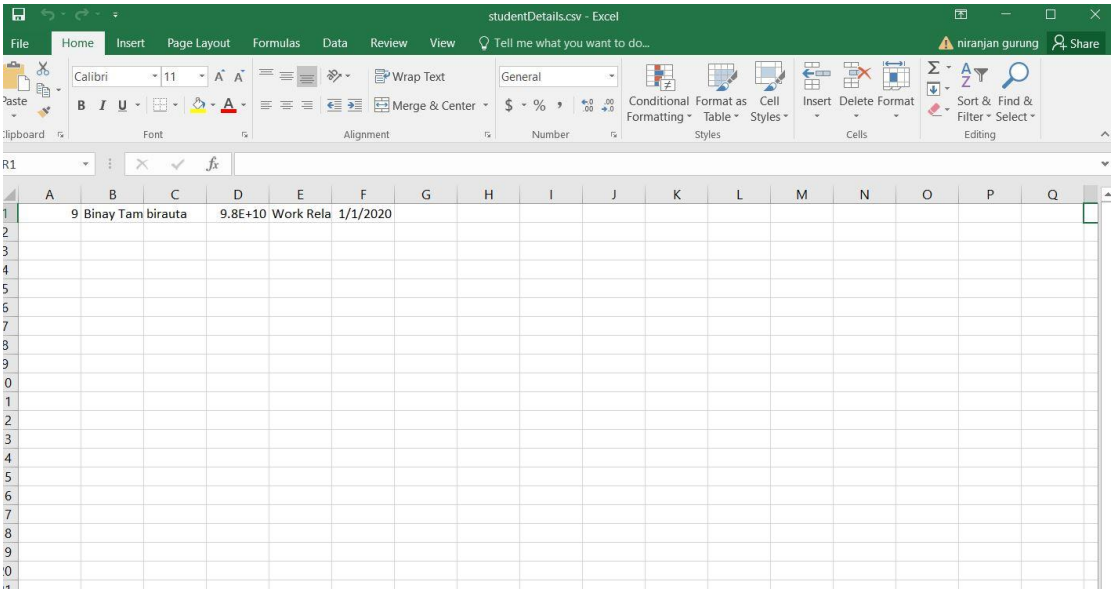



*Figure 11: Created .csv file with the added student's details*

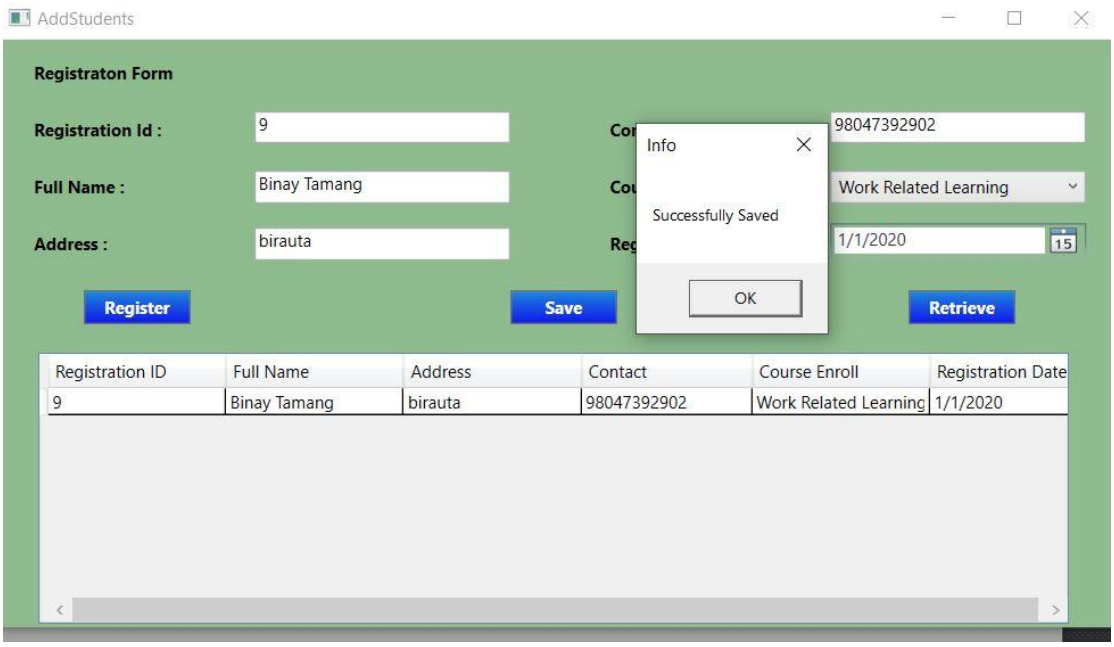



*Figure 12: saving the individual's data in .XML form*

By clicking save button we can create the "individuals .xml" which will be save in individual.xml inside Debug folder.

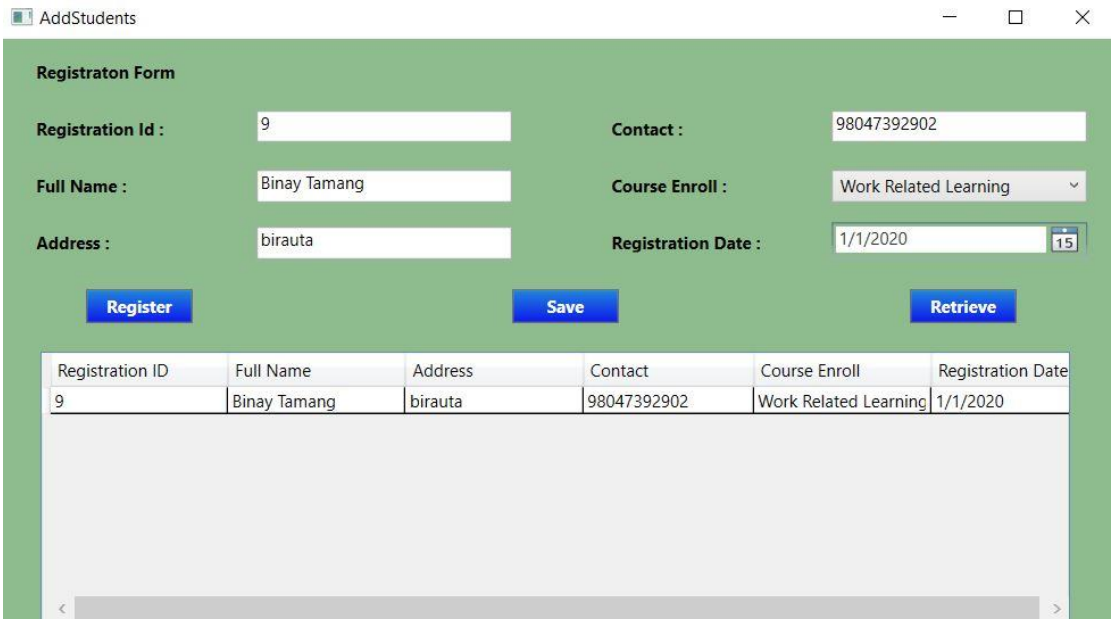



*Figure 13: Retrieve button*

By clicking retrieve button, we can also retrieve the data to edit or update.

