Module Code & Module Title

CS6004NA Application Development

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2019-20 Autumn

Name: Midhir Rana

College ID: NP04CP4A170017

University ID: 17030721

Abstract

This is an individual course work for the module “Application Development” for Student Information System which is developed using Visual Studio Platform using C# language. The coursework was released in the week 5 and supposed to be submitted in the week 11.

With the great contribution of Mr. Iswhor Sapkota, the course work was completed within the time frame.

Table of Contents

Table of Figures

1. Introduction

This is the report of the coursework which had been assigned to us. The task was to design and implement Student Information System in C# programming language using Visual Studio. The application had to be a desktop application which should allow users to input student details and then generate reports from the available data. The application should also provide the feature of importing data from a .csv file.

All the tasks of this coursework has been carried out and completed the project in time. The student information system was developed as per the coursework requirement and tested under various circumstances. This system is very much beneficial for any institution that provides academic courses and deals with numerous data. The proposed system helps to replace traditional methods of saving records. It helps to organize and preserve data. Time will be saved and numerous data can be inserted in short period of time. The details of any students, total number of students in any programme can be viewed easily. Overall, the system as it is named student information system is actually a very effective software for keeping records. It makes easier for any academic institution to handle large data.

Even though we were able to develop this kind of system, some institutions in Nepal are still using traditional way of keeping records in files. With the traditional methods, it is very time consuming and there is risk of losing data. The system which we have developed would aid those institutions in every way possible. Student information saved as hard copies can be accessed by anyone and can lead to misuse. To prevent this, student our system ensures that only authorized people have access to private student information.

2. User Manual

Below are the instructions with screenshot for end users which demonstrates how to use the system.

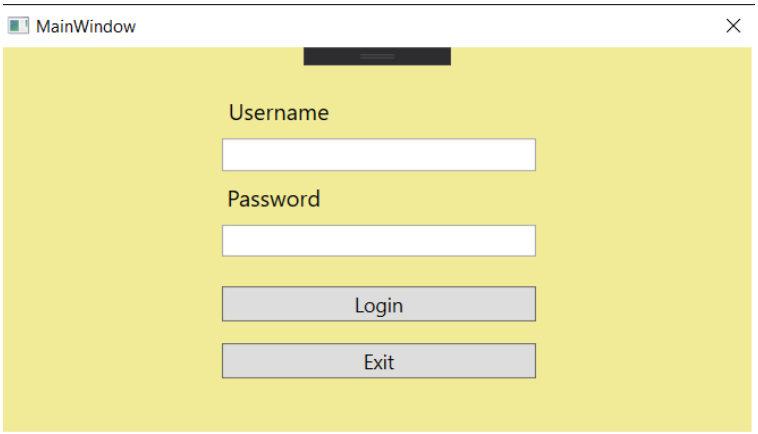

Figure 1 Login Screen

The above screen is the initial screen when a user operates the system. Valid username and password will only provide access to the system. For this project, the username and password is set to “admin”.

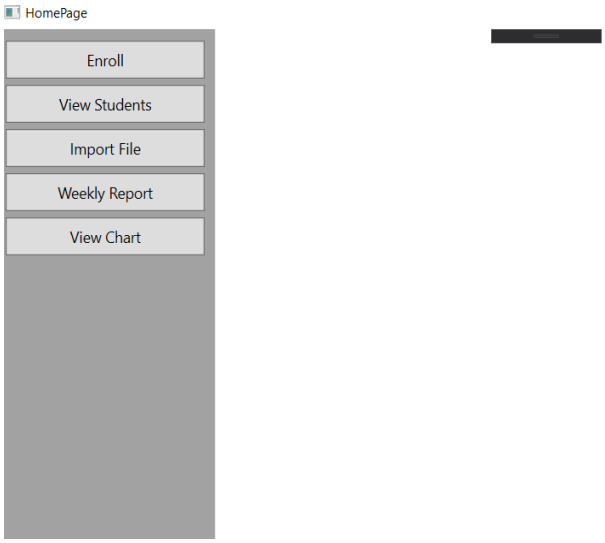

Figure 2 Home Page Screen

This is the main screen of the system. This screen contains all the features in the form of button. When clicked, all of them performs their respective functionality. “Enroll” and “Weekly Report” windows are displayed on the same window as homepage whereas others will open in a separate window.

