Informatics College Pokhara

Application Development

CS6004NP

Coursework 1

Submitted By:
Student Name: Sujata Regmi
London Met ID: 17030753
Group: L3C1
Date: 10-Jan-2020

Submitted To:
Mr. Ishwor Sapkota

Table of Contents

Table of Figure

1. Introduction

The project is all about Student Information System. The system is highly designed developed and test under various circumstances. The system has full functionality and is capable of working as the design shows. The features and functions that are required by Student Information System are almost fulfilled by the developed system. This system has a login page and main window. In main window there is home page where raw data are saved. It can make reports of daily and weekly visits. There are many more functionalities of the program and are mentioned in the full documentation.

1.1. Current Scenario

It is a technological era but in context of Nepal many companies use old traditional system are used i.e. Paper-Based System. They do not a digital system to store data and their system is completely outdated. Companies are not able to grasp the technology and handle them.

1.2. Purposed System

This it is the age of science and technological but honestly many companies do not use a digital system to store data. Because of this, this system is proposed to digitize and purposed to overcome problem mentioned above. It has login section so to access the main screen login is required and it’s secured. 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.

When running the program

- Click the ‘AppDevCoursewrk.sln’ file inside the folder ‘AppDevCoursewrk’.

Login Screen

As the end user operates the system the initial screen will be the security screen.

- The username and password of the system is “admin”. Only a valid username and password can provide access to the system.

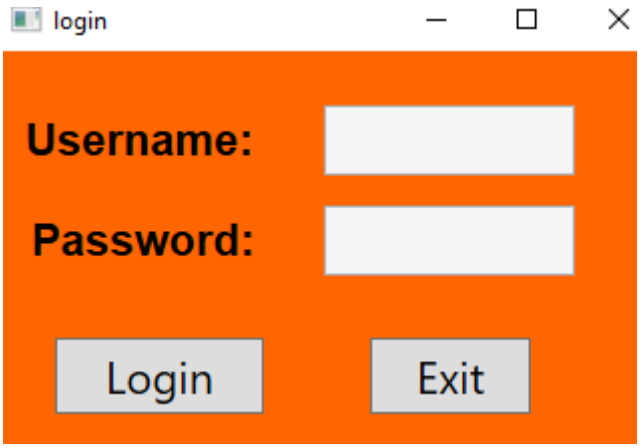


Figure 1: Login Screen

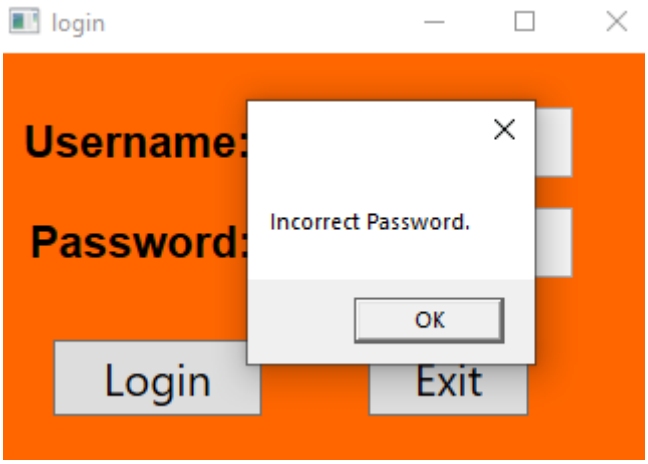


Figure 2: Login Failed

After login screen shows, user has to provide login username and password to access into main window menu.

- If user has provided incorrect username and password, then error message will be displayed.

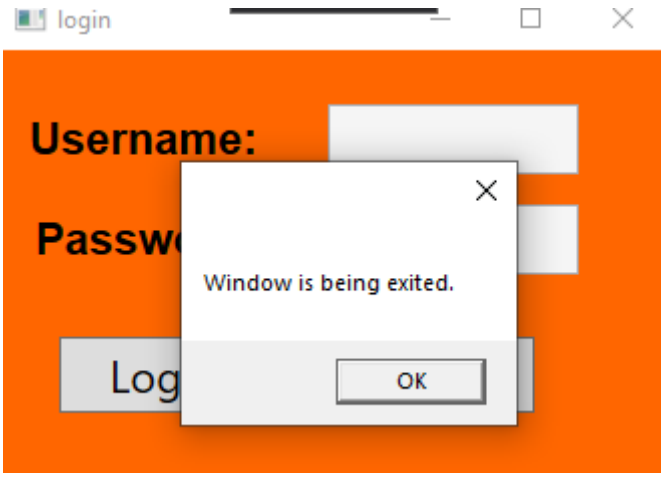


Figure 3: Exit

- When a user click the exit button then the program gets stop n window gets closed.

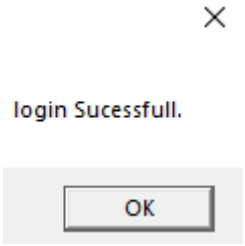


Figure 4: login successful

After login screen shows, user has to provide login username and password to access into main window menu. If user has provided incorrect username and password, then error message will be displayed.

Main Page

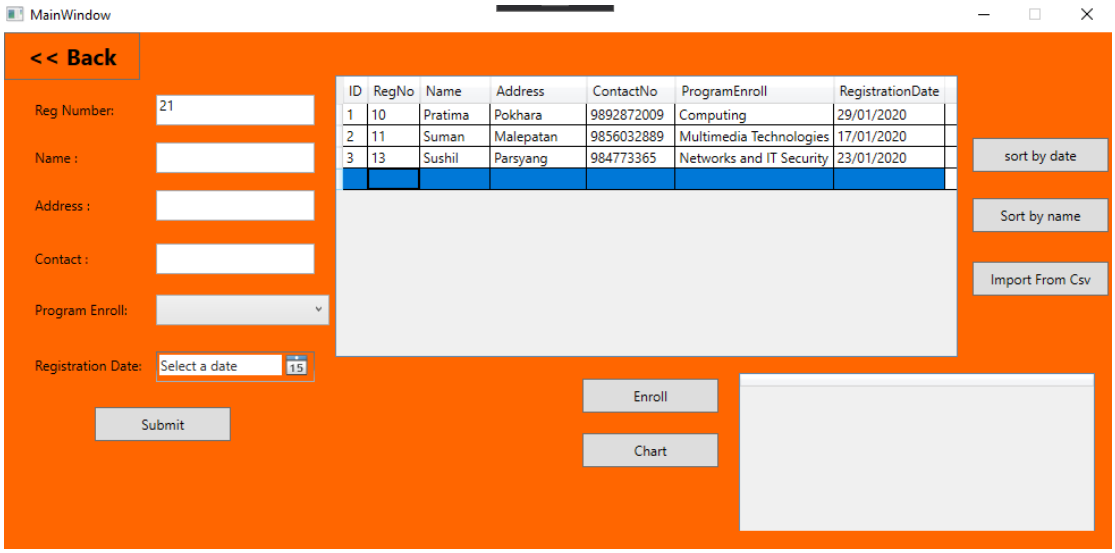


Figure 5: Homepage

After logging into the system with correct credentials the main screen provides Student Details Entry form, Student Detail grid, enrol section, Report section etc.