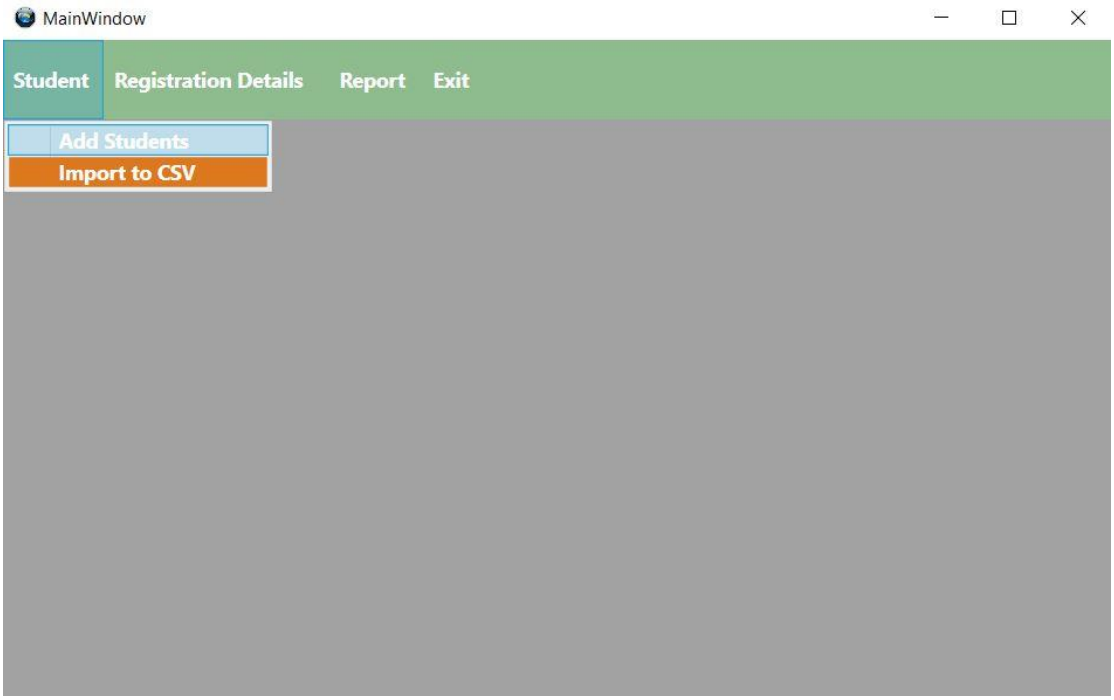



*Figure 14: import to CSV option*

By clicking Import to CSV, we will find the following window.


*Figure 15: import to CSV window*

Clicking the “Import from CSV” button, we browse the file and open the .csv file to import its bulk data.

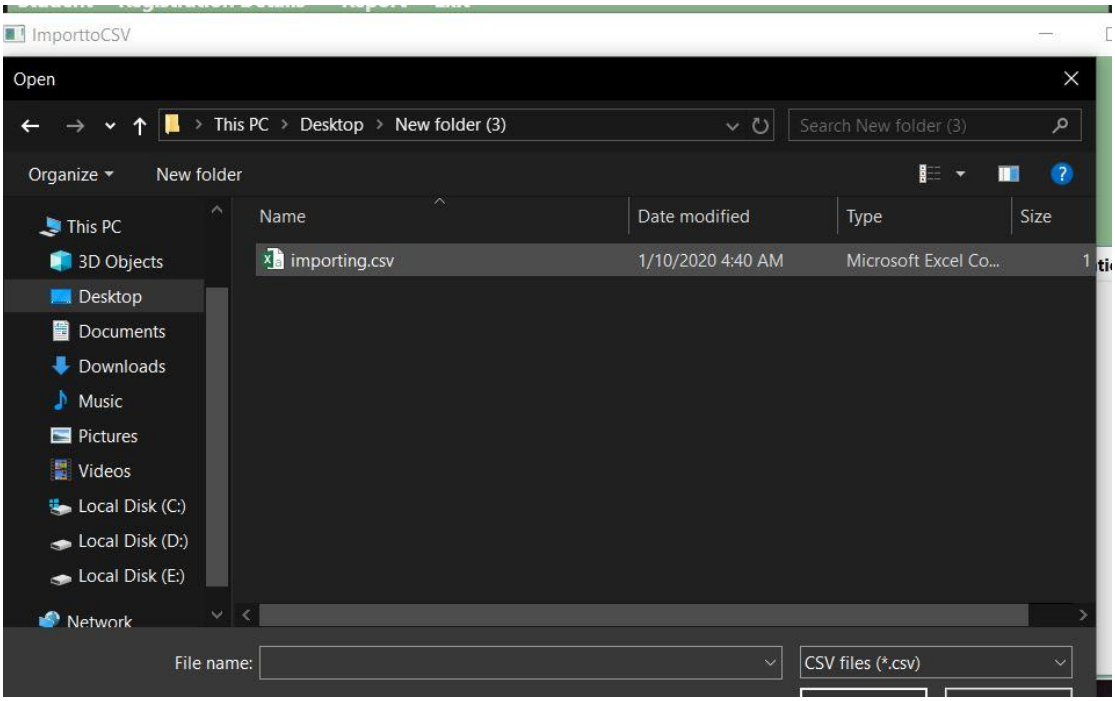
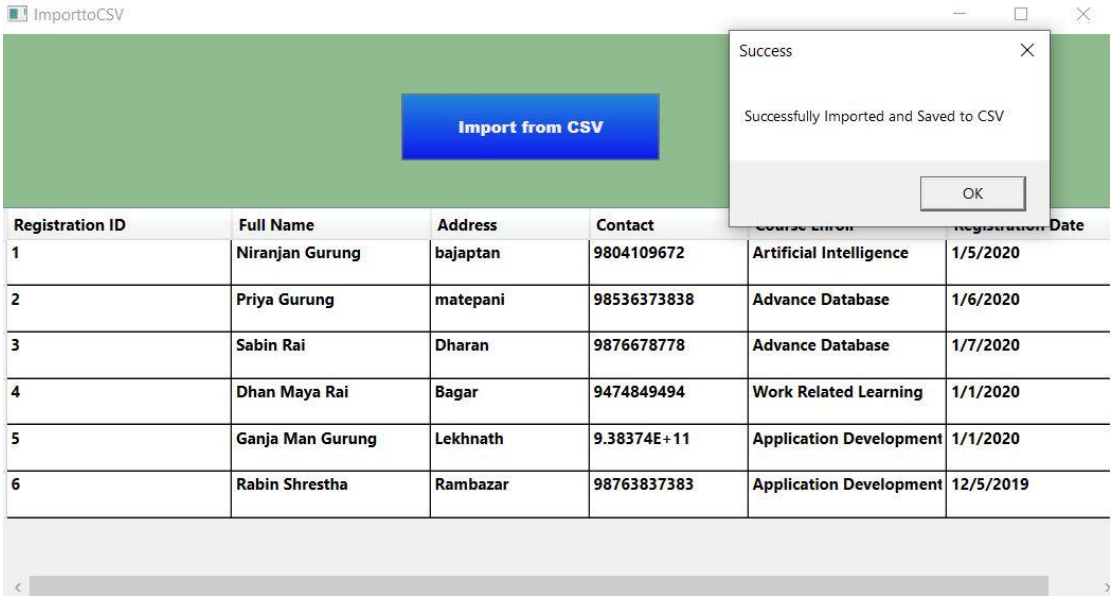