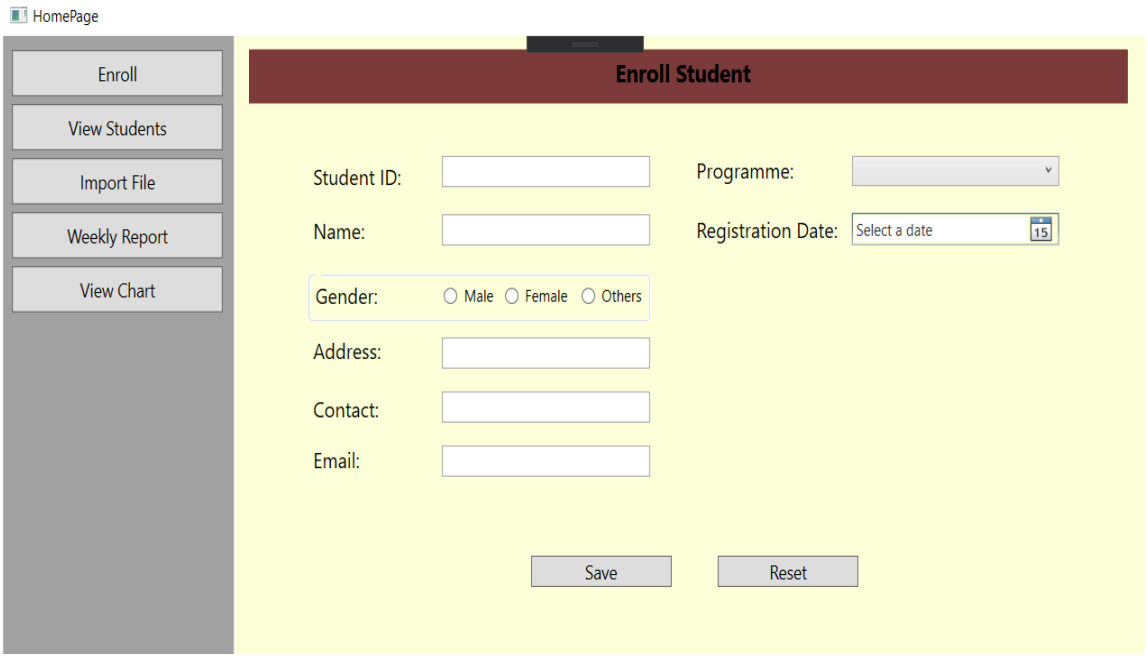

Figure 3 Enroll Screen

When “Enroll” button is clicked, the above window will be displayed on the same home page. As seen above, this window contains enroll form for new students. User can input all the details and then save it. The data is saved in a .csv format and also xml serialization is used to generate XML file. The reset button will clear all the data if the user decides not to save the record or to refill up the form. Proper validation is also used for this form. The textbox for contact field only takes integer as input.

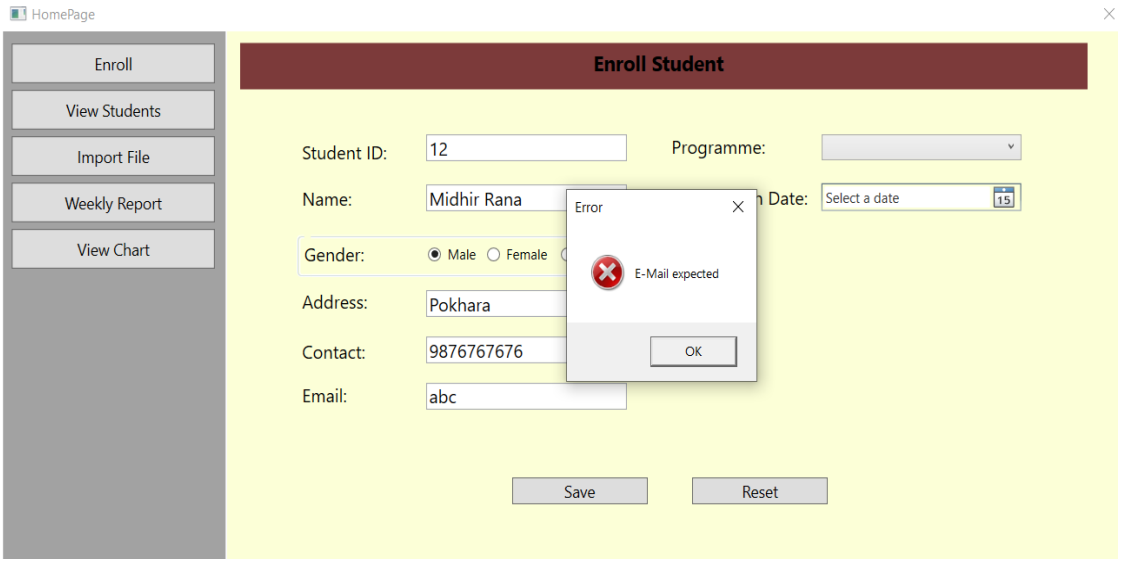

Figure 4 Email Validation

Valid Email address must be provided, otherwise a pop up window will appear as shown in the above figure.

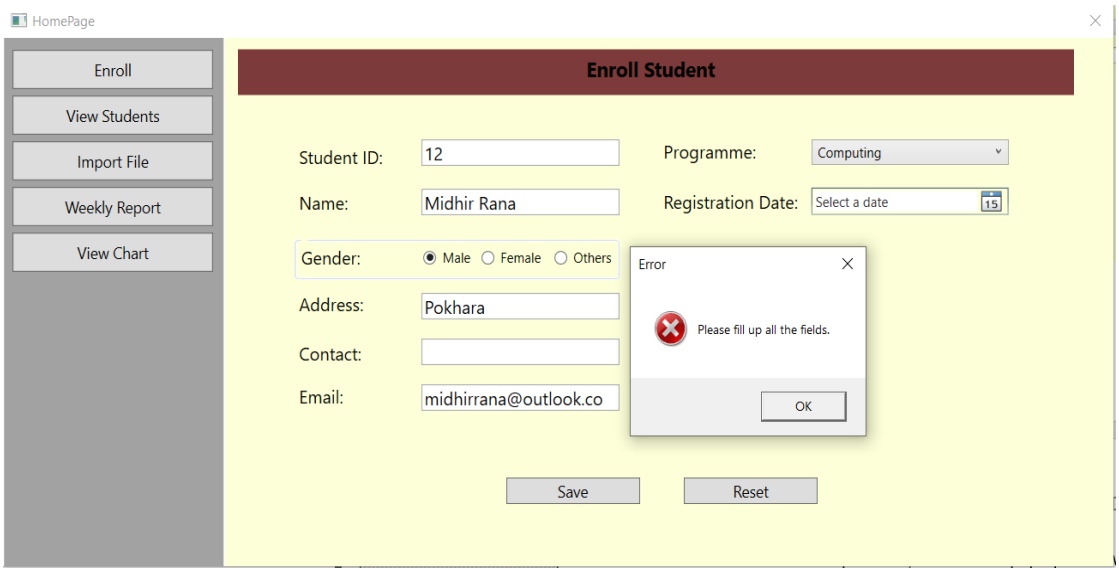

Figure 5 Validation for left out field

If any of the fields of the form is left out then a pop up window will appear giving message to fill up all the fields. Since this is a student information system, it might be mandatory to fill up all the fields.