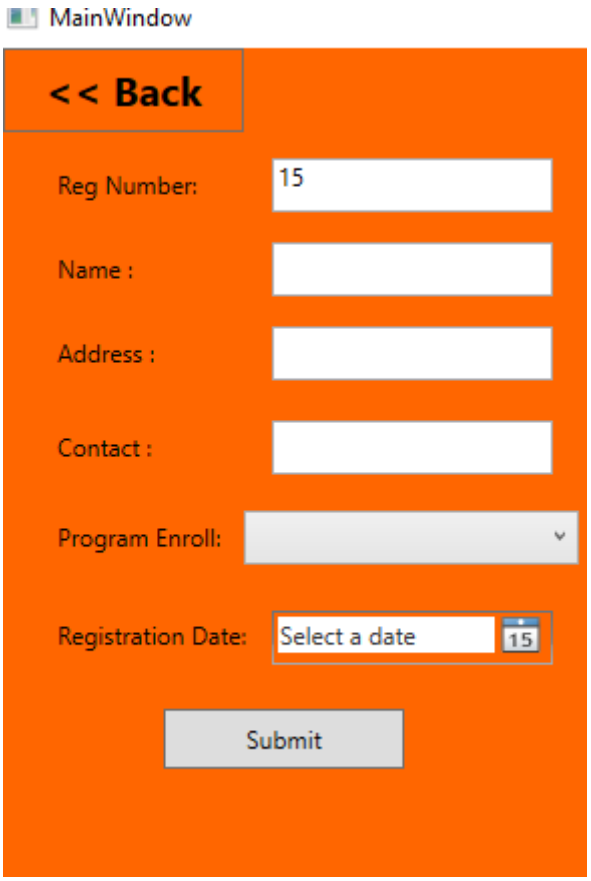


Figure 6: Student Register Form

To register a student, this form should be filled. If user try to pass empty value on any field like name, address, contact, program enrol and registration date it won't allow it to register student. Reg num is auto generate.

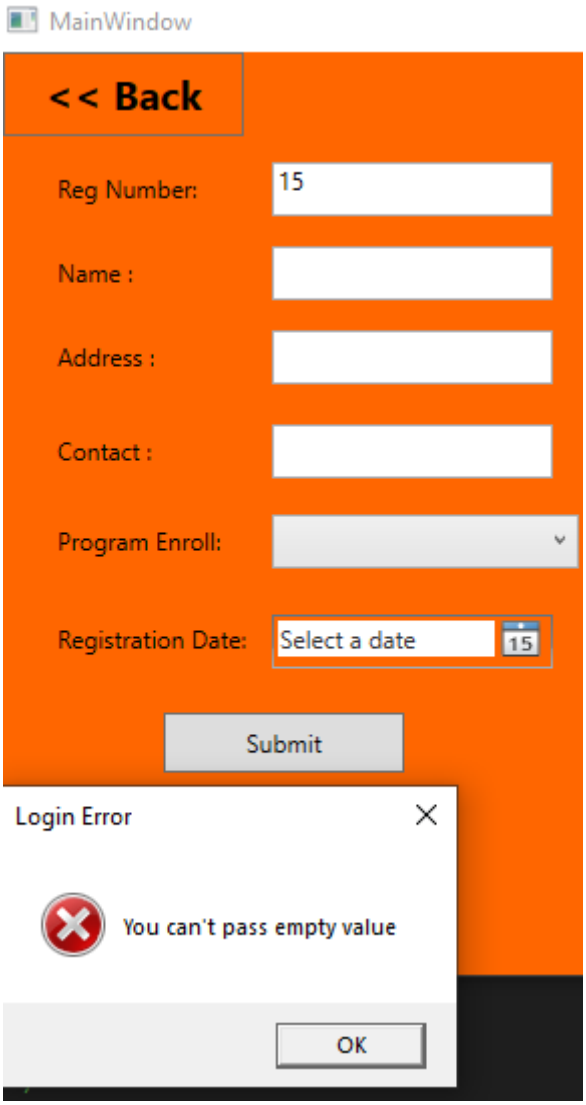


Figure 7: Passing empty value

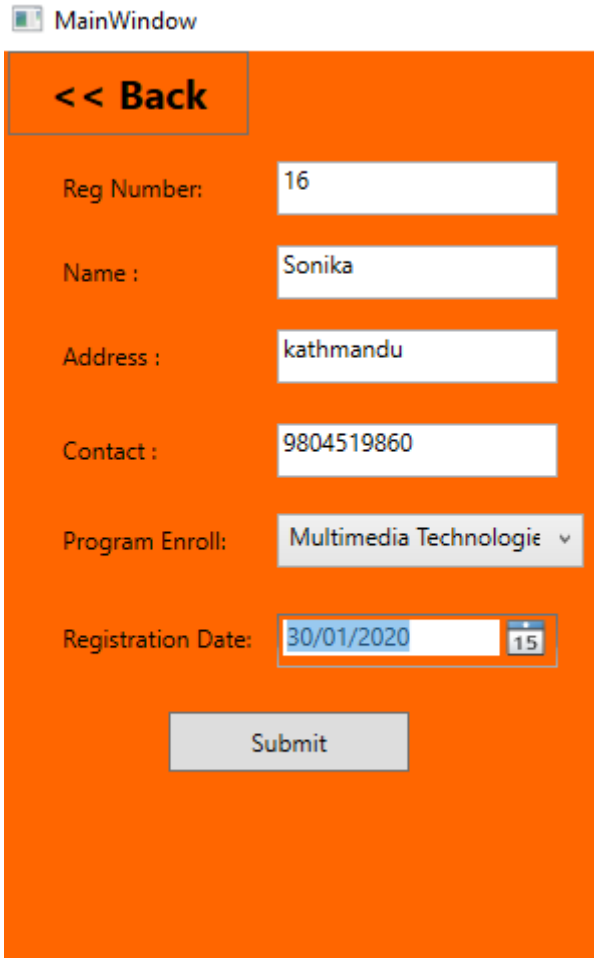


Figure 8: Registration

To register a student all the fields should be filled and after filling user should click on submit button .After that student got registered.