*Figure 16: dialogue box*




| Registration ID | Full Name | Address | Contact | Course Enroll | Registration Date |
|---|---|---|---|---|---|
| 1 | Niranjan Gurung | bajaptan | 9804109672 | Artificial Intelligence | 1/5/2020 |
| 2 | Priya Gurung | matepani | 98536373838 | Advance Database | 1/6/2020 |
| 3 | Sabin Rai | Dharan | 9876678778 | Advance Database | 1/7/2020 |
| 4 | Dhan Maya Rai | Bagar | 9474849494 | Work Related Learning | 1/1/2020 |
| 5 | Ganja Man Gurung | Lekhnath | 9.38374E+11 | Application Development | 1/1/2020 |
| 6 | Rabin Shrestha | Rambazar | 98763837383 | Application Development | 12/5/2019 |

*Figure 17: imported data to CSV file*

After Imported the data, it shows in the system in data grid and also saves the file automatically in studentDetails.csv.

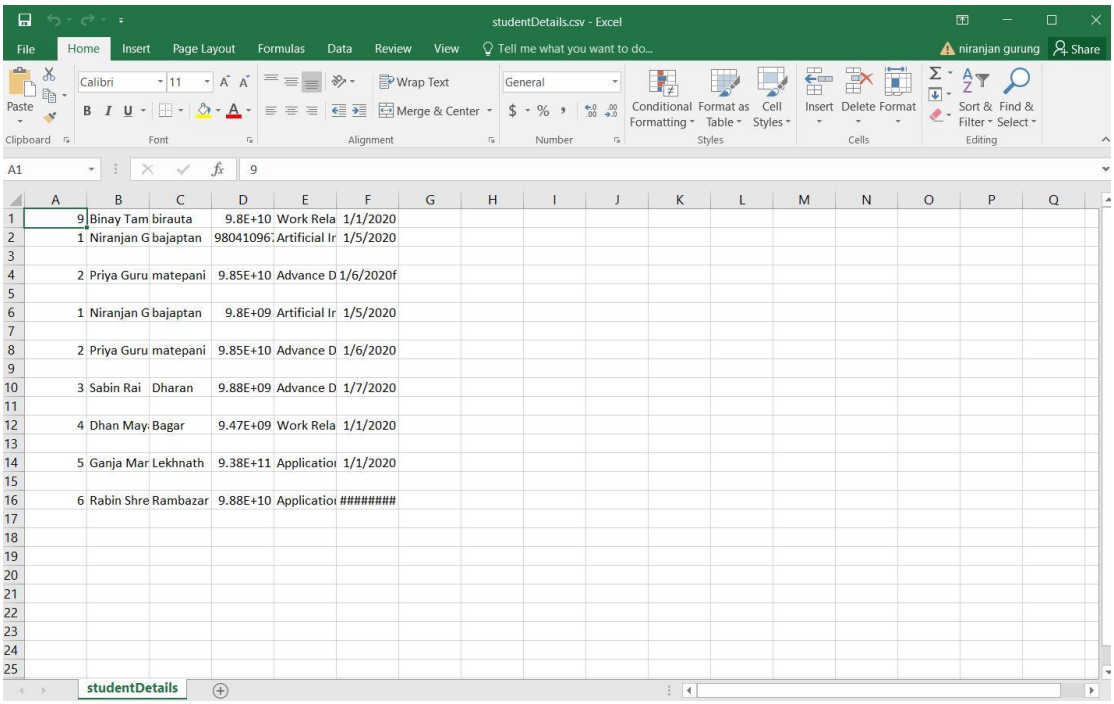



*Figure 18: Saved data in CSV file*

We can see the imported data in above in “studentDetails.csv” file.

- **Registration Details**

  Here, we retrieve the Students data from the saved file. In this case, “studentDetails.csv” and we can sort them from the full name or registration date.