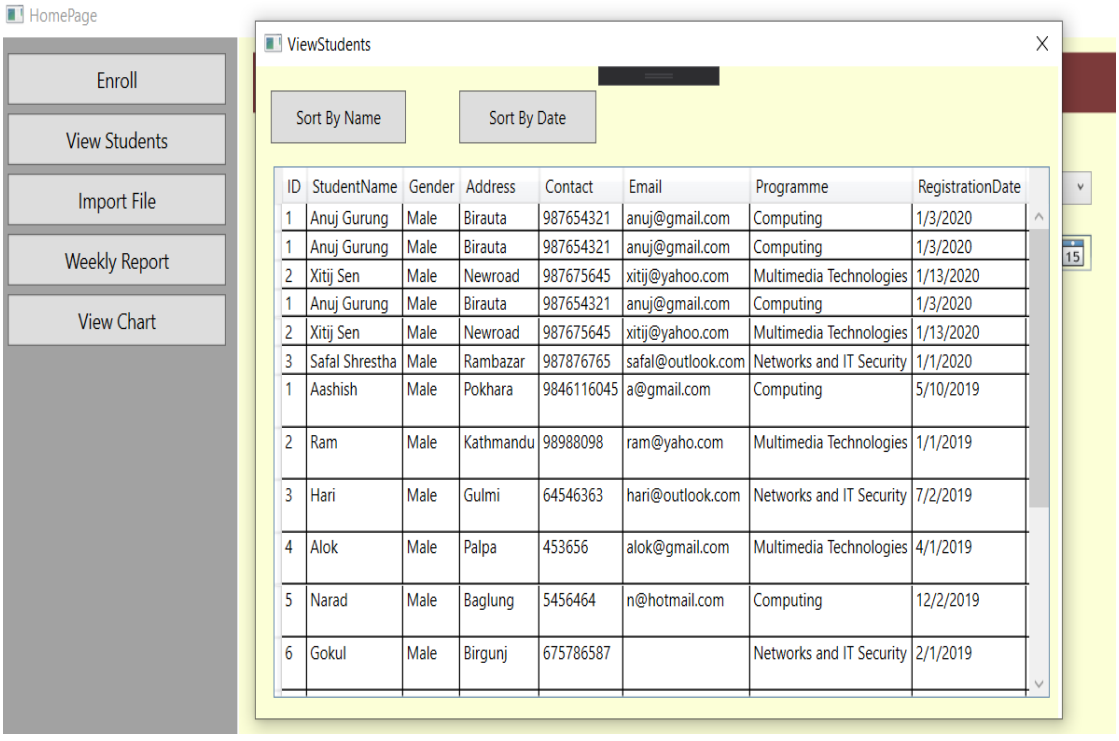

Figure 6 View Students Screen

The above figure shows all the students that had been enrolled. This window appears whenever the button “View Students” is clicked from the home page window. The data is fetched from the .csv file which is created while enrolling students. The “View Students” window has feature of sorting data on the basis of names and registered date.

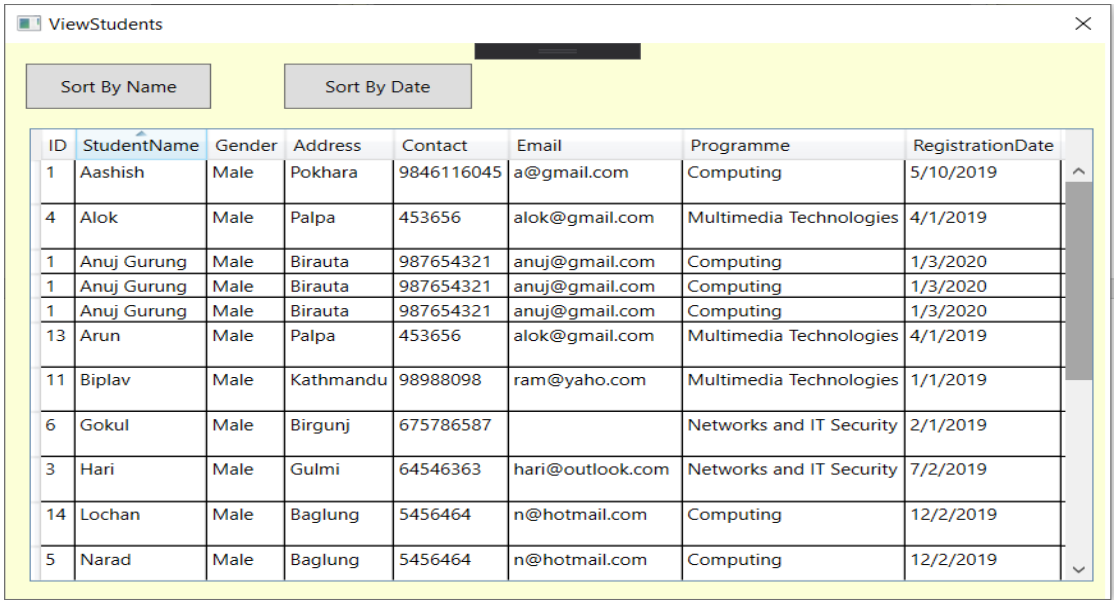

Figure 7 Sorted data by Name

The above figure shows that the data have been sorted according to the names.