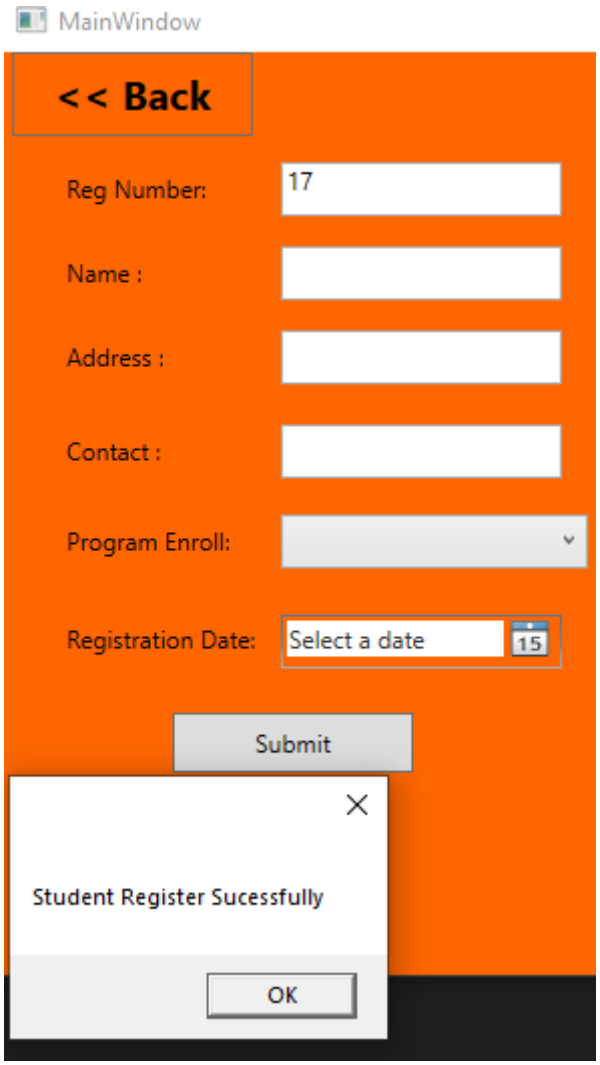


Figure 9: Registered Successfully

In this screenshot it's shown after clicking submit button it is showing message that students have been registered successfully. After that all text field got cleared except registration number because it is auto generated.

ID	RegNo	Name	Address	ContactNo	ProgramEnroll	RegistrationDate	
1	10	Pratima	Pokhara	9892872009	Computing	29/01/2020	
2	11	Suman	Malepatan	9856032889	Multimedia Technologies	17/01/2020	
3	13	Sushil	Parsyang	984773365	Networks and IT Security	23/01/2020	
4	14	Sujata	phulbari	9856032496	Multimedia Technologies	25/12/2019	
5	15	Divya	Birauta	9804599887	Networks and IT Security	28/12/2019	
6	16	Sonika	kathmandu	9804519860	Multimedia Technologies	30/01/2020	

Figure 10: Displaying all student

All the student who are registered are saved into xml. From xml the student details are displayed in grid from xml.

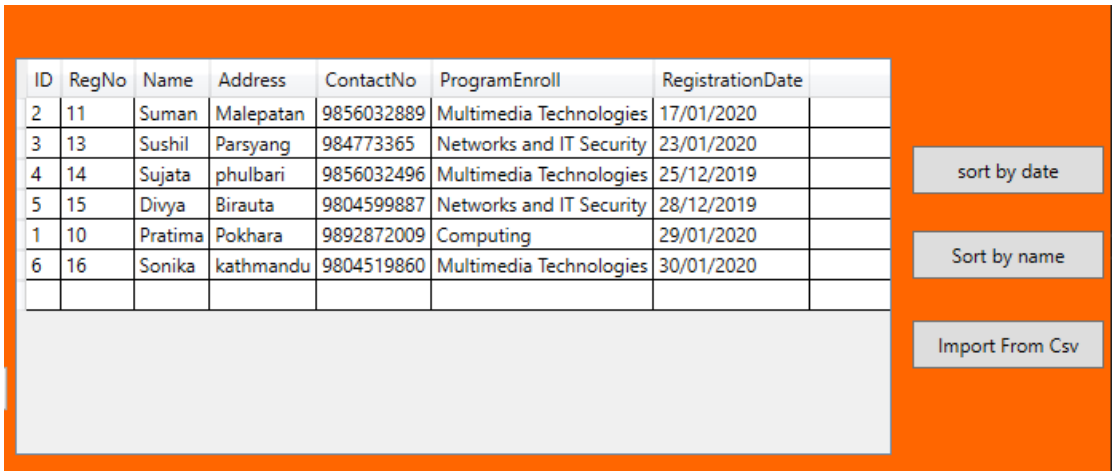


Figure 11: Sorting By date

The data in grid can be sorted. In this figure the student data are been sorted by date on click of the button “Sort by date”.

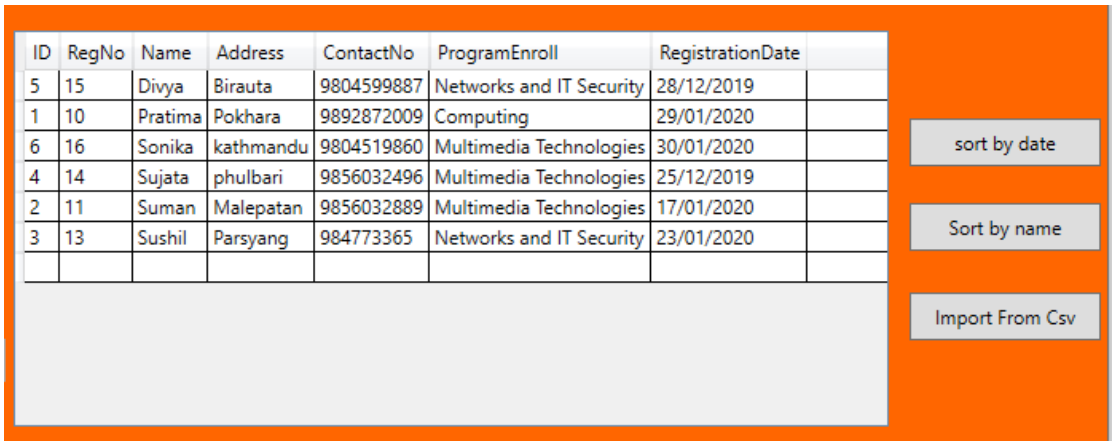


Figure 12: Sort by Name

In this figure the student data are been sorted by name on click of the button “Sort by date”.