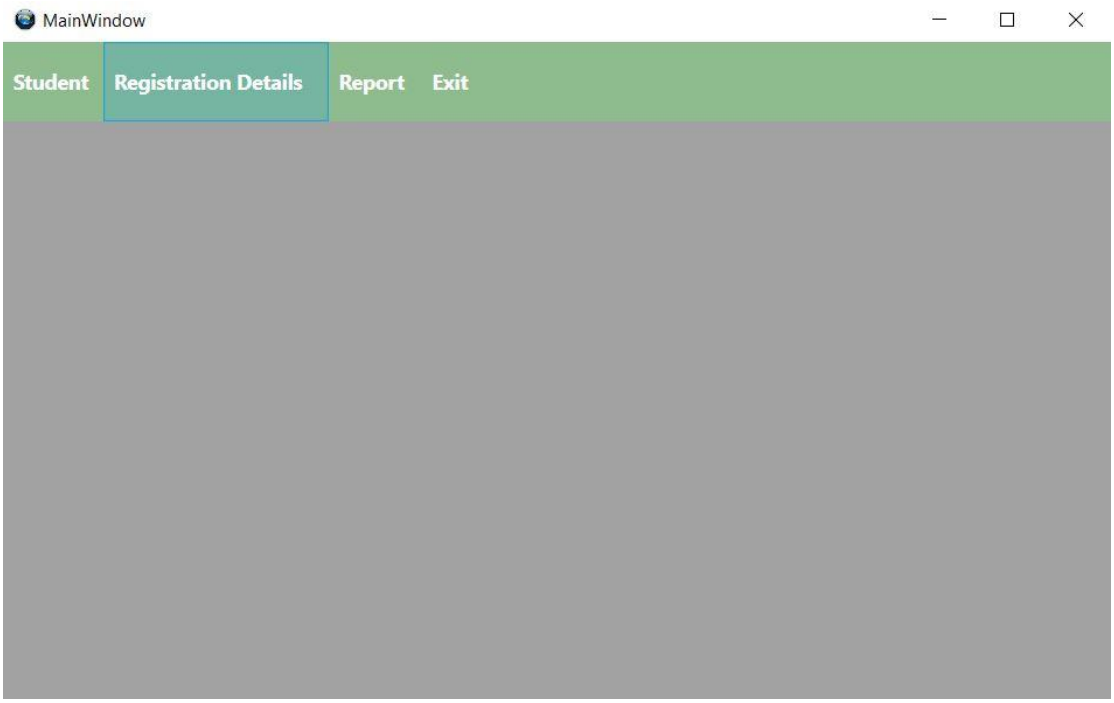



*Figure 19: clicking Registration Details*

First click in the registration details as shown in above. Then it shows the student’s registration details window as shown below.

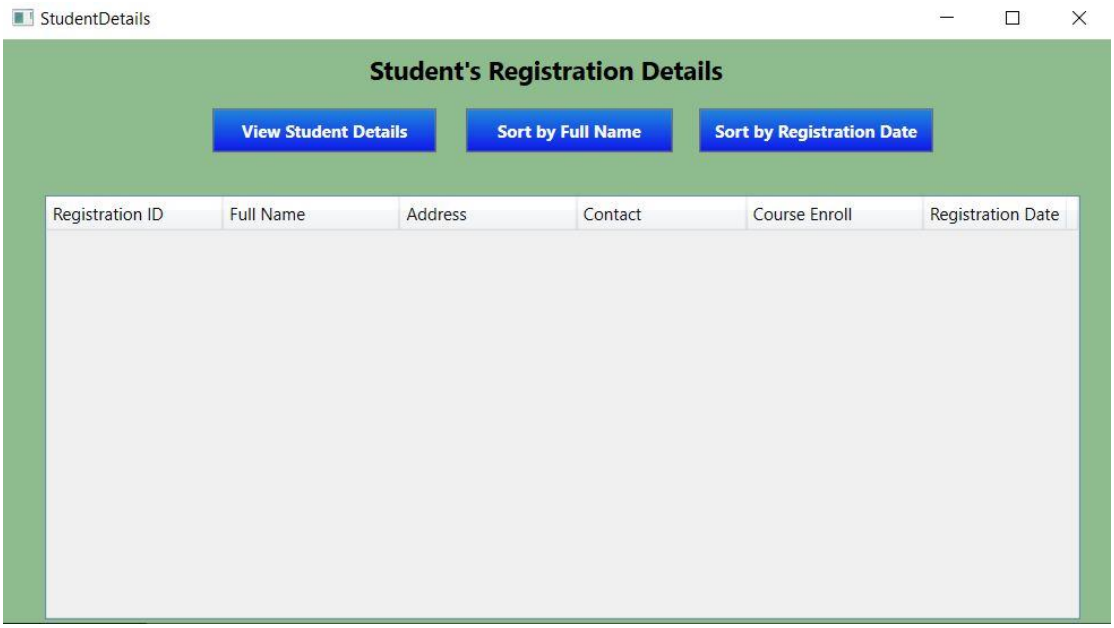



*Figure 20: Student's registration Details window*

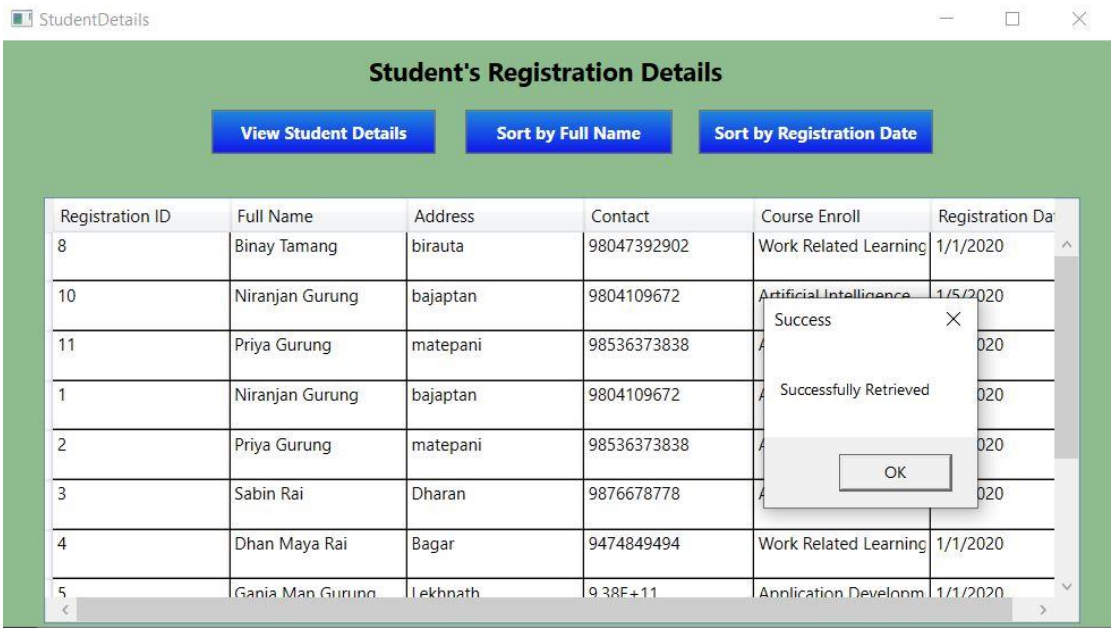



*Figure 21: viewing student details*

We can see the student's details in the data grid by clicking the "View Student Details" button. Then, we can sort them by full name and the registration date simultaneously by clicking the "Sort by Full Name" and "Sort by registration Date" button.

