Informatics College Pokhara

informatics
college   pokhara

Application Development

CS6004NI

Course Work 1

Submitted By: Rajan Gurung   **Submitted To:** Ishwor Sapkota
London Met ID: Enter ID Here   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.

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	very poorly explained and no diagramatic representation
Reflective essay	Very poorly written

C. Programming Style

Clarity of code,Popper Naming convention & comments	Very poor code
System Usability	very poorly developed application

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

Overall Comment:

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

Is sufficient for good and is clear on what is done.

Informatics College Pokhara

Application Development

CS6004NP

Coursework 1

Submitted By: **Submitted To:**

Student Name: Rajan Gurung   Mr. Ishwor Sapkota
Student ID: 17030526
Group: L3C1   Application Development
Date: 10-Jan-2020

Contents

Table of Figure

1 Introduction

The designed system is Student Information System. All the user interface and code were written and developed in C#. This system is a desktop based application without any web-based nor database application for a institution. Its UI is highly designed and is tested under numerous circumstances. The application allow the user to input the student personal detail including registration date so that the system can generate a weekly enrolment report of the student. This include detail like student ID, Name, address, contact no, email, course enrol, registration date. User can import a record from a text file such as files in .CSV format or create new record. All the features and functions that are required by Student Information System is fulfilled by the developed application. Other than this, it shows weekly tabular report showing total number of students enrolled so far in each course offered by the institution. The total number of student on each program (computing, multimedia, and networking) is shown in a chart. Last but not the least, other functions and features are well described in other section of the report.

2   User Manual

"Simplicity is the ultimate sophistication" by Leonardo da Vinci. As said by him, the way I look and understand is simplicity goes hand in hand with perfection. With that in mind, the UI, features and functions of the Student Information System is made as simple as possible for the user to interact without any trouble.

In this user manual, a detailed instructions to run the system step by step are described with additional screenshot of the system. It is also user-friendly and responsive.

Login

The initial screen of the application is the login window. This window can also be view as security screen. Without correct username and password user cannot access any features and functions provided by the application. The default username and password of the current application is "admin". Only a valid username and password can provide access to the system.
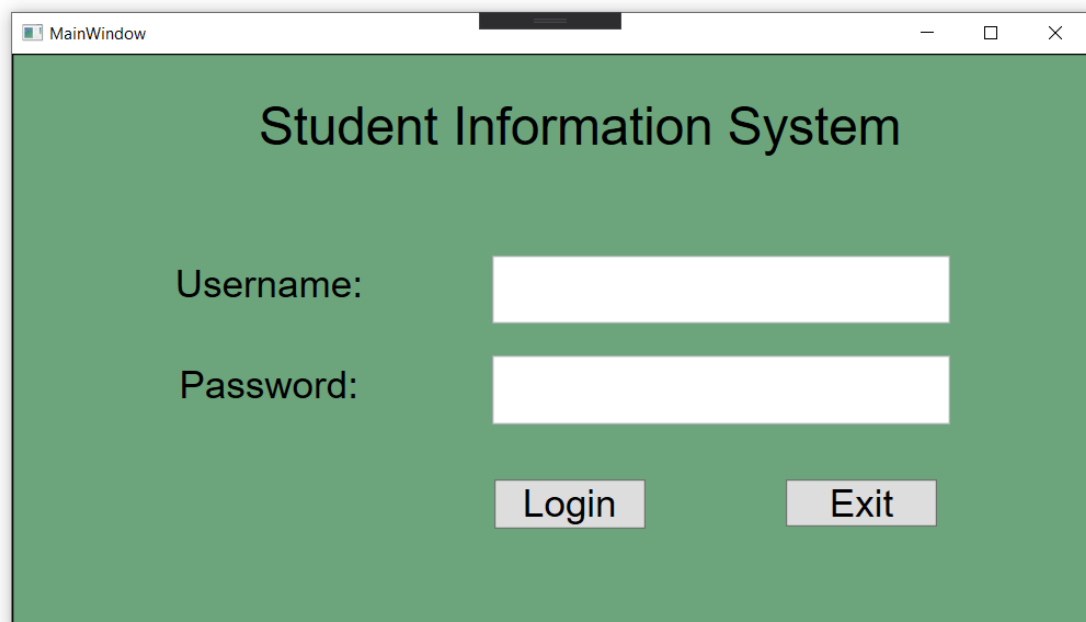
Figure 1: Login Window

Validation

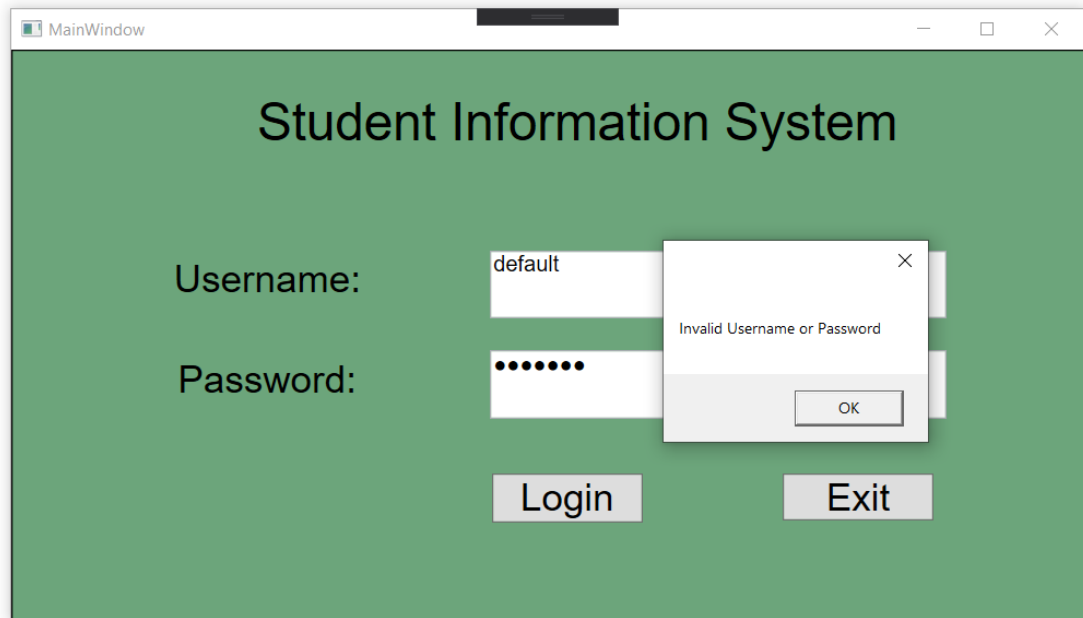If the username and password is not correct, it shows:

Figure 2: Login Window Validation

Home

Once the user enters the username and password successfully, the first window that appears in front of them is the home window. This is the main screen of the system which contains five buttons. They are Student Enroll, Generate Report, Import, Weekly Report and Chart. Each of these buttons opens different windows. Each of these buttons has its own features and are responsive. If a user wants to view weekly report of students then he/she clicks the weekly report which opens of the weekly report window. Same goes with all other remaining buttons. Since this window only contains buttons that goes to different other windows, there is no validation.
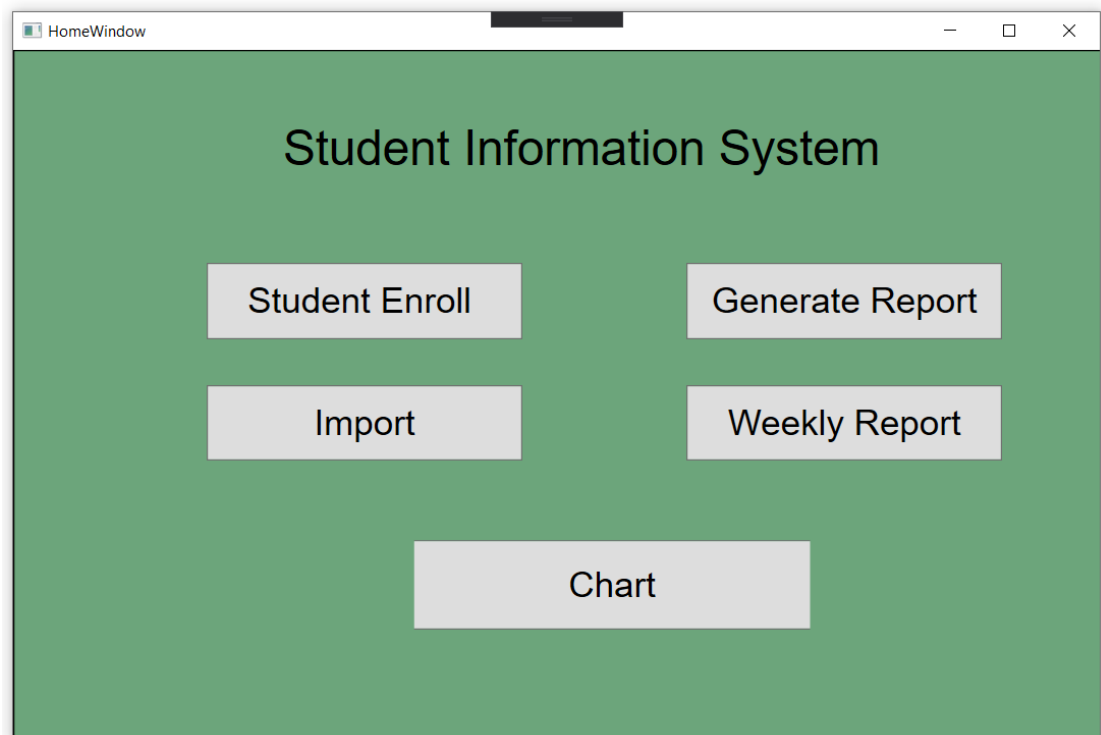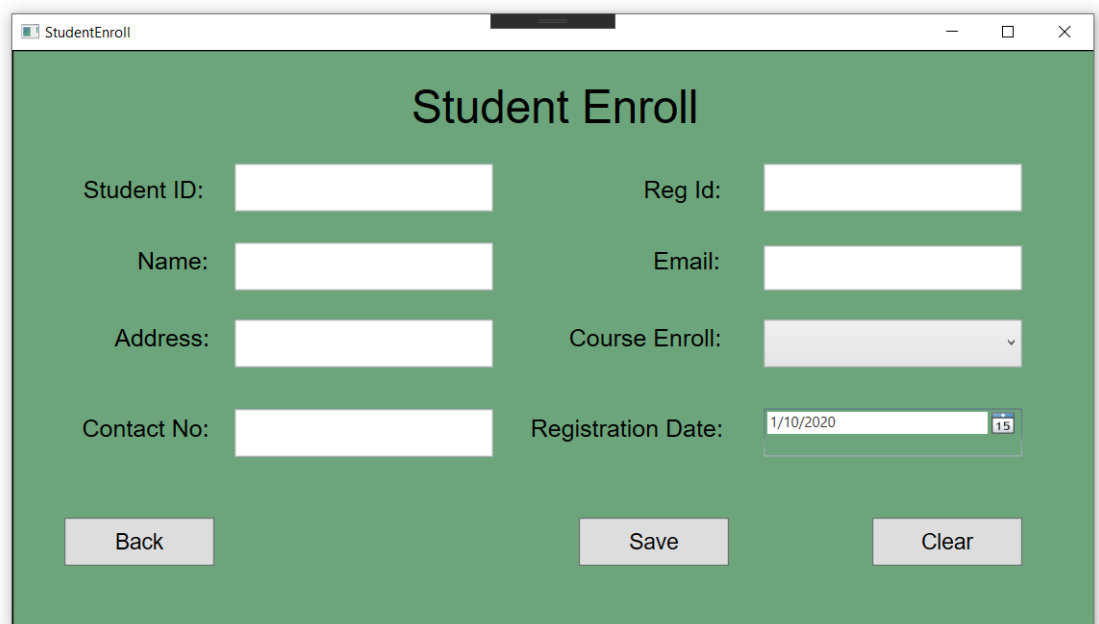
Figure 3: Home Window

Student Enroll

Student Enroll window is opened once the user clicks the Student Enroll button. It is a form that helps to enroll new students to the institution. Student can manually add there details such as registration id, student name, address, etc. in the textbox. Once the student adds all the required information, they click the save button create and save student details. Student can go back to the home window by simply pressing the "Back" button.
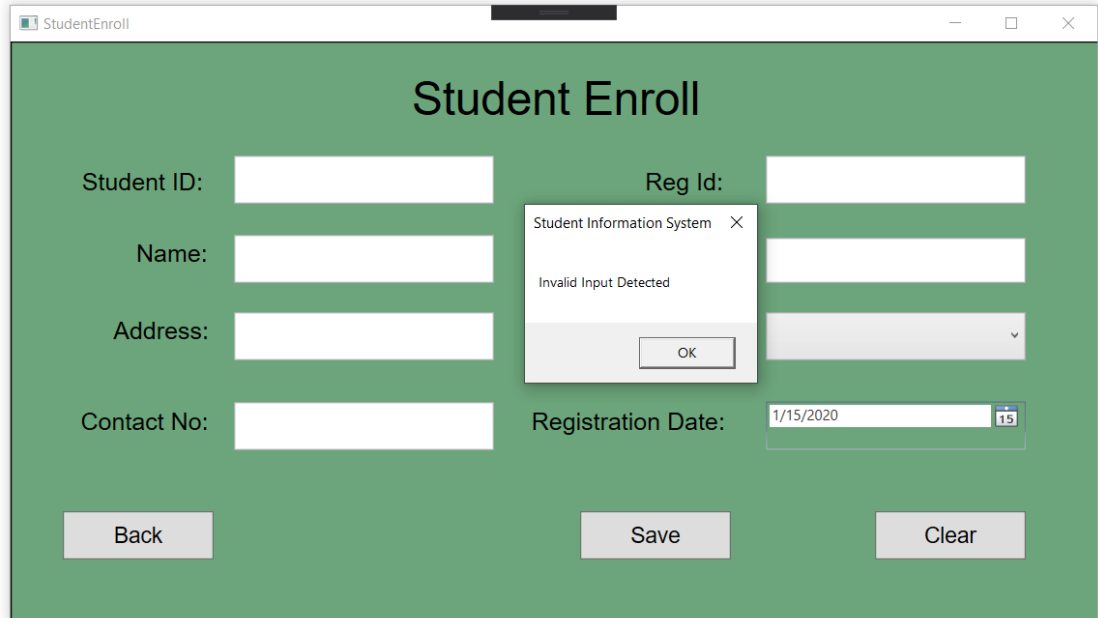