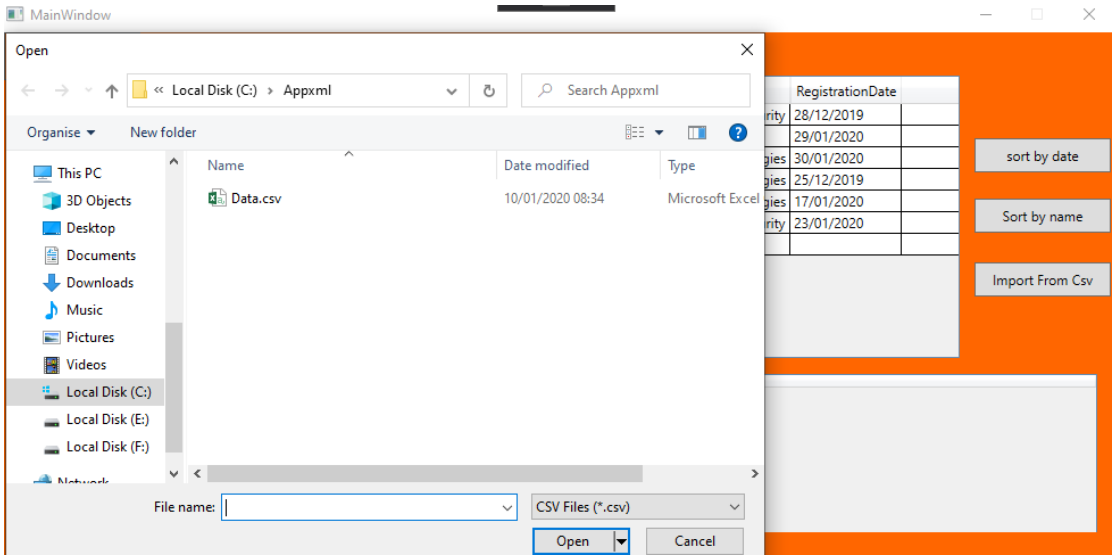


Figure 13: On click of button Import from Csv

In this figure when the button “Import From Csv” is clicked then it opens a dialogue box from where a user can only select “.csv” file.

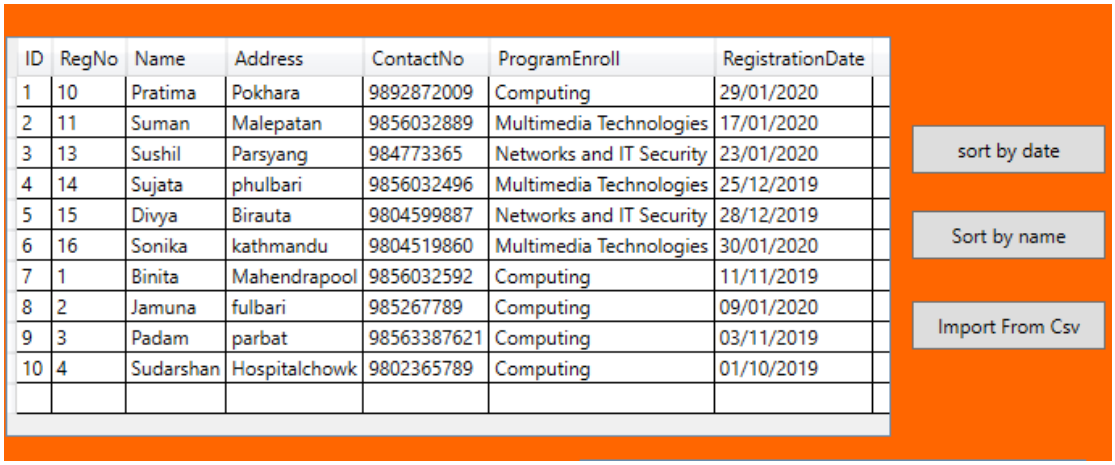


Figure 14: Displaying Data From Csv

In this figure when the csv file is selected it display data from csv with data available on xml. After data imported from Csv it display in grid n saves it in XML.

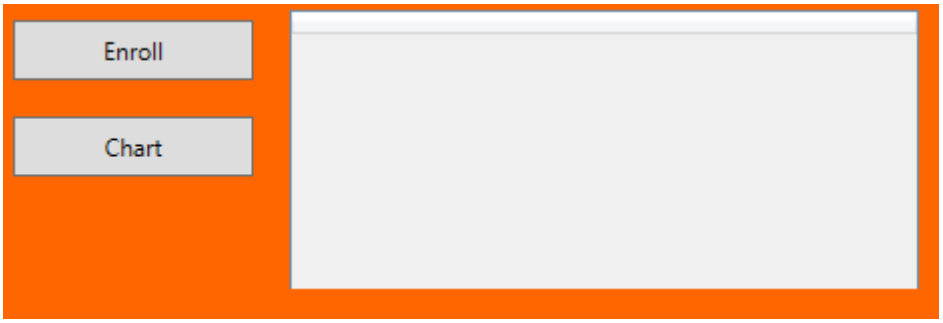


Figure 15: Grid for Enroll

This is the Grid where the students' ennoblement along with program are displayed.

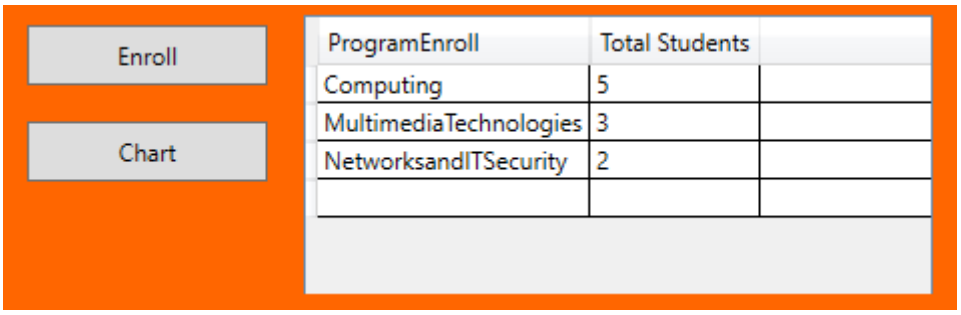


Figure 16: Enroll status

After the button "Enroll" is clicked then it show all the program enrol with total student enrolled in the grid.