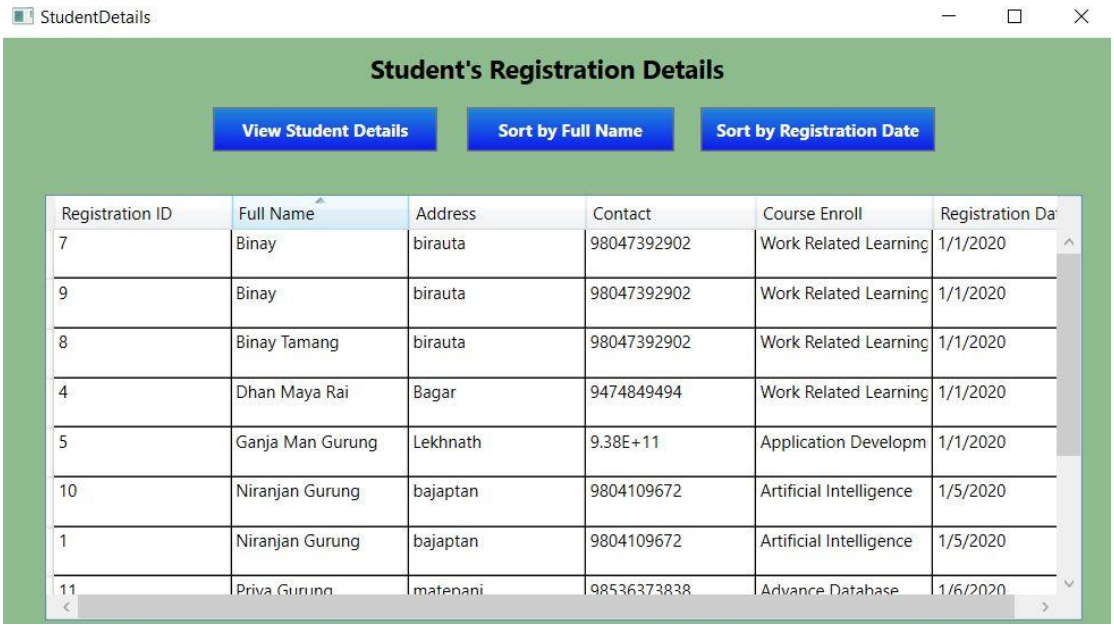



*Figure 22: sort by full name*


*Figure 23: Sort by Registration Date*

- **Report**

  Inside Report, we have two options for weekly report and chart view of the report.

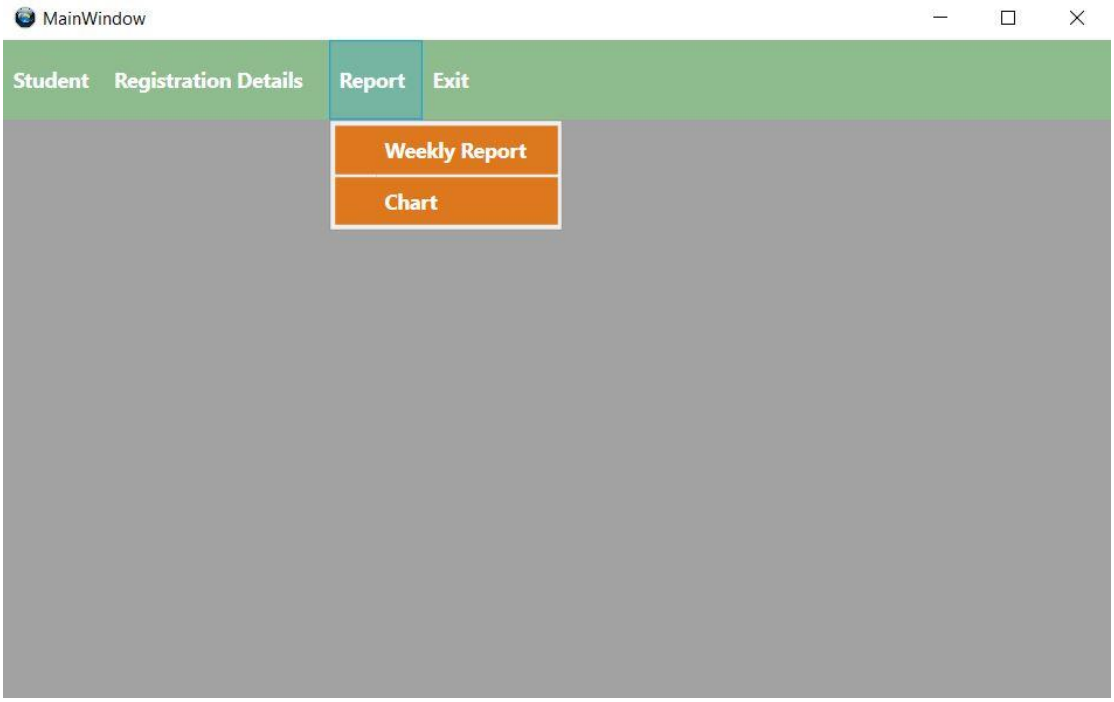



*Figure 24: Report*

Here, we can create the weekly report of total number of students enrolled in different subjects in the tabular form.