Figure 4: Student Enroll Window

Validation
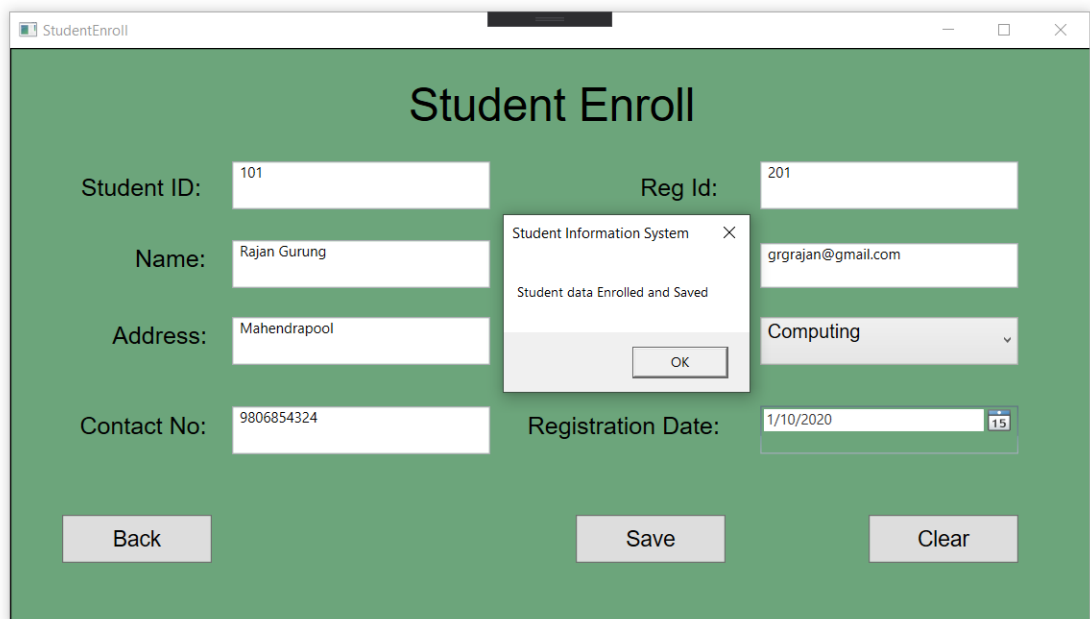
A message box appears if student does not fill up the form correctly.

Figure 5: Student Eroll Error Messagebox

After user enter their details correctly, it shows

Figure 6: Student Enroll Message

Generate Report

It consists of three buttons. To display the enrolled student data user can click on "Display Enrolled Student" button. User can also display two different reports, listing the students detail like id, name, course enrol and registration date one sorted by student first name and the other sorted by registration date.
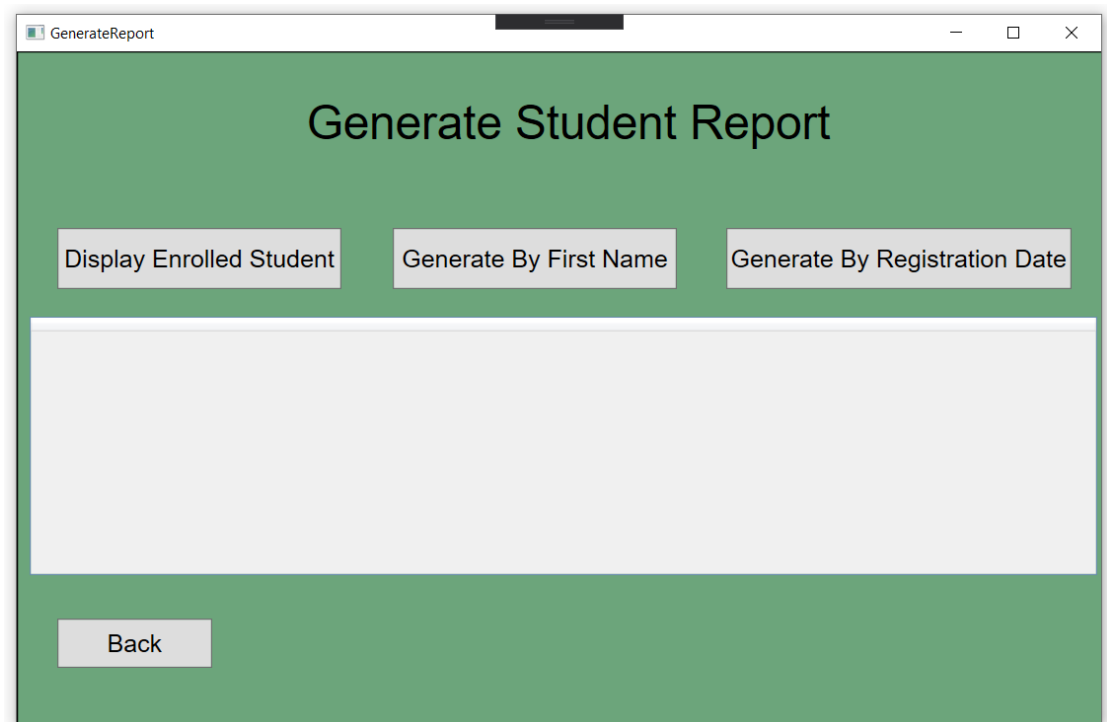