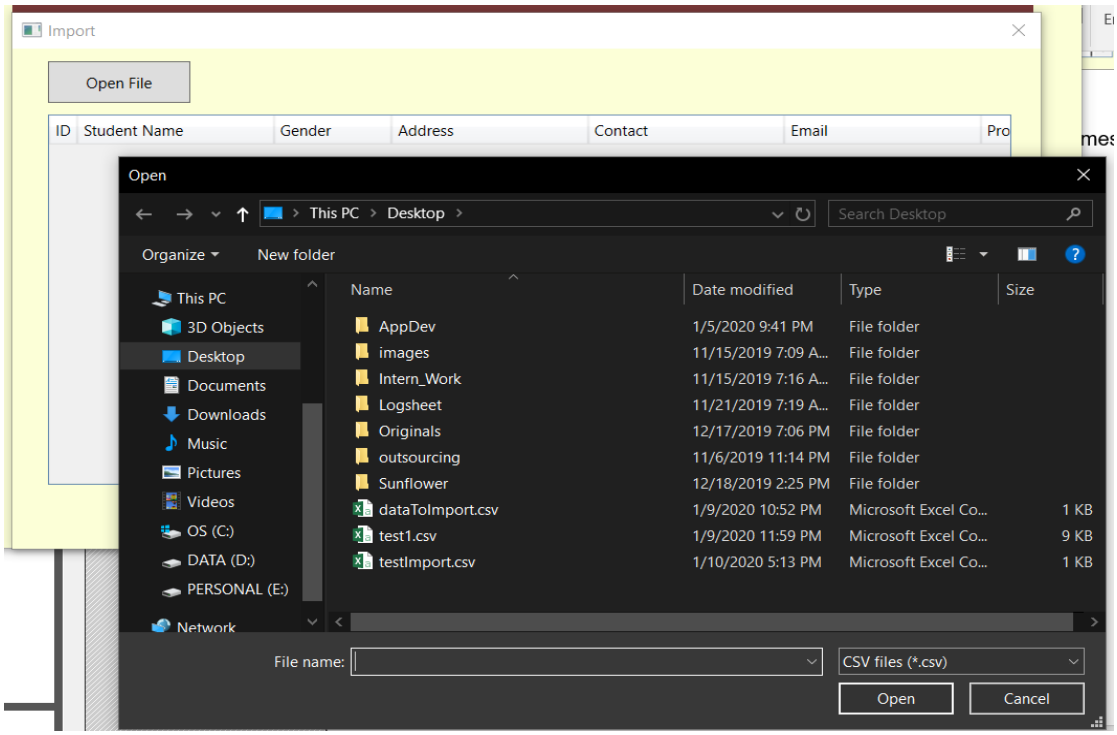

Figure 8 Import File

When clicked the button “Import file”, a new import window is opened, from where the user can import .csv file. This only reads .csv format files. Filter method is used in the programming to filter out the non-compatible files.

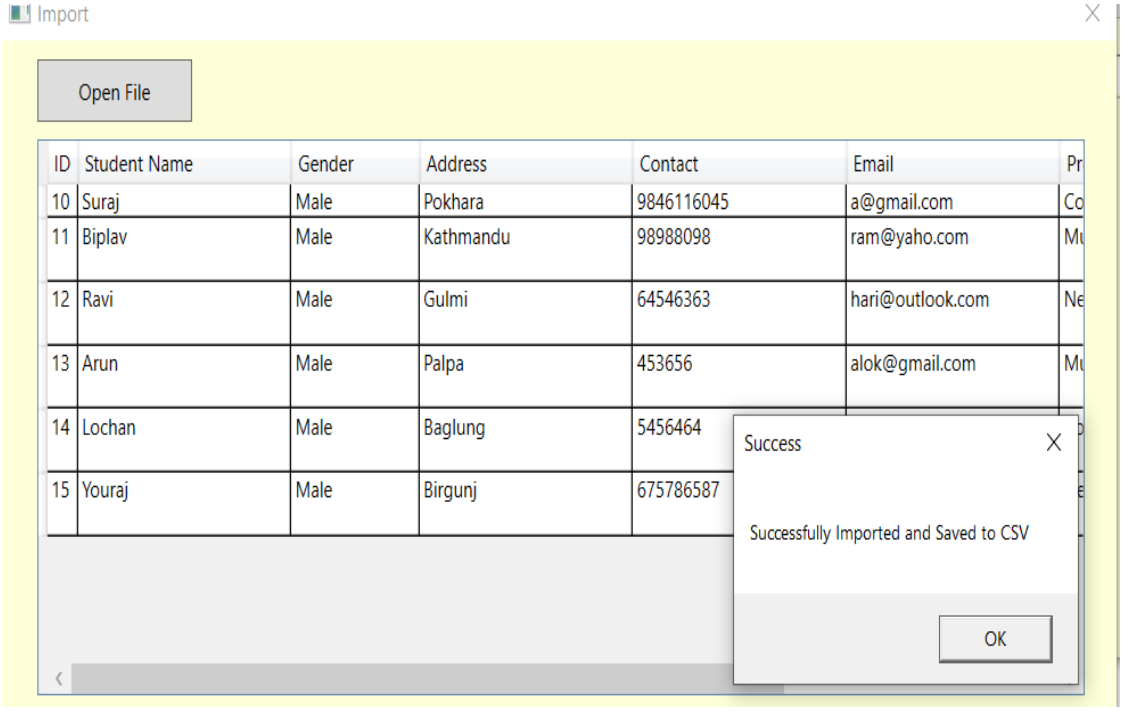

Figure 9 Imported .csv file successfully

The above figure shows that the .csv file has been imported successfully. The imported data is displayed on the datagrid under proper headings.