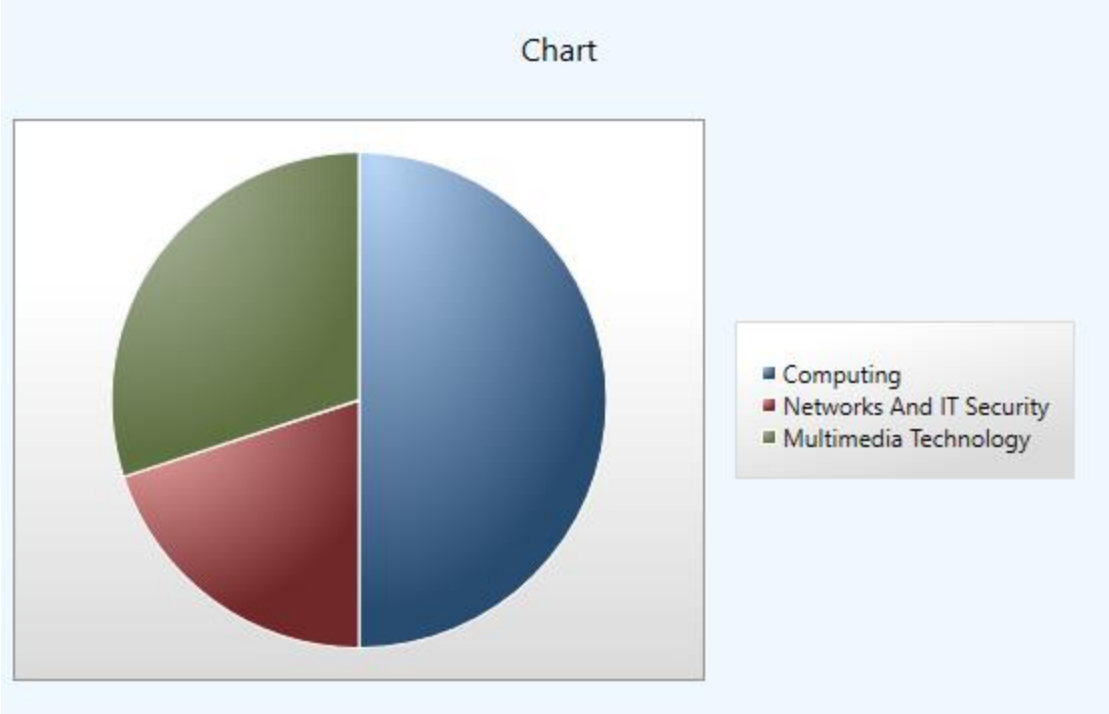


Figure 17: Chart

After the button “Chart” is clicked then it show data in pie chart.

Schema Generate

```
<?xml version="1.0" standalone="yes"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Course">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ID" msdata:AutoIncrement="true" msdata:AutoIncrementSeed="1" type="xs:int" />
              <xs:element name="Name" type="xs:string" minOccurs="0" />
              <xs:element name="DisplayText" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Student">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ID" msdata:AutoIncrement="true" msdata:AutoIncrementSeed="1" type="xs:int" />
              <xs:element name="Name" type="xs:string" minOccurs="0" />
              <xs:element name="Address" type="xs:string" minOccurs="0" />
              <xs:element name="ContactNo" type="xs:string" minOccurs="0" />
              <xs:element name="ProgramEnroll" type="xs:string" minOccurs="0" />
              <xs:element name="RegistrationDate" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="StudentReport">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ID" msdata:AutoIncrement="true" msdata:AutoIncrementSeed="1" type="xs:int" minOccurs="0" />
              <xs:element name="RegNo" type="xs:string" minOccurs="0" />
              <xs:element name="Name" type="xs:string" minOccurs="0" />
              <xs:element name="Address" type="xs:string" minOccurs="0" />
              <xs:element name="ContactNo" type="xs:string" minOccurs="0" />
              <xs:element name="ProgramEnroll" type="xs:string" minOccurs="0" />
              <xs:element name="RegistrationDate" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
<
```

Figure 18: Generated Schema

This is the generated schema file of Student Management System.

XML

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <StudentReport>
    <ID>1</ID>
    <RegNo>10</RegNo>
    <Name>Pratima</Name>
    <Address>Pokhara</Address>
    <ContactNo>9892872009</ContactNo>
    <ProgramEnroll>Computing</ProgramEnroll>
    <RegistrationDate>29/01/2020</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>2</ID>
    <RegNo>11</RegNo>
    <Name>Suman</Name>
    <Address>Malepatan</Address>
    <ContactNo>9856032889</ContactNo>
    <ProgramEnroll>Multimedia Technologies</ProgramEnroll>
    <RegistrationDate>17/01/2020</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>3</ID>
    <RegNo>13</RegNo>
    <Name>Sushil</Name>
    <Address>Parsyang</Address>
    <ContactNo>984773365</ContactNo>
    <ProgramEnroll>Networks and IT Security</ProgramEnroll>
    <RegistrationDate>23/01/2020</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>4</ID>
    <RegNo>14</RegNo>
    <Name>Sujata</Name>
    <Address>phulbari</Address>
    <ContactNo>9856032496</ContactNo>
    <ProgramEnroll>Multimedia Technologies</ProgramEnroll>
    <RegistrationDate>25/12/2019</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>5</ID>
    <RegNo>15</RegNo>
    <Name>Divya</Name>
    <Address>Birauta</Address>
    <ContactNo>9804599887</ContactNo>
    <ProgramEnroll>Networks and IT Security</ProgramEnroll>
    <RegistrationDate>28/12/2019</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>6</ID>
```

Figure 19: Generated XML 1

```
  <StudentReport>
    <ID>6</ID>
    <RegNo>16</RegNo>
    <Name>Sonika</Name>
    <Address>kathmandu</Address>
    <ContactNo>9804519860</ContactNo>
    <ProgramEnroll>Multimedia Technologies</ProgramEnroll>
    <RegistrationDate>30/01/2020</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>7</ID>
    <RegNo>1</RegNo>
    <Name>Binita</Name>
    <Address>Mahendrapool</Address>
    <ContactNo>9856032592</ContactNo>
    <ProgramEnroll>Computing</ProgramEnroll>
    <RegistrationDate>11/11/2019</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>8</ID>
    <RegNo>2</RegNo>
    <Name>Jamuna</Name>
    <Address>fulbari</Address>
    <ContactNo>985267789</ContactNo>
    <ProgramEnroll>Computing</ProgramEnroll>
    <RegistrationDate>09/01/2020</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>9</ID>
    <RegNo>3</RegNo>
    <Name>Padam</Name>
    <Address>parbat</Address>
    <ContactNo>98563387621</ContactNo>
    <ProgramEnroll>Computing</ProgramEnroll>
    <RegistrationDate>03/11/2019</RegistrationDate>
  </StudentReport>
  <StudentReport>
    <ID>10</ID>
    <RegNo>4</RegNo>
    <Name>Sudarshan</Name>
    <Address>Hospitalchowk</Address>
    <ContactNo>9802365789</ContactNo>
    <ProgramEnroll>Computing</ProgramEnroll>
    <RegistrationDate>01/10/2019</RegistrationDate>
  </StudentReport>