Figure 7: Generate Student Report Window

Validation

If the system cannot find .xml file, it throws an error showing in messagebox.
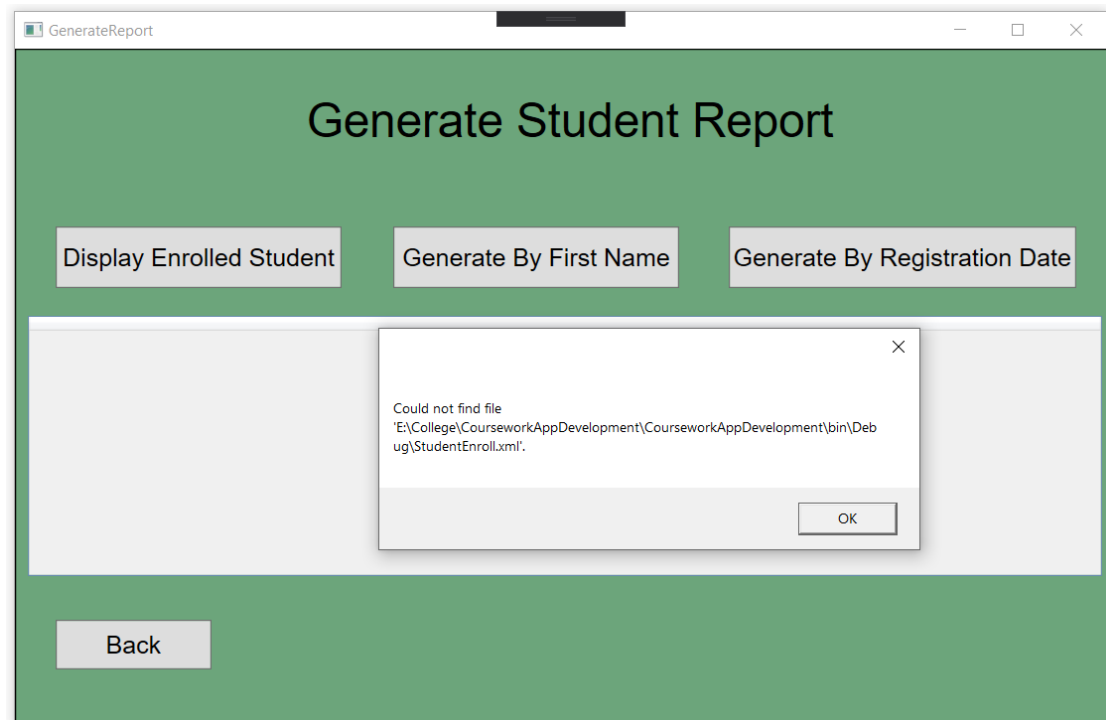

Figure 8: Generate Student Report Error

Once the student is erolled, it can be viewed in generate report by clicking the "Display Enrolled Student" button.
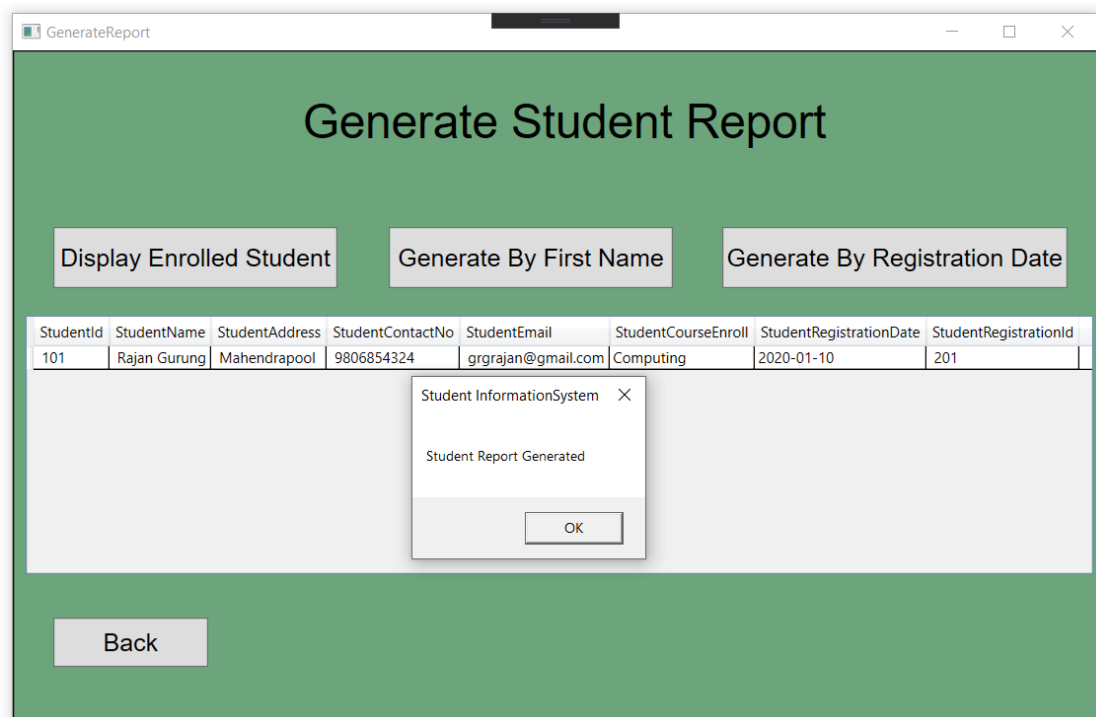
Figure 9: Display Enrolled Student
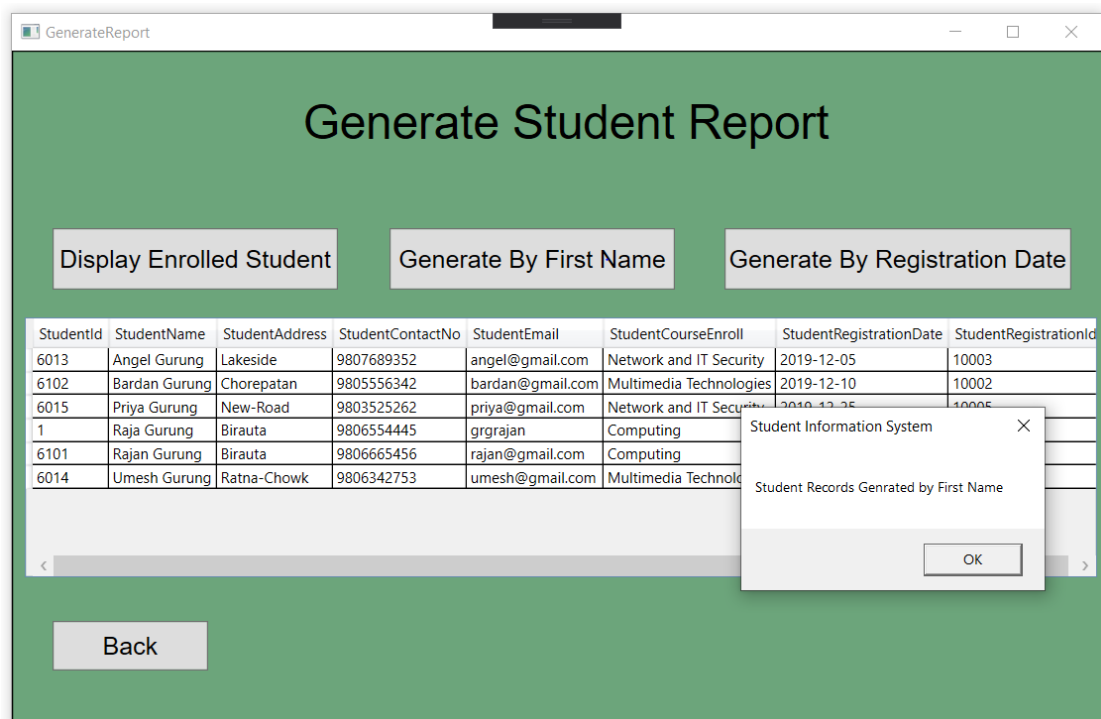
Figure 10: Generate By First Name