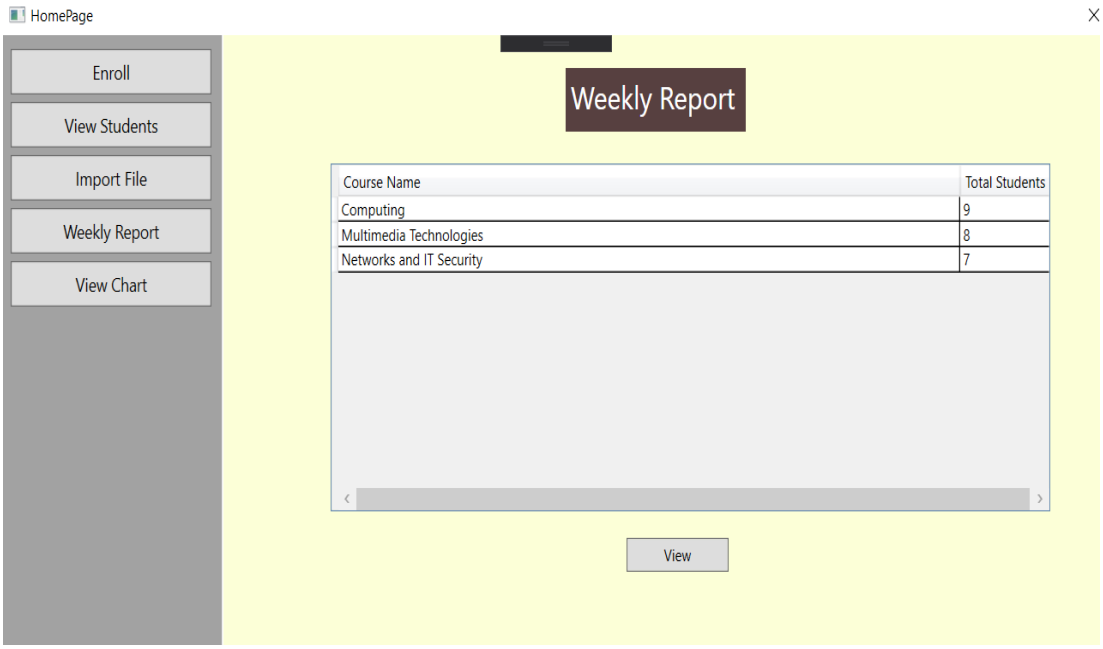

Figure 10 Weekly Report Screen

The “Weekly Report” screen shows the total number of students under the courses they had been enrolled on. When the button “Weekly Report” on the homepage is clicked, a new window is opened displaying a datagrid with headings “Course Name” and “Total students”. After clicking “View” button, the system retrieves data form the .csv file and counts the number of data with similar courses and then displays on the datagrid. The system compares the given course with the course on the .csv file under the column of course/programme and then counts the total number of students on that specific course. This process is done for each courses.

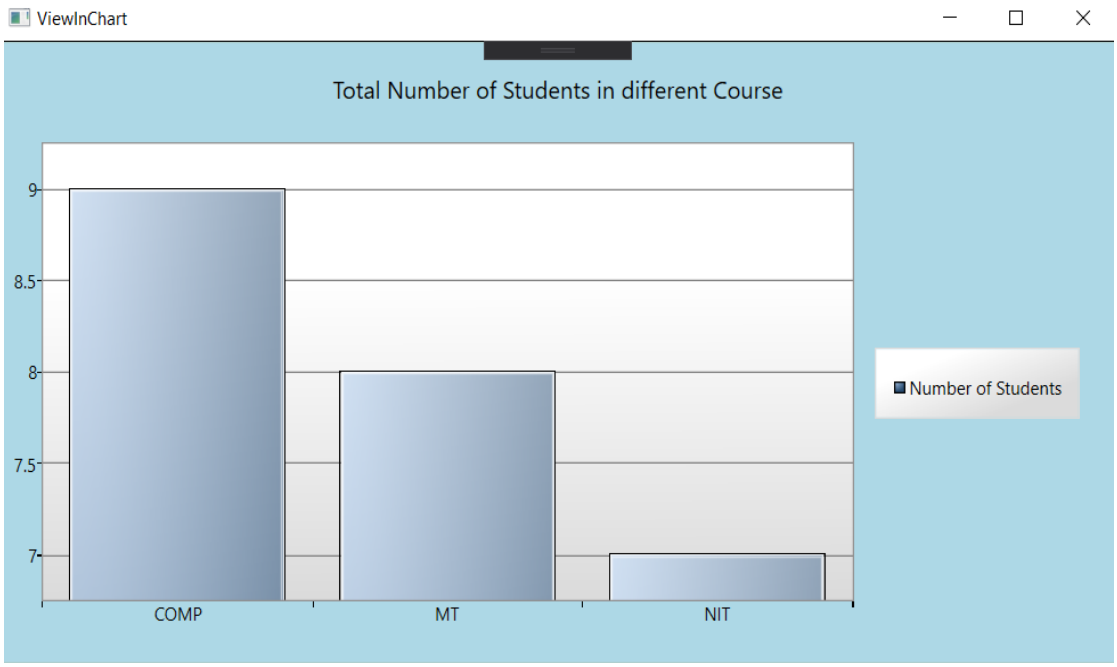

Figure 11 View Chart Screen

User can view the weekly report on a bar graph. It can be obtained by clicking on the “View Chart” button on the home page.

3. Software Architecture

In simple words, software architecture is the process of converting software characteristics such as flexibility, scalability, feasibility, reusability, and security into a structured solution that meets the technical and the business expectations. This definition leads us to ask about the characteristics of a software that can affect a software architecture design. Software characteristics describe the requirements and the expectations of a software in operational and technical levels. (Aladdin, 2018)

Below is the software architecture diagram of the developed system.