</NewDataSet>
```

Figure 20: Generated Schema 2

When student is registered then it saves in XML file. When data of csv file is imported in grid then it saves in XML. This figure is all student which who are registered and data that are imported from Csv.

3. Journal Articles

1. Creativity of student information system projects: From the perspective of network embeddedness.

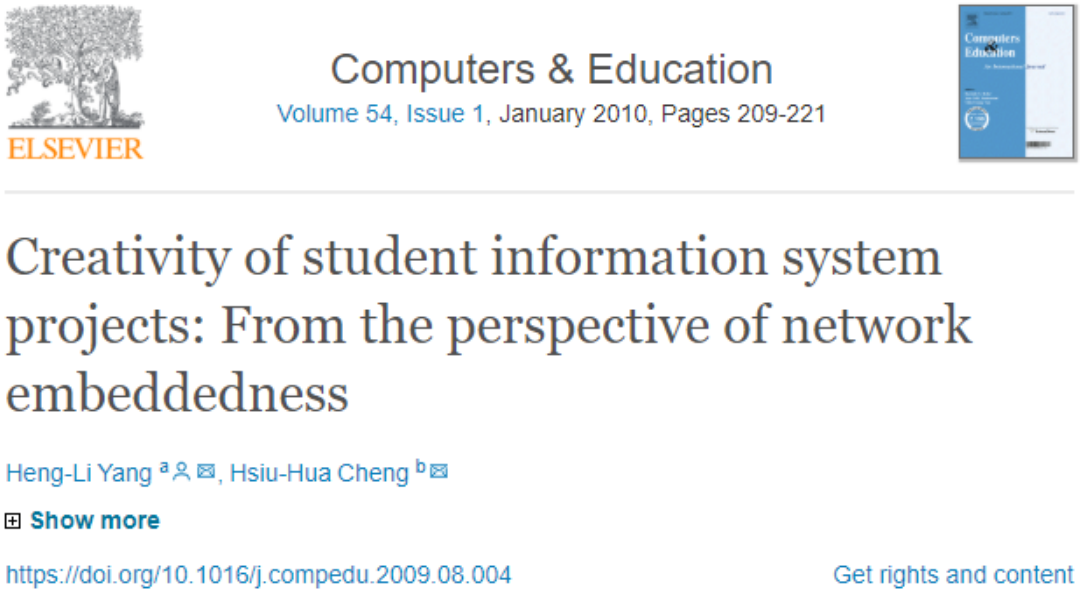
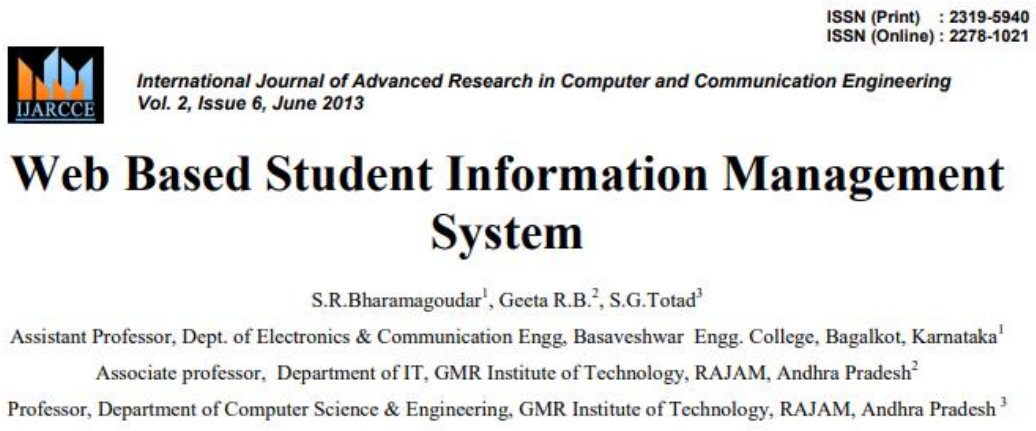


Figure 21: Creativity of Student Information System

In this journal it is saying, many companies have pursued innovation to obtain a competitive edge. Thus, educational reform focuses commonly on training innovative students. This find out about adopted the idea of an affiliate community of tasks to check out how task embeddedness influences project group creativity. This work surveys 60 projects in a Management Information Systems Department of a University. The validity of the unique study hypotheses is examined with the aid of the usage of moderate hierarchical regression evaluation to decide how task embeddedness influences venture crew creativity and check how the group innovation climate moderates the relationships between project embeddedness and project crew creativity. Analytical outcomes indicate a fine affiliation between structural embeddedness

and project crew creativity, a negative relationship between positional embeddedness and project crew creativity, and a positive impact of team innovation climate on the relationships between community embeddedness and project crew creativity (Yang & Cheng, 2010).

2. Web Based Student Information Management System


ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

Web Based Student Information Management System

S.R.Bharamagoudar[1], Geeta R.B.[2], S.G.Totad[3]
Assistant Professor, Dept. of Electronics & Communication Engg, Basaveshwar Engg. College, Bagalkot, Karnataka[1]
Associate professor, Department of IT, GMR Institute of Technology, RAJAM, Andhra Pradesh[2]
Professor, Department of Computer Science & Engineering, GMR Institute of Technology, RAJAM, Andhra Pradesh [3]

Figure 22: Web Based Student Information Management System

In this journal it is saying that Student Information Management System (SIMS) provides an easy interface for the renovation of student information. It can be used through academic institutes or faculties to maintain the data of students easily. The creation and management of accurate, updated information regarding a students' tutorial career is seriously essential in the university as well as colleges. The student records system offers with all types of student details, academic-related reports college details, course details, curriculum, batch details, placement details, and different resource-related important points too. It tracks all the important points of a student from the day one to the end of the course which can be used for reporting purpose, tracking of attendance, progress in the course, completed