Figure 11: Generate By Registration Date

Import

In the import window, student can import record from a text file like in .CSV format. Once the record is import, it is shown in the datagrid table of "Generate Report" Window.
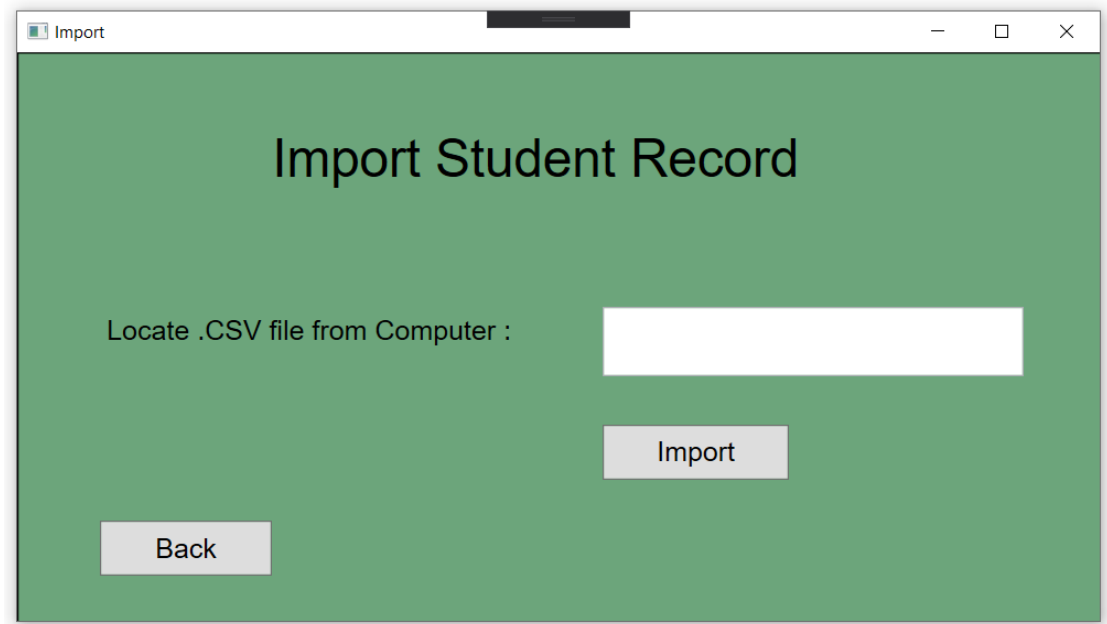
Figure 12: Import Student Record Window

Validation:

If user tries to import files without .CSV extension, it gives an error message.
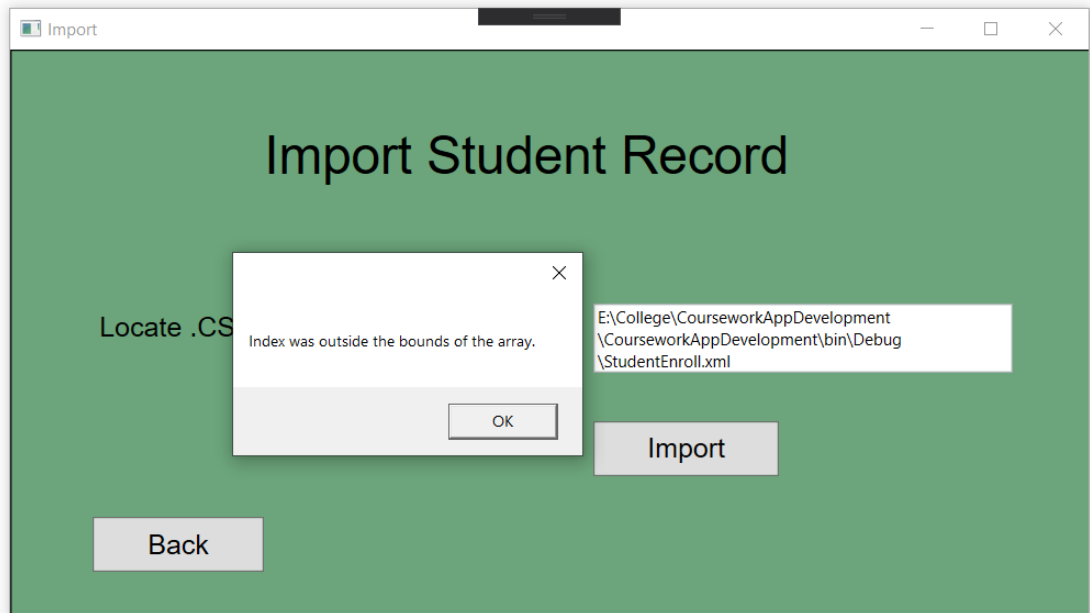
Figure 13: import Student Record Error Message
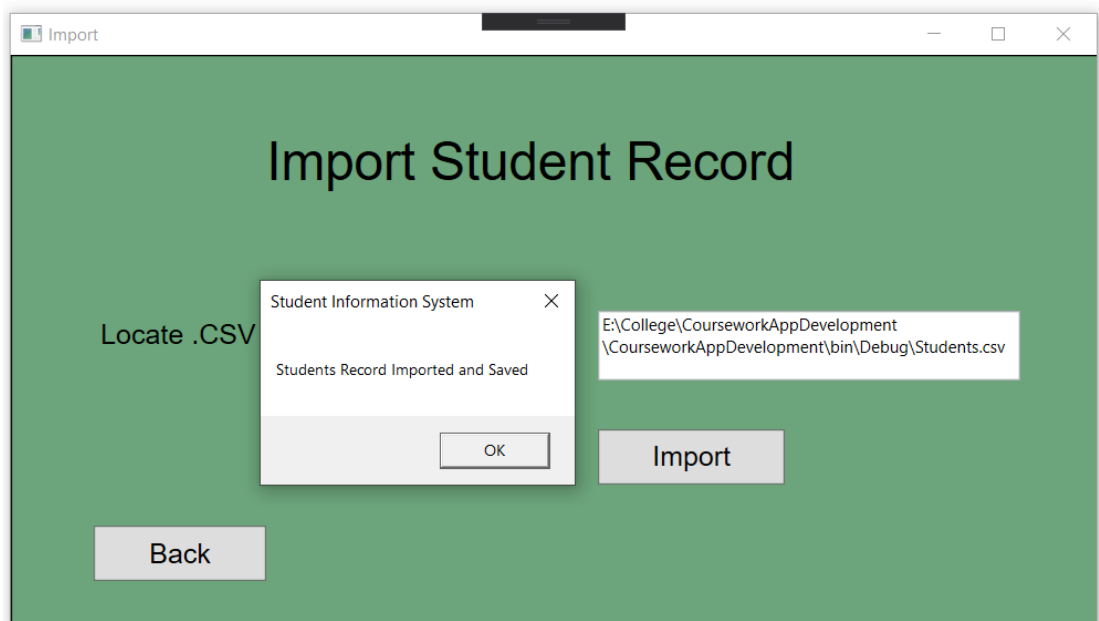
Figure 14: Import Student Record Saved
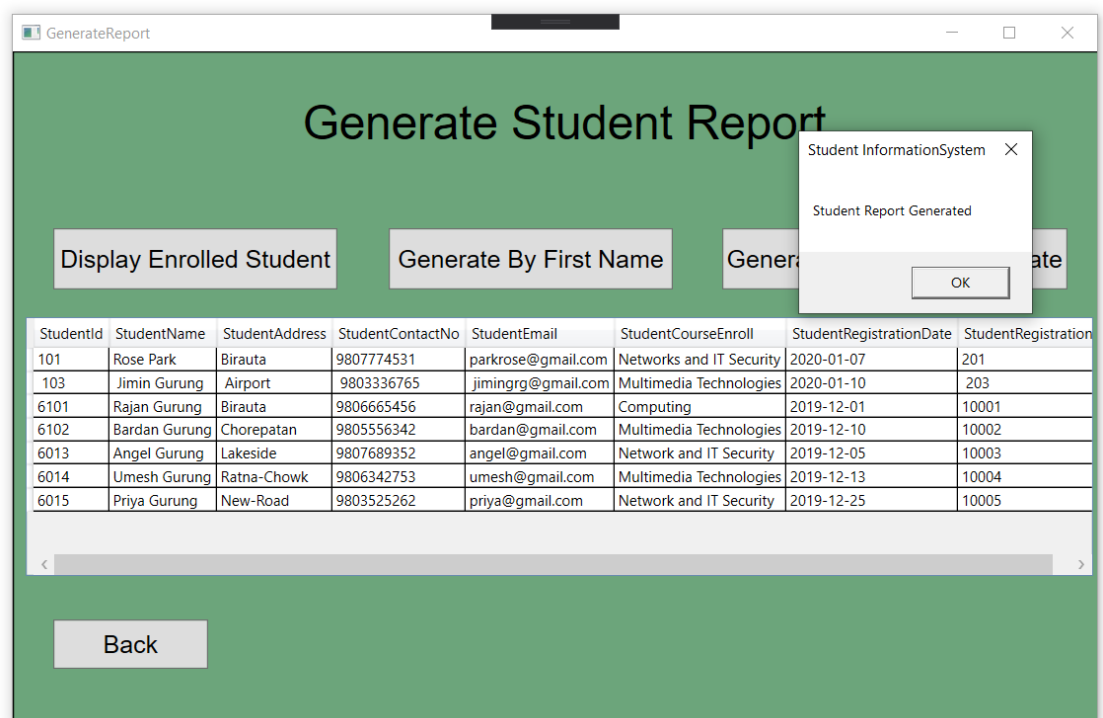
Figure 15: Display Imported Records