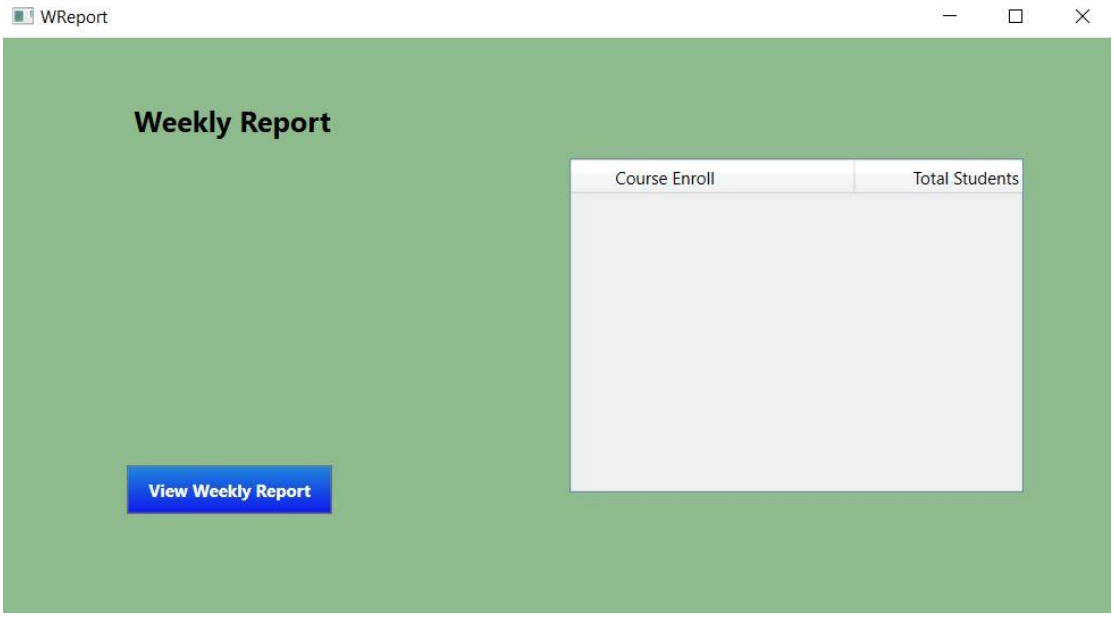



*Figure 25: Weekly report Window*

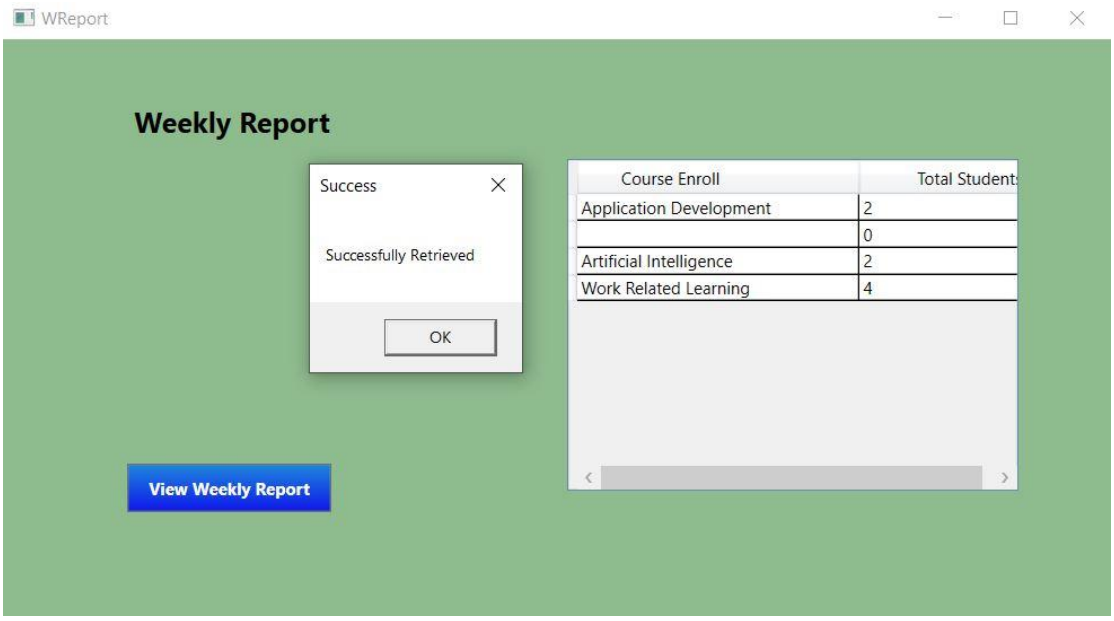



*Figure 26: Weekly Report Display*

In addition, we can see this report in chart too.

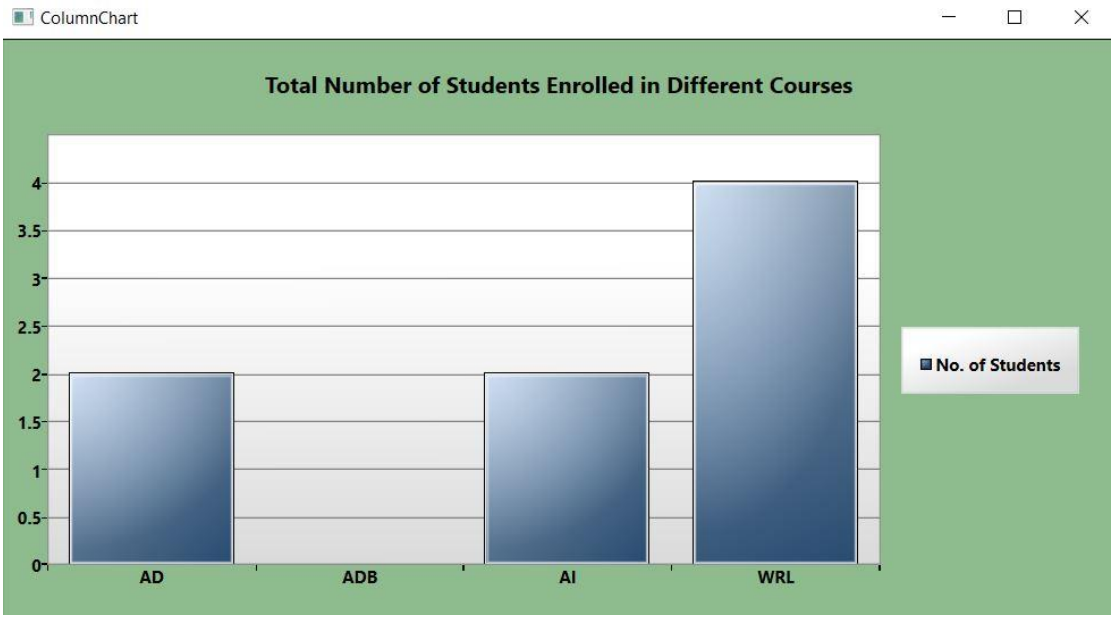



*Figure 27: Chart Display*

- **Exit**

  It closes the system.