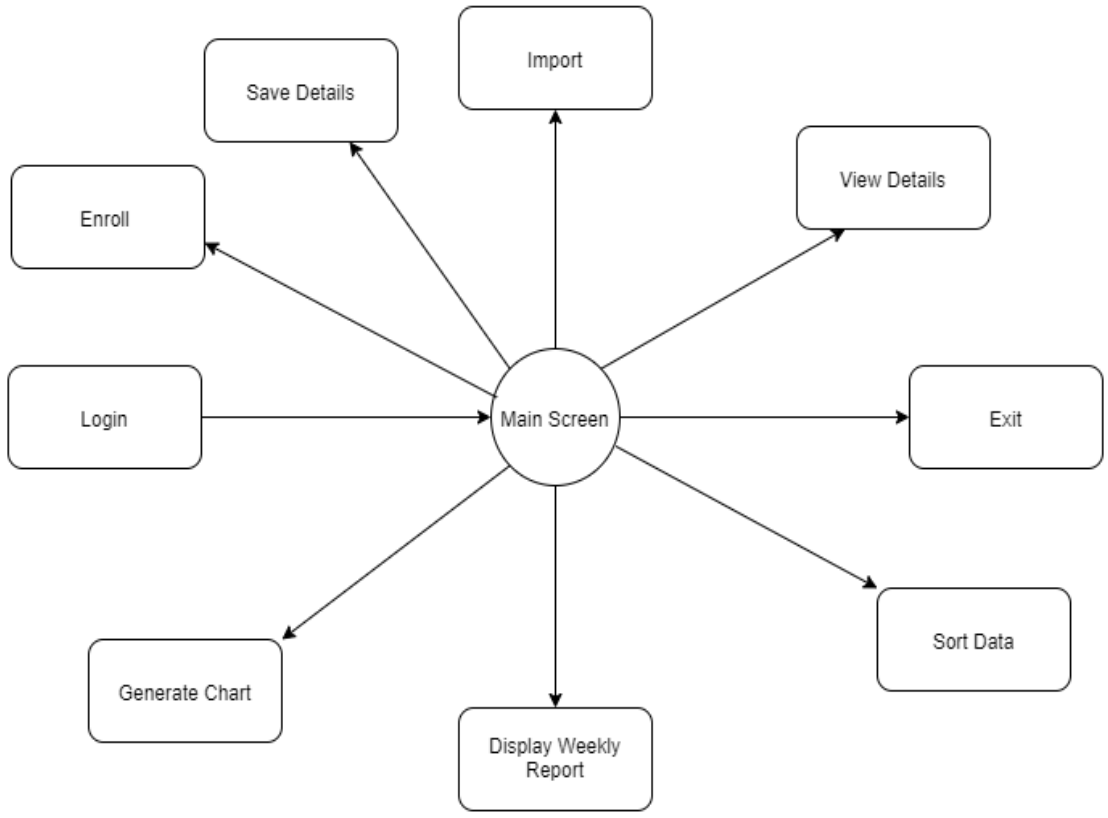

Figure 12 Software Architecture Diagram of the developed system

The above figure shows the fundamental functions of the system. User can log in to the system and get into the main screen, from where the user can choose various features of the system to perform. The system will then operate accordingly. After the purpose is done, user can exit the program and all the data will be saved.

4. Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram. (tutorialspoint, 2019)

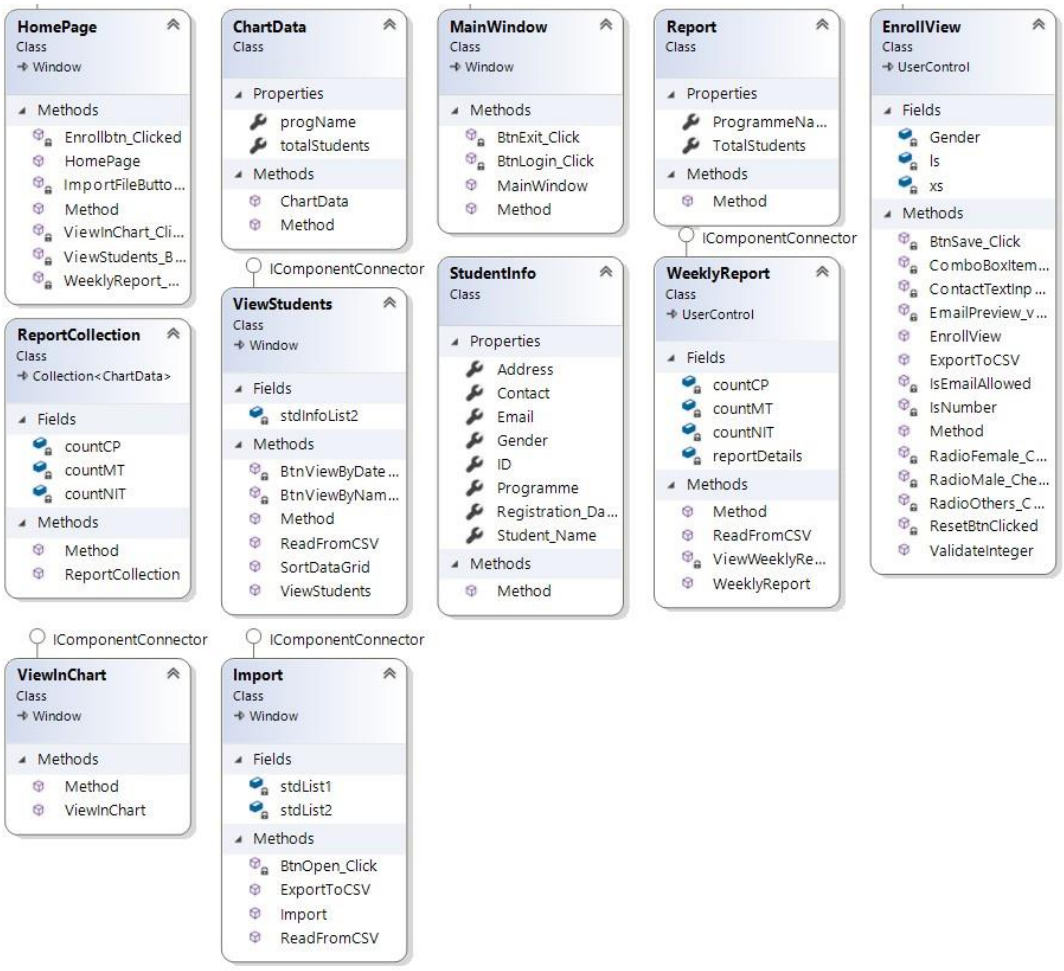

Figure 13 Class diagram of the developed system

4.1 Classes and Methods

There are various classes, properties and methods that have been created for the development of this project. The class diagram has illustrated all of them in general. Here, those classes, properties and methods are well described in detail. The methods which is used by the system are listed and well explained below.

Class

A class is a user-defined blueprint or prototype from which objects are created. Basically, a class combines the fields and methods (member function which defines actions) into a single unit. In C#, classes support the polymorphism, inheritance and also provide the concept of derived classes and base classes. (Mangal, n.d.)

1. **HomePage, MainWindow, ViewInChart, Import, ViewStudents**
   These all are classes of type Window class which holds the necessary methods, properties and variables to interact between a user and a standalone application. This type of class provides the ability to create, configure, show, and manage the lifetime of both windows and dialog boxes.
2. **ChartData**
   This class has a constructor and two properties which are progName and totalStudents.
3. **Report**
   This class has two properties which are ProgrammeName and TotalStudents.
4. **ReportCollection**
   This class has a constructor which opens a file, reads the data and counts the number of students that has been enrolled on each courses. Here the results are shared to another class called CharData by creating an object of that class.
5. **StudentInfo**
   This class has the properties essential for the student enrolment.