Weekly Report

In the weekly window, user can display weekly tabular report showing total number of students enrolled so far in each course offered by the institution by simply pressing the "Display Record" button.
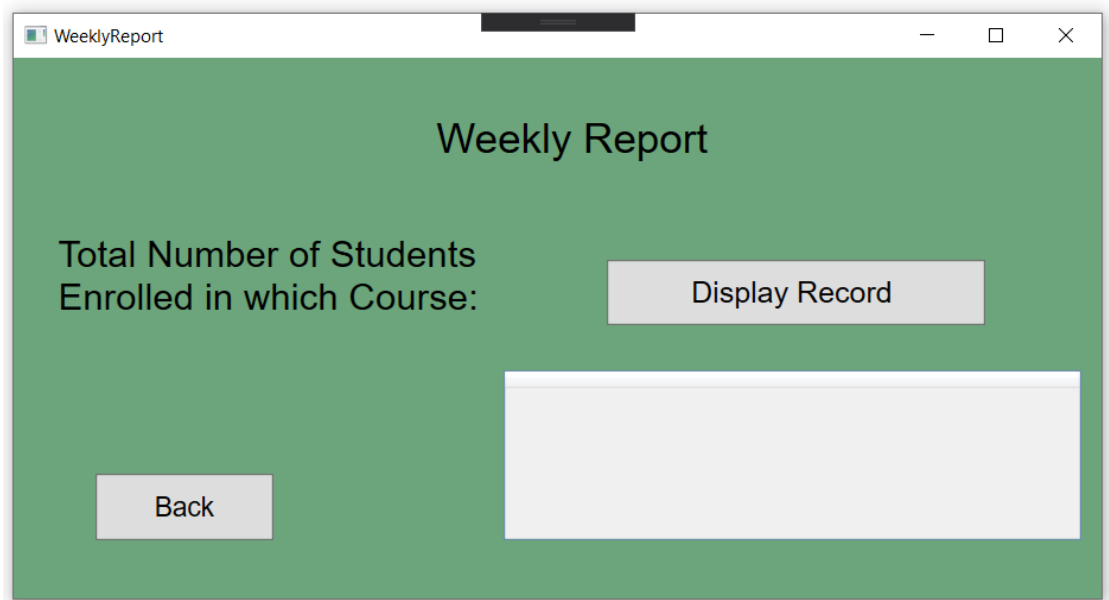
Figure 16: Weekly Report Window

Validation

If there is no record to display in .xml file it throws an error.
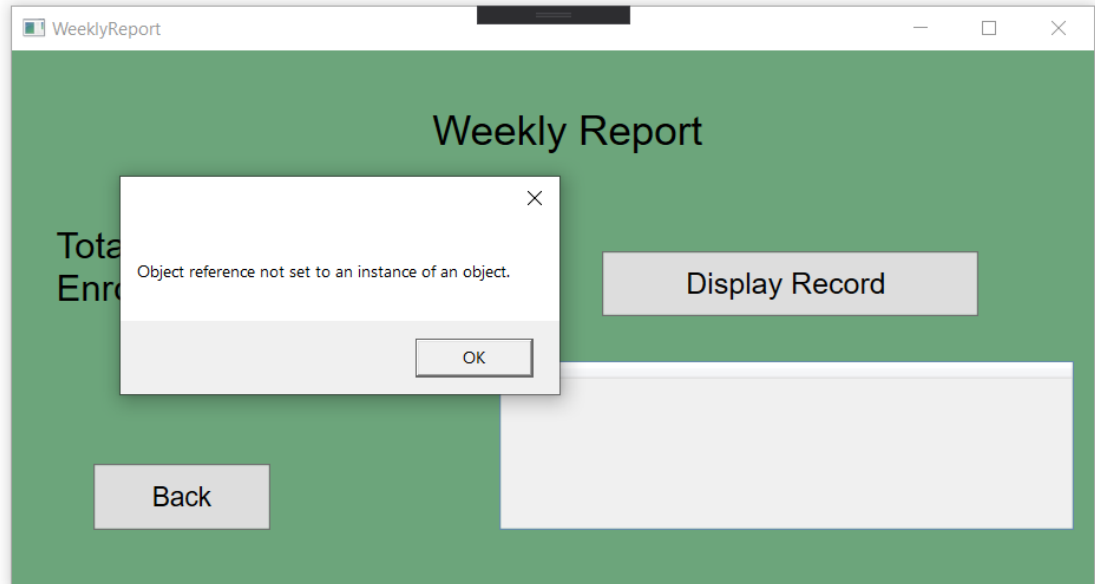
Figure 17: Weekly Report Error Message
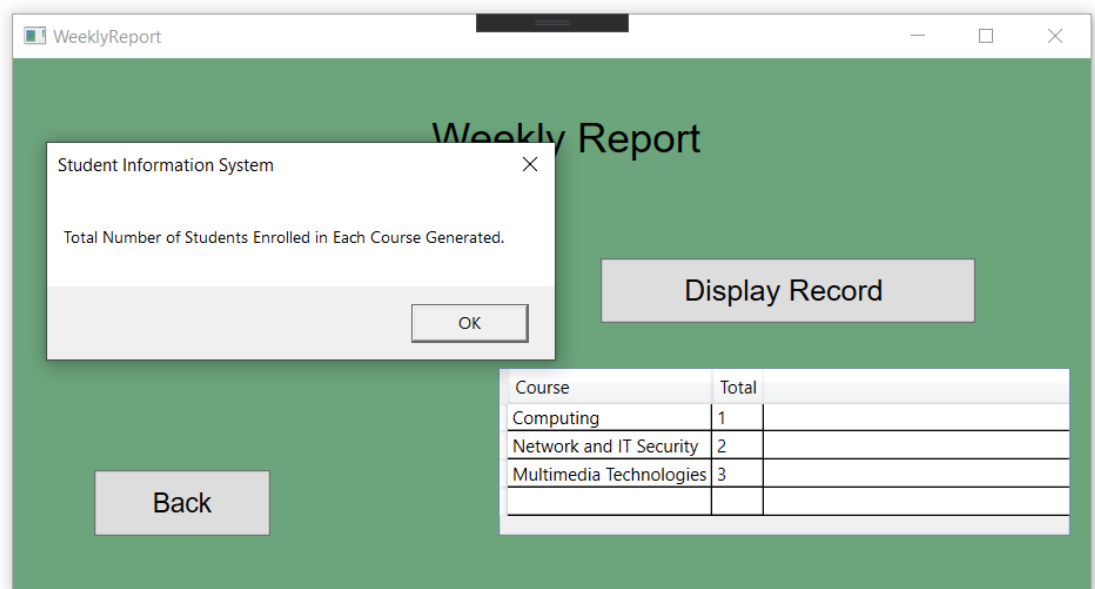
Figure 18: Total Number of Students Enrolled in Course