semesters, years, coming semester 12 months curriculum details, exam details, project or any different project details, closing examination result and all these will be on hand thru a secure, on-line interface embedded in the college's website. It will also have college details, batch execution details, students' details in all aspects, the more than a few academic notifications to the team of workers and students updated through the university administration (Bharamagoudar, et al., 2013).

3. Student Information Systems: A Guide to Implementation Success

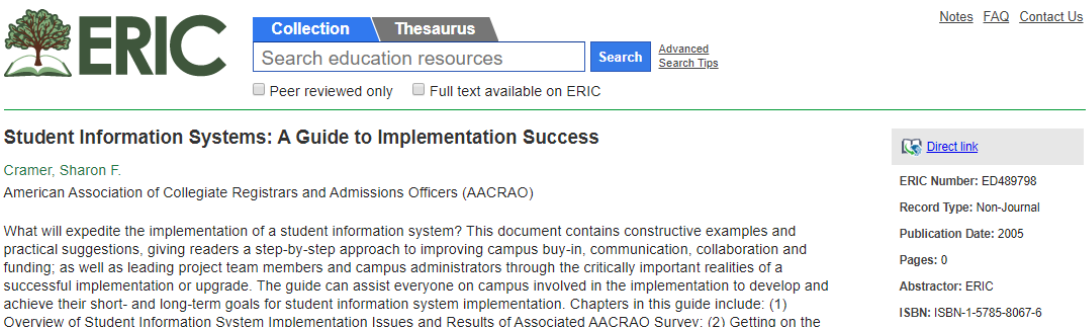


Figure 23: Student Information Systems

This journal contains constructive examples and realistic suggestions, giving readers a step-by-step approach to enhancing campus buy-in, communication, collaboration, and funding; as well as leading project group members and campus administrators through the critically important realities of successful implementation or upgrade. The information can help all people on campus involved in the implementation to develop and obtain their short- and long-term dreams for student information system implementation (Cramer, Sharon F., 2005).

4. System Architecture

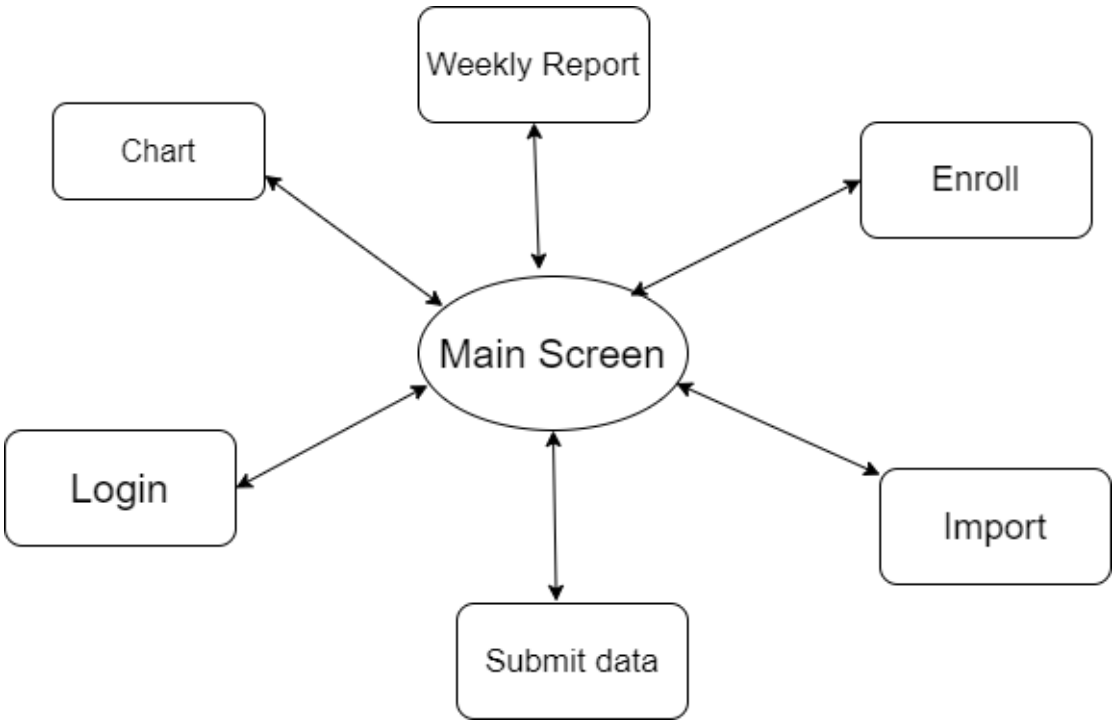


Figure 24: System Architecture

4.1. Class Diagram

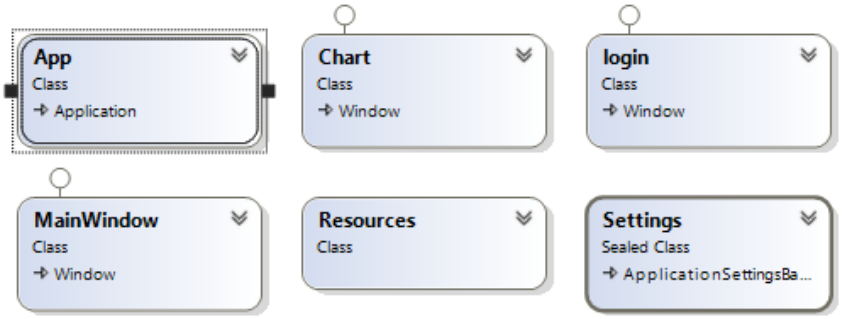


Figure 25: Class Diagram

4.2. Individual Diagram

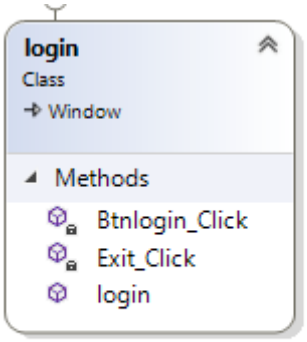


Figure 26: Login Class Diagram

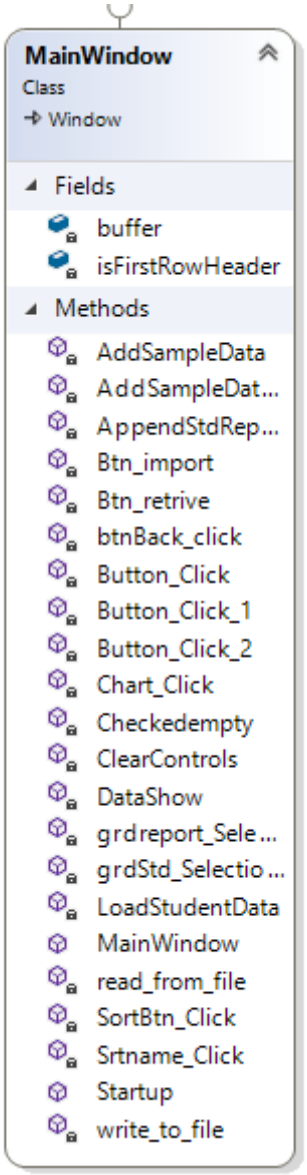


Figure 27: Main Window Class Diagram

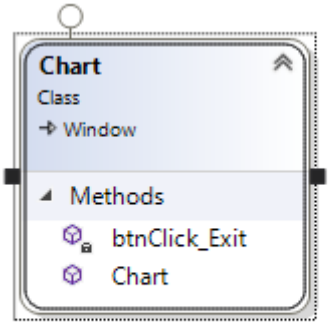


Figure 28: Chart Class Diagram

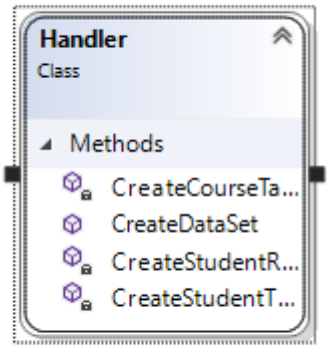


Figure 29: Handler Class Diagram

4.3. Flowchart and Algorithm for Reports

4.3.1. Student Enroll Flowchart

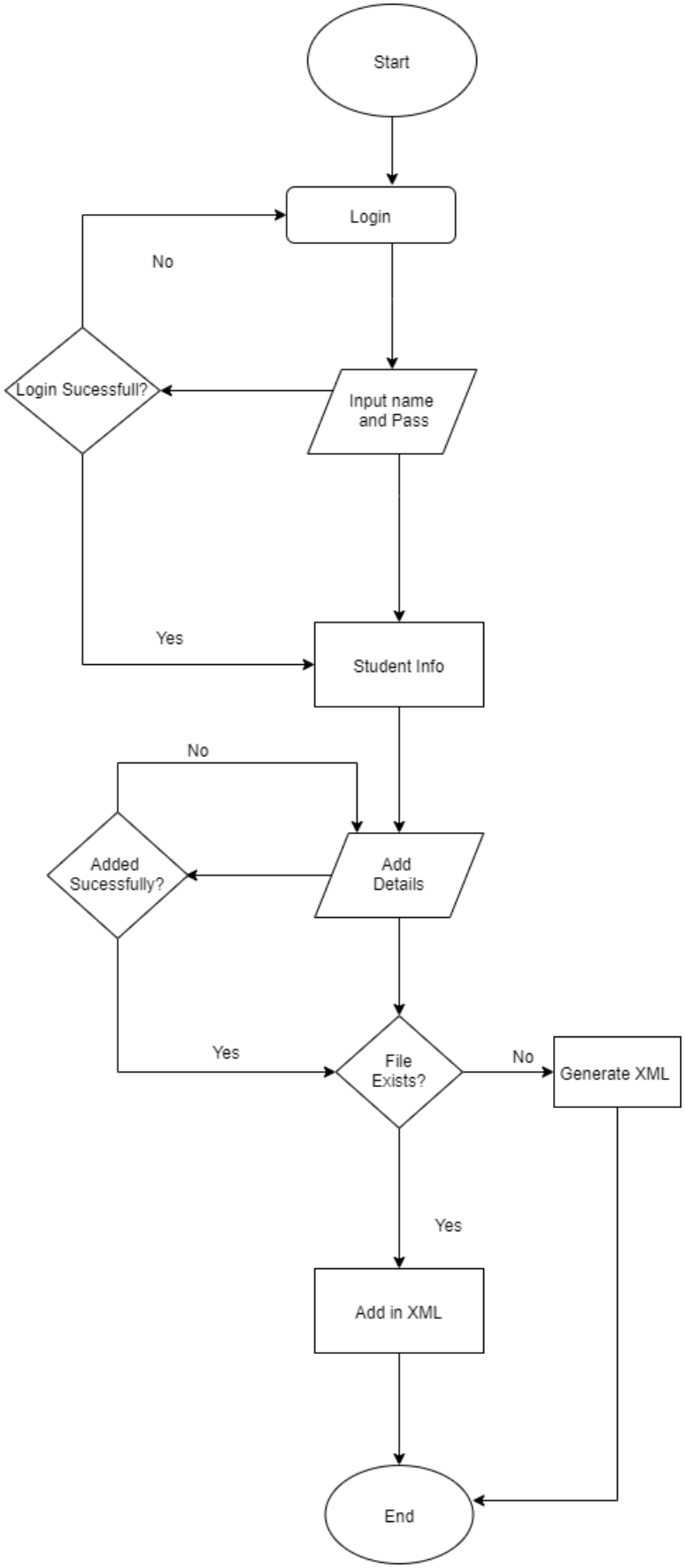


Figure 30: Flowchart of student enroll

4.3.2. Student Enroll Algorithm

1. Start
2. Login
3. Input Username and Password
4. Check whether username and password is correct or not.
5. If correct, then gather student information.
6. If incorrect then roll back to input login.
7. Add Student Details
8. If the details of student added successfully then go in next steps.
9. If unsuccessful then roll back to add student details.
10. Check whether the file already exists or not.
11. If file doesn’t exists then generate xml with details and End
12. If file already exists then add details in XML.
13. End.

5. Sorting Algorithm

The sorting Algorithm used in the Student information system is bubble sorting algorithm.

Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjoining elements is compared and the elements are swapped if they are not in order. It is not suitable for massive data units as its average and worst case complexity are of O(n2) where n is the number of items (Tutorials point, 2020).

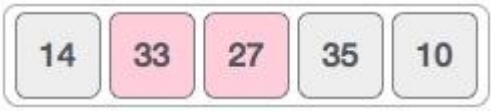
Working of Bubble Sort

We take an unsorted array for our example. Bubble sort takes O(n2) time so we're keeping it brief and precise.