6. **WeeklyReport, EnrollView**

   These two are the class type of User control class. In the UserControl class, is the concept of grouping markup and code into a reusable container, so that the same interface, with the same functionality, can be used in several different places. The need of creating this type of class is to utilize the model, view concept in order to display the respective contents on same main window. These classes holds the necessary methods and variables which are explained below.

Method

A method is a code block that contains a series of statements. A program causes the statements to be executed by calling the method and specifying any required method arguments. Methods are declared by specifying the access level such as public or private, optional modifiers such as abstract or sealed, the return value, the name of the method, and any method parameters. These parts together are the signature of the method. (Microsoft, 2015)

1. **BtnSave_Click**

   This method will first check if all the fields of the enrol form is filled up or not. If not, a pop up window will appear with an appropriate message. Otherwise, this method retrieves data from all the textbox and save it to new .csv file and also in XML format. For saving in .csv format it calls another method that is, ExportToCSV. Here, XML serialization is used. And, if a .csv file has been created already then the data will be appended on the same file.

2. **ResetBtnClicked**

   This method provides the facility of clearing the textbox. This will reset the form as if it was just started. It might be useful if the user inserts incorrect data or decides not to enrol.

3. **ISNumber**

   This method has a return type of bool. It is created to check whether a user has inserted an integer value or not. It has string variable as a parameter.

4. **ValidateInteger**

   This method validates whether the data in the textbox is integer or not. It calls another method for this process, that method is “ISNumber”. If the output from the ISNumber method is false then this method will not let the user to input the current key. This method only lets user to input integer values.

5. **ContactTextInput_keypress**

   This method is one of the event handler method for contact textbox where it calls ValidateInteger method to validate if the user is giving an integer value or not.

6. **RadioMale_Checked**

   This method assigns the value Male to the global variable Gender whenever the radio button for Male is selected from the enrol form.

7. **RadioFemale_Checked**

   This method assigns the value Female to the global variable Gender whenever the radio button for Female is selected from the enrol form..

8. **RadioOthers_Checked**

   This method assigns the value Others to the global variable Gender whenever the radio button for Others is selected from the enrol form.

9. **ExportToCSV**

   This method actually saves the students data in .csv file format. BtnSave_Click calls this method passing two arguments of type List and string. Here, TextWriter is used to create output and StreamWriter is used to read file location.

10. **EmailPreview_validate**

    This method is one of the event handler method, which checks if the user has given a valid email address or not. This method calls another method for validation. If the return value from that method is false then it notifies user to enter correct email address.

11. **ISEmailAllowed**

    This method has a return type of bool. It takes string as argument. It checks the valid format of an email address and then returns true or false as Boolean value.

12. **Enrollbtn_Clicked**

    This method opens the Enrol form when user clicks the button “Enroll” from the homepage or main screen.

13. **ViewStudents_ButtonClicked**

    This method opens new window of View Students when user clicks the button “View Students” from the homepage or main screen.

14. **ImportFileButton_Clicked**

    This method opens new window of Import when user clicks the button “Import File” from the homepage or main screen.

15. **WeeklyReport_keyAction**

    This method opens the Weekly Report window when user clicks the button “Weekly Report” from the homepage or main screen.

16. **ViewInChart_Clicked**

    This method opens the Chart window when user clicks the button “View Chart” from the homepage or main screen.

17. **BtnOpen_Click**

    This method opens a dialog box to let users to import .csv file. It filters the files comparing with .csv format and only shows compatible files. It then calls ReadAllText method of the System.IO.File library which takes the data from the .csv file and calls another method ReadFromCSV to convert the data into string and then save it to the user’s .csv file.

18. **ReadFromCSV**

    This method takes string as an argument, reads the data from .csv file and converts the datas into string then saves it by calling another method that is ExportToCSV.

19. **BtnViewByName_Click**

    This methods sorts the data from the datagrid on the basis of name of the students. The process is done by calling another method called SortDataGrid.

20. **SortDataGrid**

    This method takes two arguments of type Datagrid and integer. It sorts the data that are displayed on the datagrid on the basis of column, and the integer value that has been passed which specifies the column index.

21. BtnViewByDate_Click

This methods sorts the data from the datagrid on the basis of registration date of the student enrollment. The process is done by calling another method called SortDataGrid.

22. BtnExit_Click

This method is called whenever user clicks the button “Exit” from the login panel or the main window. When clicked, system will ask the user if they want to exit the program, if clicked yes then it will close the login panel.

23. BtnLogin_Click

This method is called whenever user clicks the button “Login” from the login panel or the main window. When clicked, user will get into the system and may proceed.

24. ViewWeeklyReport_Click

This method when called reads the data form the available .csv file where all the details of students are saved. It retrieves the data and then count the number of students on each programmes and then displays the result on the datagrid.

5. Flowchart

A flowchart is a formalized graphic representation of a logic sequence, work or manufacturing process, organization chart, or similar formalized structure. The purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process. Flowcharts use simple geometric symbols and arrows to define relationships. (Rouse, 2008)

As per this coursework, we are required to draw a flowchart for the student enrolment process only. Below is the flowchart for the process.