Chart

In this window, a pie chart is generated which shows the total number of students enrolled.
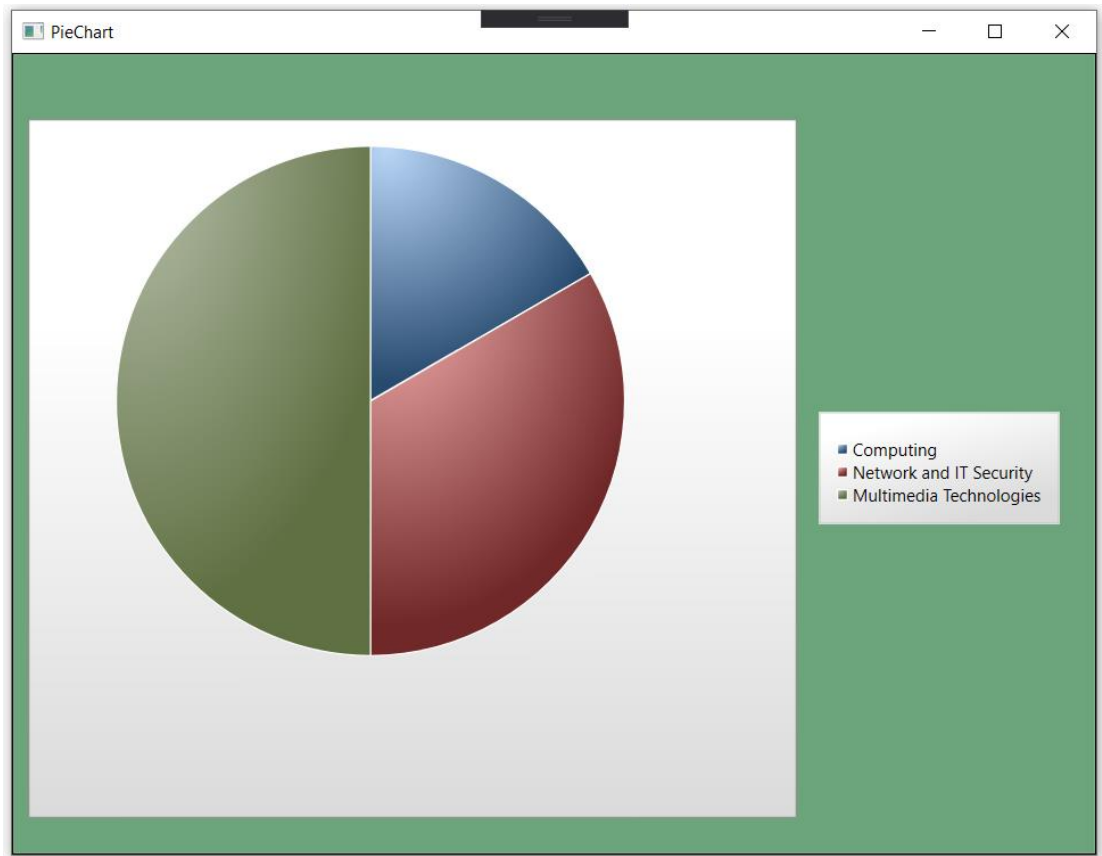
Figure 19: Pie Chart

3   Description of Classes and its method

StudentInformationClass

In this class, eight string based parameter are created and set to private so that it cannot be accessed by others. After setting all the parameter, get:set is applied to get StudentId, StudentName, StudentAddress, StudentContactNo, StudentEmail, StudentRegistrationDate, StudentRegistrationID.

StudentEnroll Class

All the activities related to student enrollment i.e. creating and entering new student details takes place. In this class, a form is created in which students fill-up and the data is saved in .xml file. It contains and shows student id, student name, address, contact number, email, registration date, registration id. The methods used in this class are:

AddStudentData()

This method contains StudentInformationClass obj, with string parameter RecordName. XMlSerializer class object is created and used which helps to control how objects are encoded into XML. This method helps to add new student data in .xml file in a single root node.

onClick Events

btnSaveStudentEnroll_Click

In this button, validation plus .xml file is generated by calling the AddStudentData() method. It shows "data enrolled and saved' message if the values are correctly entered.

btnClearForm_Click

This button clears up the entire form so that the new student can re-enter their respective details.

btnBack_Home_Click

Back button is created so that user can go back to the home window.

GenerateReport Class

This class contains four **onClick Events**. btnGenerateBack_Click is created to go back to home window, btnGenerate_Click is created to display enrolled student details, btnReportName_Click is created to display students details by first name whereas btnGenerateByregistrationDate_Click is created to show student details via registration date.

Import Class

There all altogether two **onCick Events**. btnImport_Click button is created to import .csv file and add it in the .xml file. If a user tries to enter other than .csv file it throws an error message whereas if the user imports the .csv file its shows "record imported and saved" message and the imported details can be view in weekly report window. The btnImportBack_Click button is used to go back to home window.