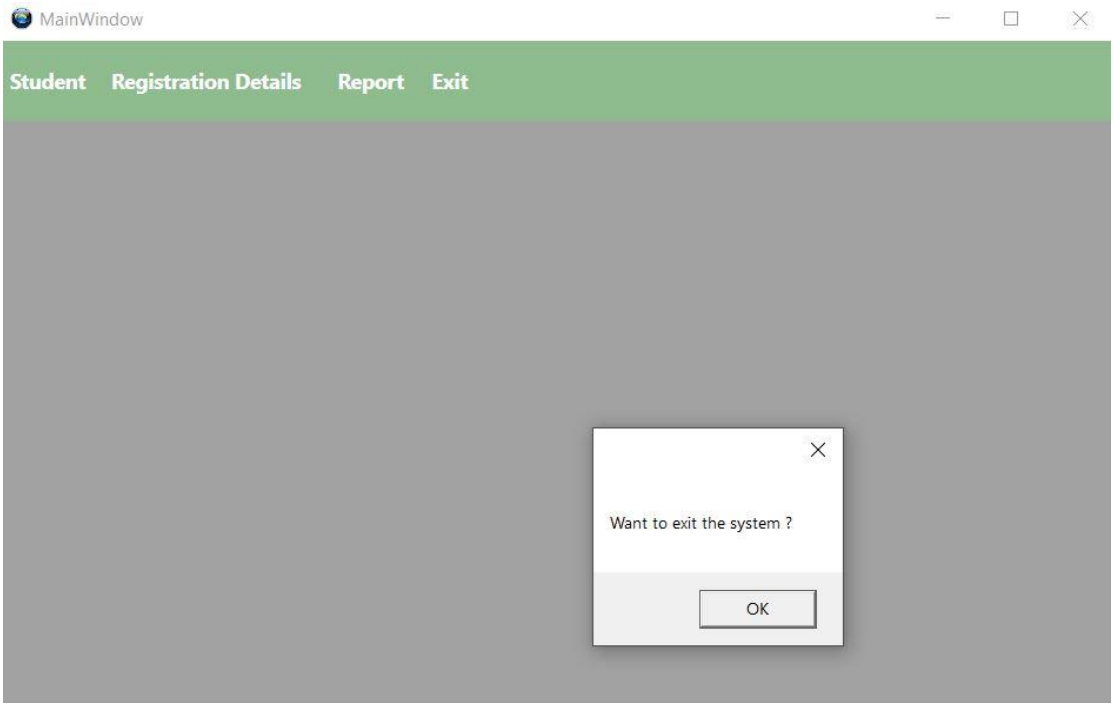



*Figure 28: Exiting the system*

## 3. System Architecture

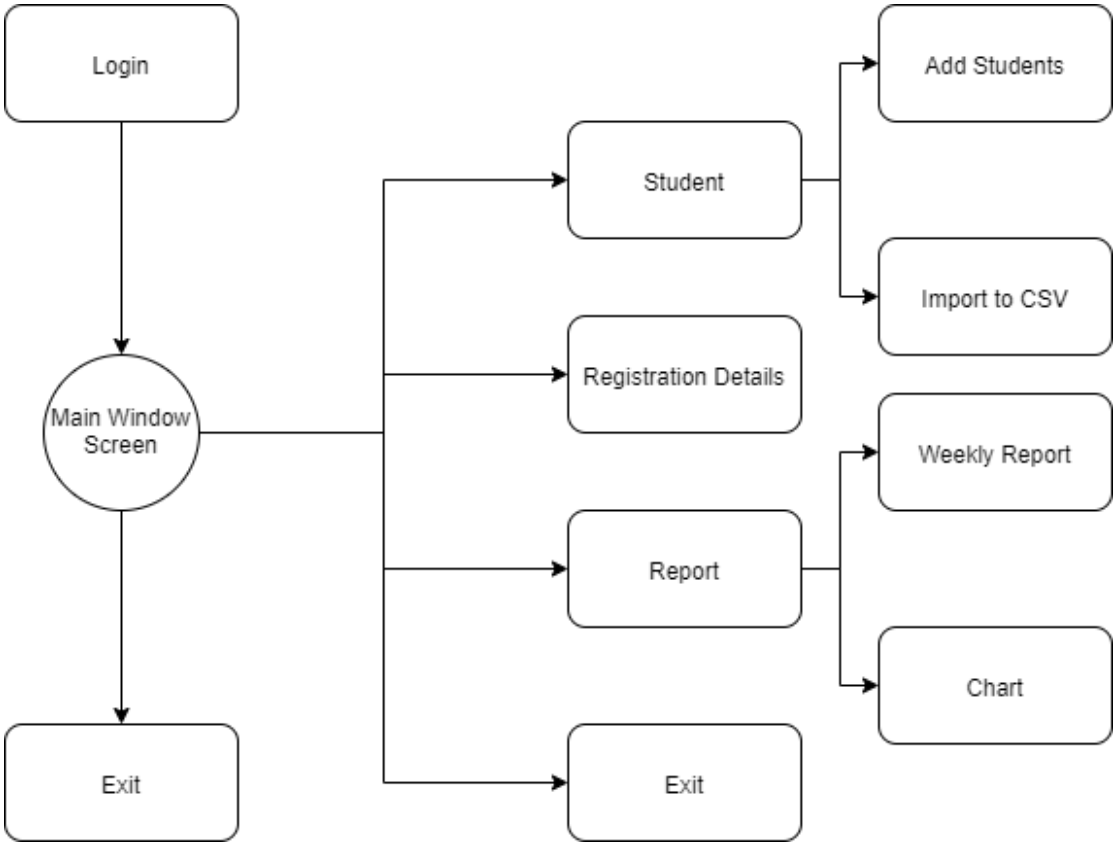



*Figure 29: System Architecture Diagram*

## 3.1 Class Diagram

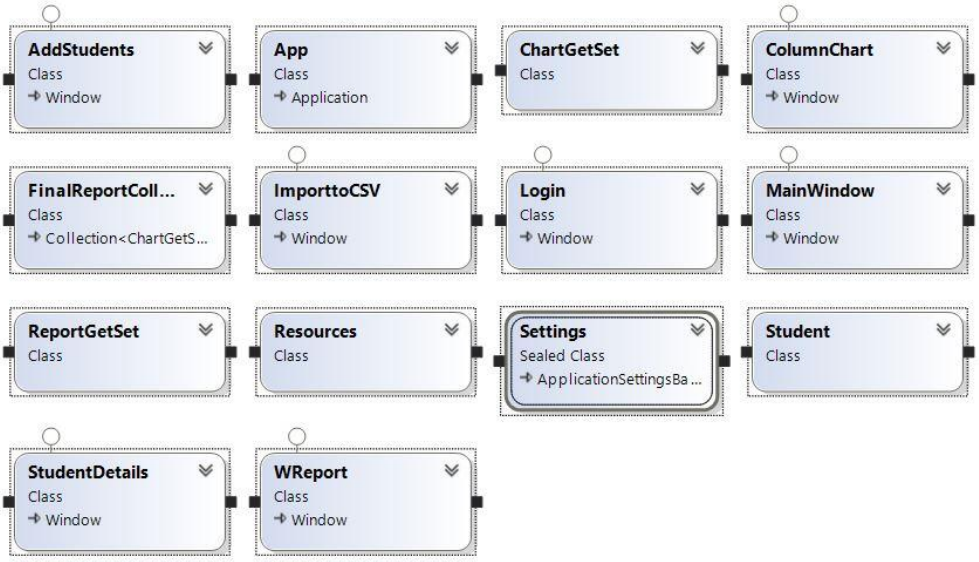



*Figure 30: Class diagram of Student_information_system*

### 3.2 Flowchart

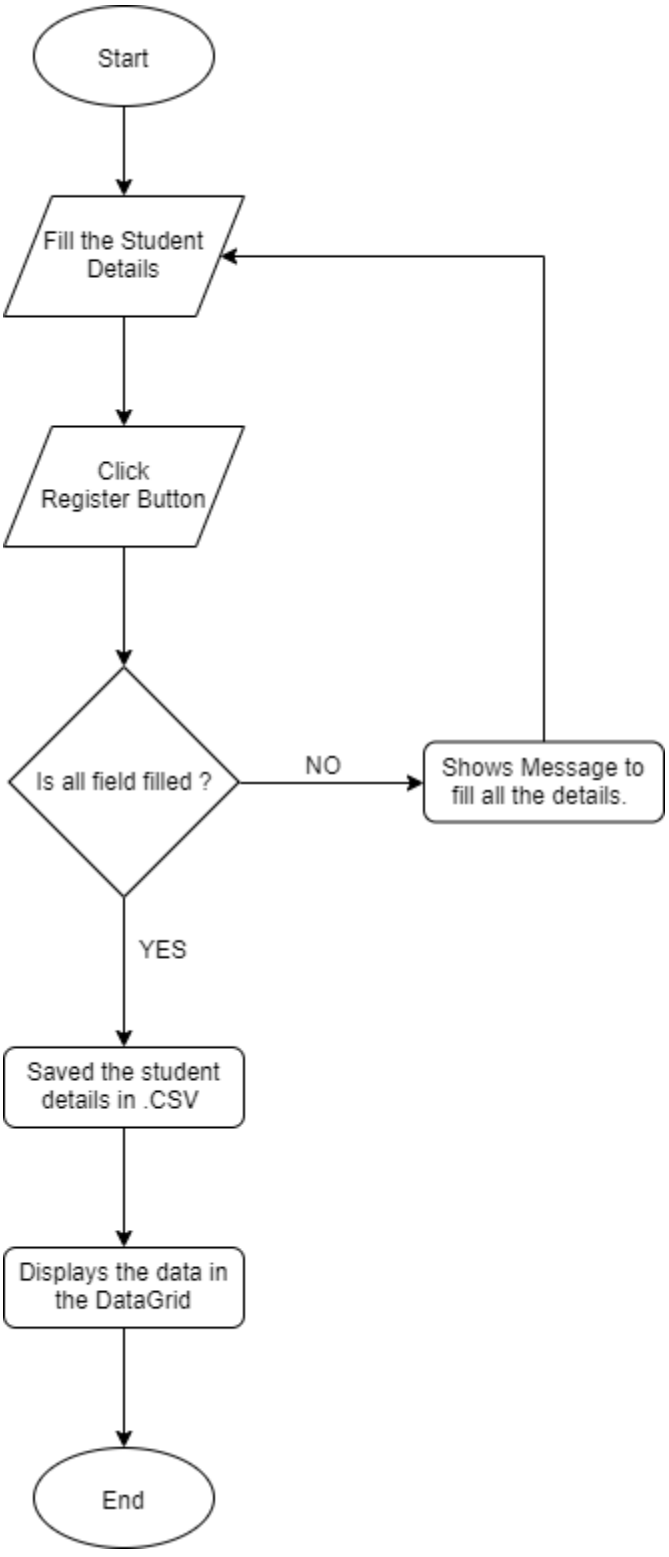



*Figure 31: Flowchart of Student Enrolment*

## 3.3 Methods Description

**The methods descriptions with their class are as follows :**

**Login Class**

- **BtnExitLogin_Click() :** When this method is called, the login window will exit.
- **BtnLogin_Click():** When this method is called, the login screen disappears and Home screen will appear.

**Main Class**

- **AddStudent_Click() :** It opens the page for enrolling the students.
- **ImportStudent_Click() :** It opens the page for importing the .csv file data.
- **Chart_click() :** It opens the page for viewing the bar chart.
- **RegistrationDetails_Click() :** It opens the page for viewing the details of the students.
- **WeeklyReport_Click() :** It opens the page for viewing the weekly report of the students.

**AddStudents Class**

- **SubmitRegistrationForm() :** It takes input from the text fields and saves text field data to the .csv file.
- **Savebtnclic() :** It takes input from the text fields and saves text field data to the .xml file using serialization.
- **Retrievetnclick() :** It retrieves data from the xml file and stores back to the text fields using deserialization.

**ImporttoCSV Class**

- **importtbn_click() :** It imports the .csv file from the user file explorer.
- **ReadFromCSV():** It reads the csv file selected by the user.

- **ExportToCSV():** It saves the data in new csv file that is imported from the user selected csv file.

**StudentDetails Class**

- **btnViewStudentDetails_click() :** It shows all the details of the students in the table.
- **btnSortByRegistrationDate_click ():** It sorts the table data according to the registration date.
- **btnSortByName_click():** It sorts the table data according to the first name of the student in the ascending order.
- **SortDataGrid():** It uses the sorting algorithm to sort the data grid.

**WReport Class**

- **btnViewWeeklyReport_click ():** It generates the weekly report of the students showing total number of students enrolled so far in the available courses.
- **ReadFromCSV():** It reads the CSV file.