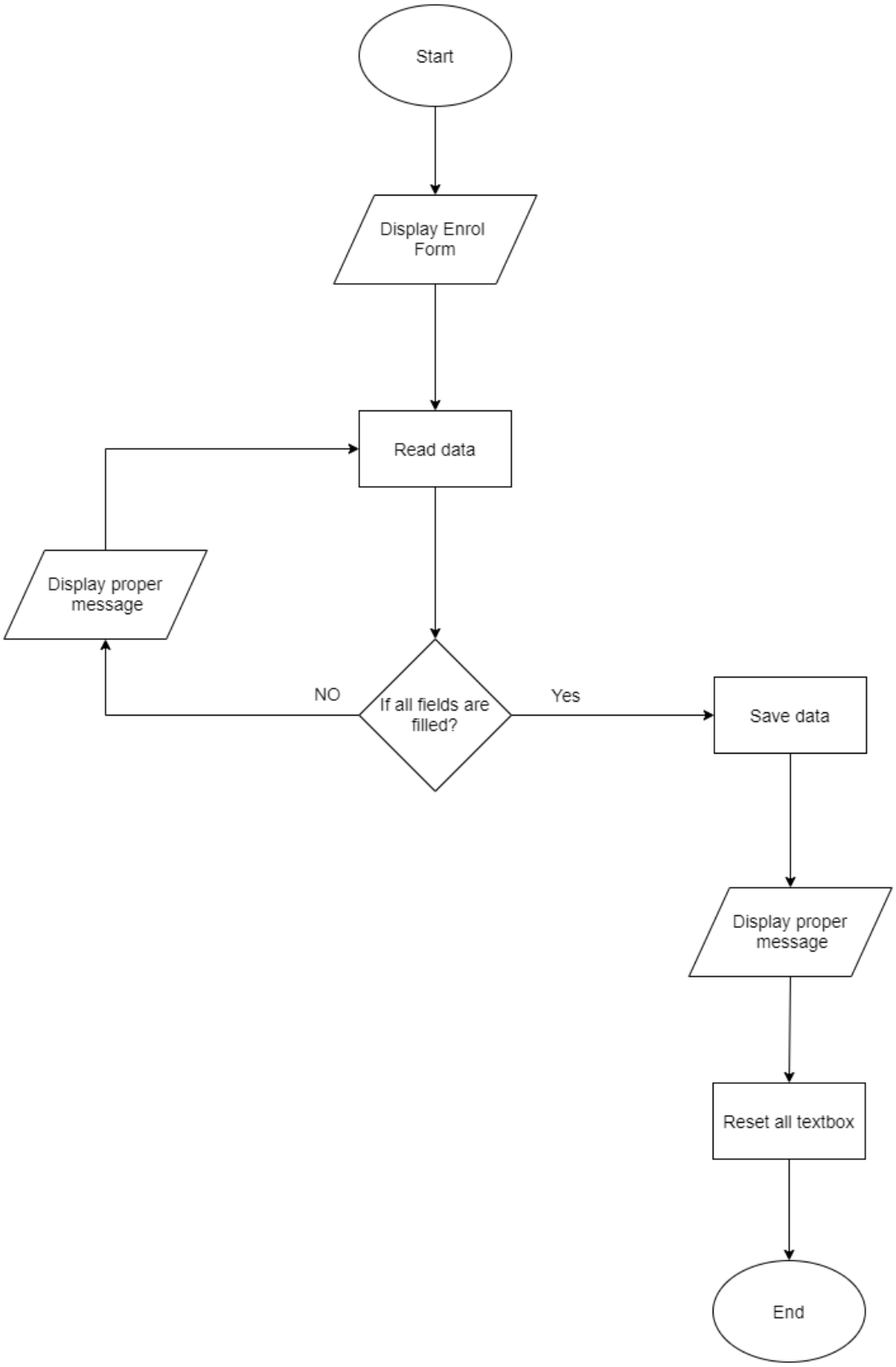

Figure 14 Flowchart for enrolment process

6. Data Structure

Data Structure is a way of collecting and organising data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage. It represents the knowledge of data to be organized in memory. It should be designed and implemented in such a way that it reduces the complexity and increases the efficiency. (studytonight.com, n.d.)

Among the various data structures, List<T> is used for this coursework. A list is an abstract data type. It is a collection of heterogeneous elements and supports a specific bunch of operations. Size of a list also grows dynamically whereas size of other data structure like Array, remains static throughout the program. As size of list can increase or decrease at run time so there is no memory wastage. The list is also better for frequent insertion and deletion.

7. Conclusion

The course work assigned to us has finally come to an end. Lots of studies and researches were done on related topics. The research and the studies really helped a lot. I have put much effort on this project and it has helped me to understand more about the basic and the fundamental aspects of developing Desktop Application.

This coursework was very interesting and fun to do. There were some challenges that we encountered, but solved with the help of fellow classmates. Different concept and ideas were shared with the teachers and classmates to achieve the completion of this project. The most challenging part of this coursework was to work with the files, saving and accessing the files took a lot of time to understand. A lot of research had to be done for this part only. Similarly, XML serialization was also bit confusing. One of the error that we encountered was repeating of root node while inserting a new data. But, it was handled later. Minor run time errors were also encountered, which were removed after going through the codes thoroughly. Proper methods and functionality as per required by the coursework are mainly focused. Validation is implemented on enrol form for better and accurate data collection. Suitable methods and classes have been utilized for the simplicity of codes. Design aspects are also focused for this system. Proper login panel with validation and appropriate windows for each functionalities are developed.

While doing this coursework I learnt to use Visual Studio, learnt C# programming language. The use of WPF forms has made the concept of .xaml and .xaml.cs clear to me. Learnt about the window class, user control class and also adding reference for generating chart. This was indeed a new experience for me. From this project I got an opportunity to build a desktop application. Overall this project helped me a lot and progressed me in the IT field. I am very interested in further coursework and aim to do better.

References

Aladdin, M. (2018, July 27). *Software Architecture - The Difference Between Architecture and Design*. Retrieved from codeburst.io: https://codeburst.io/software-architecture-the-difference-between-architecture-and-design-7936abdd5830

Mangal, K. (n.d.). *C# Class and Object*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/c-sharp-class-and-object/

Microsoft. (2015, July 20). *Methods (C# Programming Guide)*. Retrieved from docs.microsoft.com/: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/methods

Rouse, M. (2008, August). *What is flowchart?* Retrieved from whatis.techtarget.com/: https://whatis.techtarget.com/definition/flowchart

studytonight.com. (n.d.). *Introduction to Data Structures and Algorithms*. Retrieved from studytonight: https://www.studytonight.com/data-structures/introduction-to-data-structures

tutorialspoint. (2019). *UML - Class Diagram*. Retrieved from www.tutorialspoint.com: https://www.tutorialspoint.com/uml/uml_class_diagram.htm