WeeklyReport Class

This contains StudentCourse() method. In this method, first it checks if the "StudentEnroll.xml" file exits or not. If it exists, it reads the file and display it in the datagrid. Like other window class, this contains two **onClick Events** buttons. The btnTotal_Click finds out the total number of student enroll in which course and shows a message "total number of students erolled in each course". If not it throws an error message. There is also a back button so user can go back to the previous window.

Chart

In this class, a pie chart is generated using the total number of students enrolled in each course. To make the chart, I downloaded WPF Toolkit from the website. (Mahesh Chand, 2019) After downloading it, I followed the step by step method as show in the website. If there is no course enrolled, it throws and error message. The other button is the back button.

4 System Architecture
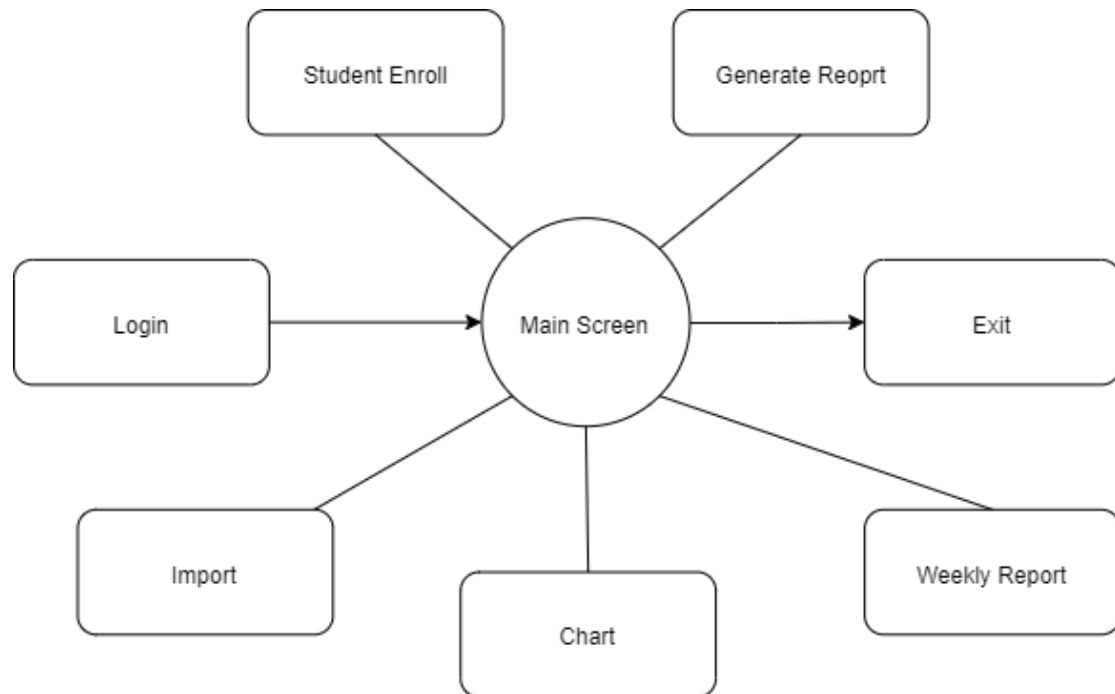

Figure 20: System Architecture

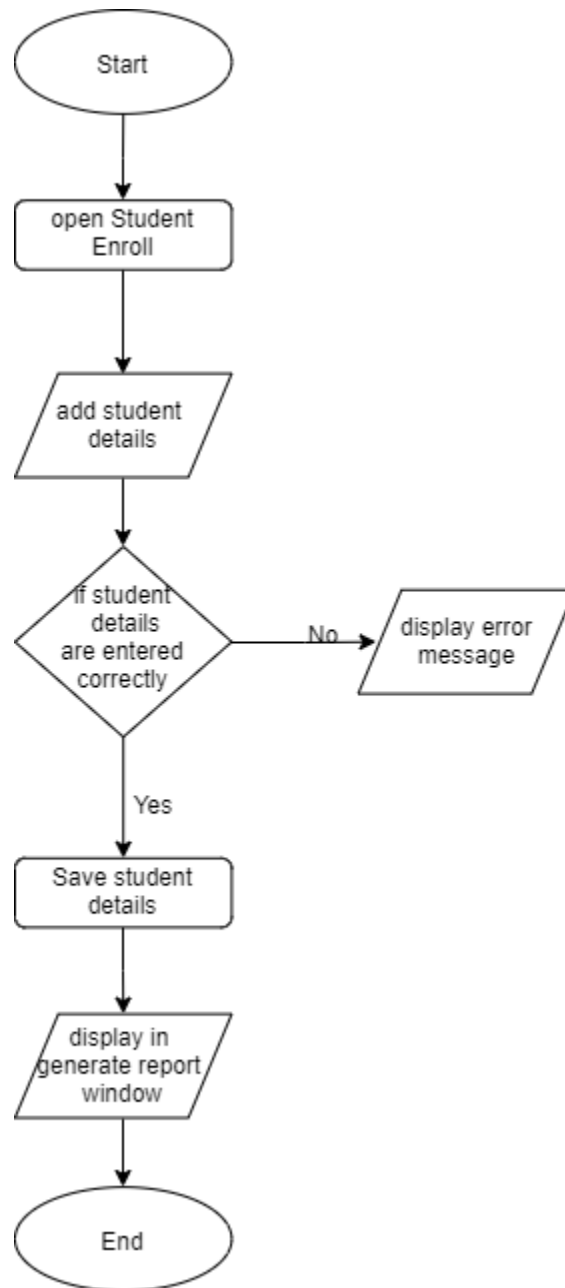# 5   Student Enrollment flow chat

Figure 21: Flowchart Student Enroll

6 Algorithms of Student Enroll

<u>Steps</u>

1. Start
2. Add student details
3. If student details are not entered correctly
4. Display error message
5. If details are entered correctly
6. Save student details
7. Display student details in generate report window
8. End

7 Reflection

The Student Information System is developed using Visual Studio Community 2019. The logic that were used in this system reflects the real world working events. The task was to develop a system for an institution with a well responsive and user friendly UI.

Up until, I have not used visual studio before. While using visual studio for the first time it felt similar to NetBeans as both of them had drag and drop features. But after using visual studio, for couple weeks, it felt easier than other platforms in terms of drag and drop. It is arguably one of the best features that I have used as our coursework is based on dag and dropping elements in Windows Presentation Foundation (WPF). With the help of drag and drop feature, I can easily create the design of my kind. It is much reliable, faster, and accurate than coding.

After preparing all the design, it was time to implement methods to the designed textbox, datagrid and buttons. While creating and saving a new .xml of student details, it keeps making several root nodes. The main problem of this was it keeps on making different root node for each and every time a student is enrolled and his/her details is saved. After searching for many hours, I finally found out how to solve it through stack overflow and watching YouTube tutorials. None the less, the system was developed with simple but effective UI so that the user of any age group can use the system without any trouble.

Bibliography

Mahesh   Chand,   2019.   *Charting   in   WPF.*   [Online]
Available   at: https://www.c-sharpcorner.com/uploadfile/mahesh/charting-in-wpf/
[Accessed 10 January 2020].