**ColumnChart Class**

- **ColumnChart ():** It calls the ColumnChart class and displays the bar chart on the page.

**ChartGetSet Class**

- **cEnrolledName ():** It set and get the course name. The return type of this property is String.
- **totalStudentsEnrolled ():** It set and get the total number of students. The return type of this property is Integer.

**Report Class**

- **courseEnroll ():** It set and get the course name. The return type of this property is String.
- **totalStudents ():** It set and get the total number of students. The return type of this property is Integer.

**Student Class**

- **ID ():** It set and get the id number of the student. The return type of this property is String.
- **Fullname ():** It set and get the full name of the student. The return type of this property is String.
- **Address ():** It set and get the address of the student. The return type of this property is String.
- **Contact ():** It set and get the contact number of the student. The return type of this property is String.
- **courseEnroll ():** It set and get the course name. The return type of this property is String.
- **RegistrationDate ():** It set and get the registration date of the student. The return type of this property is String.

## 4. Conclusion

This system is developed in Microsoft Visual Studio 2019. Visual Studio 2019 is an integrated, complete solution with development tools, cloud services and extensions that enables to create great applications and games for desktops, the web, Windows Store, Android and iOS (www.comparex-group.com, n.d.). This system fulfils the entire functions that was mention on coursework. I have used the C# for the programming

language in Windows Presentation Foundation Client application using .Net Framework. It was fun to learn new language also there were many difficulties, which gave me challenges but with the help of module leader and doing research, I overcame all those difficulties. I have tries all my best to complete the coursework inside due date. Because of this system, I was very confident in data handling, error handling and coding. This project was the best way to increase my creativity and think differently.

## 5. References

www.comparex-group.com, n.d. [Online]
Available at: https://www.comparex-group.com/web/microsites/microsoft/products/development/ms-visual-

studio/ms-visual-studio-2019.htm
[Accessed 6 1 2020].

The End