This algorithm 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 are comparing 33 with 27.

27 is smaller than 33 and these two values will be swapped.



The new array look like this −

Next we compare 33 and 35. We find that both are in already sorted positions.

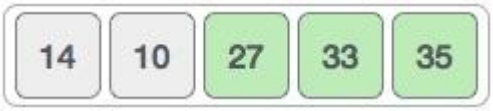
Then we move to the next two values, 35 and 10.

We know then that 10 is smaller 35. Hence they are not sorted.

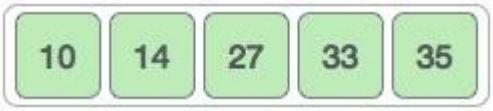
We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this −

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

After each iteration, at least 1 value moves at the end.



And when there's no swap required, bubble sorts learns that an array is completely sorted.



(Tutorials point, 2020)

6. Reflection

Developing the system in Microsoft Visual Studios 2019 maintaining C# as main programming language is new experience for me. But creating in C# surroundings is new for me. Developing an information system for student is actually a challenging challenge though. Serialization and deserialization are another new thing while growing the system. Though, developing new classes and techniques helps to tempo the improvement task. Importing and exporting of CSV file is also a new challenge and it surely assist me in gaining knowledge of file handling. Creating a class diagram within the visual studio helps me in documentation phase. With the growth of technology, the visual studio and its community helps beginner developer like us to pace our development speed.

7. Conclusion

The initial coursework for the module CS6004NP Application Development was to build successfully. It is made to store student information. It required a long time to build up the task in Visual Studio Enterprise 2019 utilizing C# in WPF. During the project work, all the data requirements, functional requirements, and design specifications were closely observed and described in this document. The document produced now is ready for the physical implementation of the software for persistent Student Information System. The document produced from Student Information system requirements collection and analysis is very useful and important for the application development. The framework has login screen to add security to the task. After login, the framework shows a Main screen where every functionalities are found. Aside from various shape components, class outline for every one of the structures and classes were utilized.

References

Bharamagoudar, S., R.B, G. & Totad, S., 2013. Web Based Student Information Management. *International Journal of Advanced Research in Computer and Communication Engineering,* 2(6).

Cramer, Sharon F., 2005. Student Information Systems: A Guide to Implementation Success. *ERIC,* Volume ED489798.

Tutorials point, 2020. *How Bubble Sort Works.* [Online] Available at: https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm

Yang, H.-. Y. & Cheng, H.-. H., 2010. Computer & Education. *ELSEVIER,* 54(1), pp. 209-221.