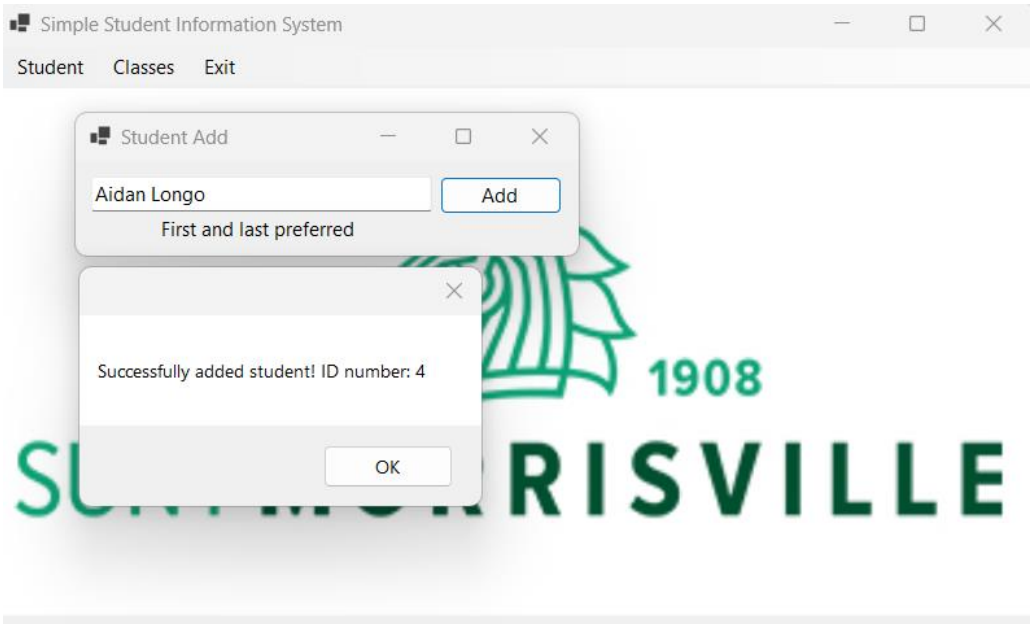
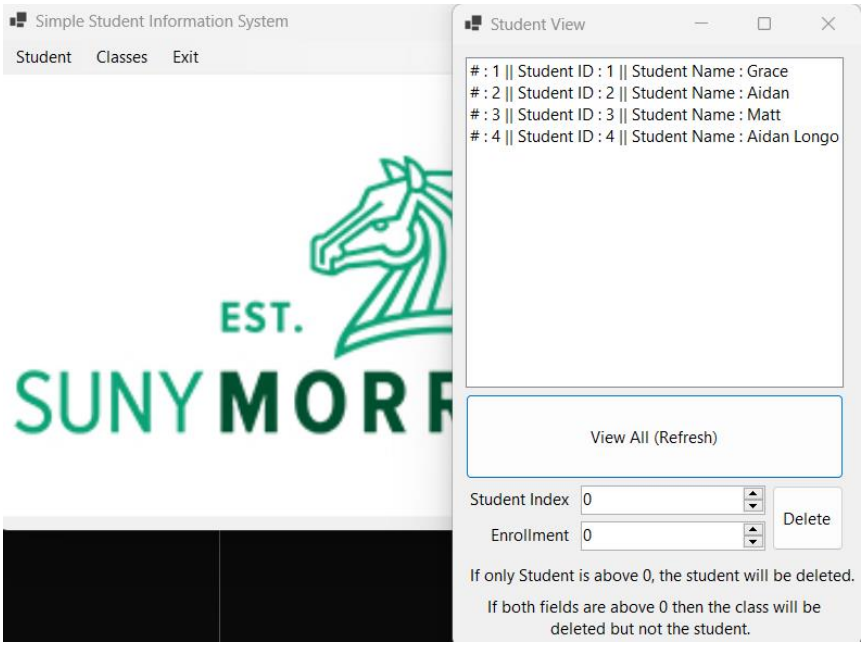
Screenshots:

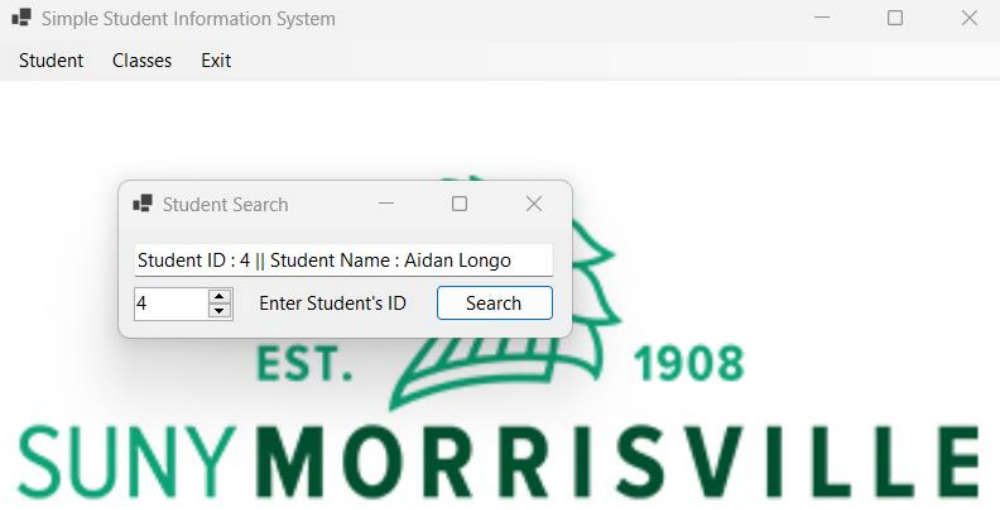
Add Student:



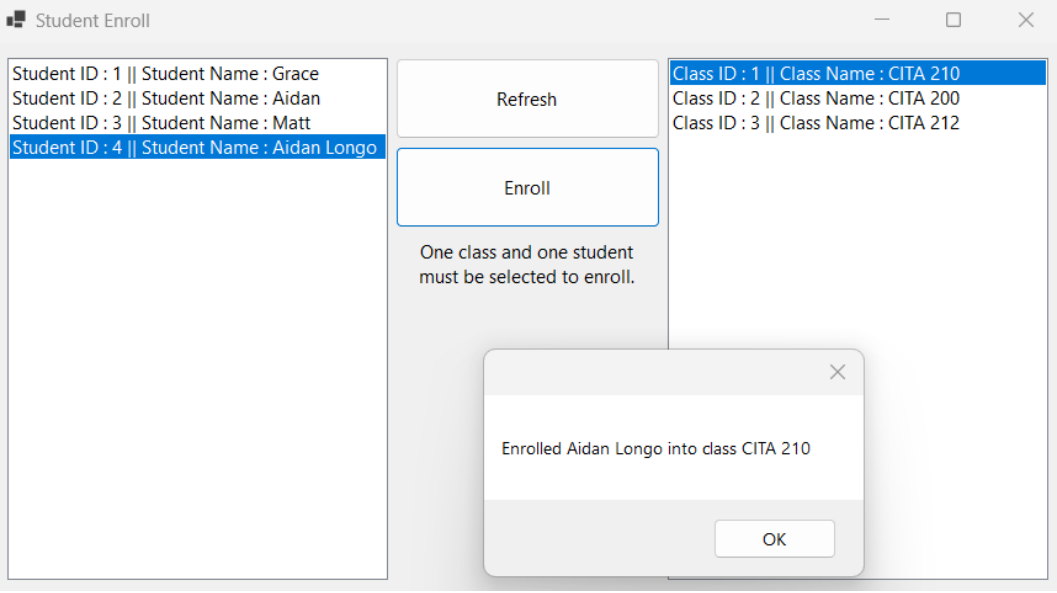
View Student:



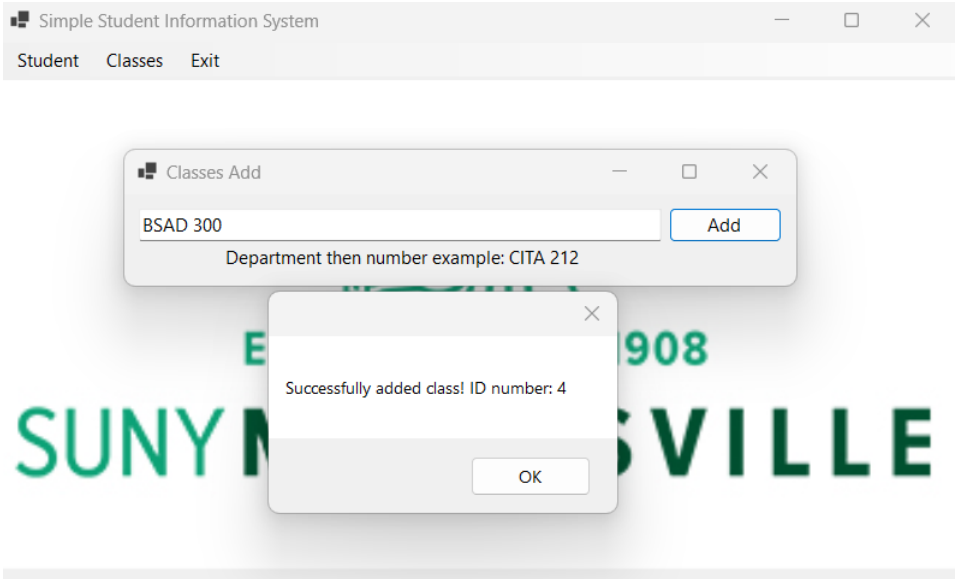
Search Student:



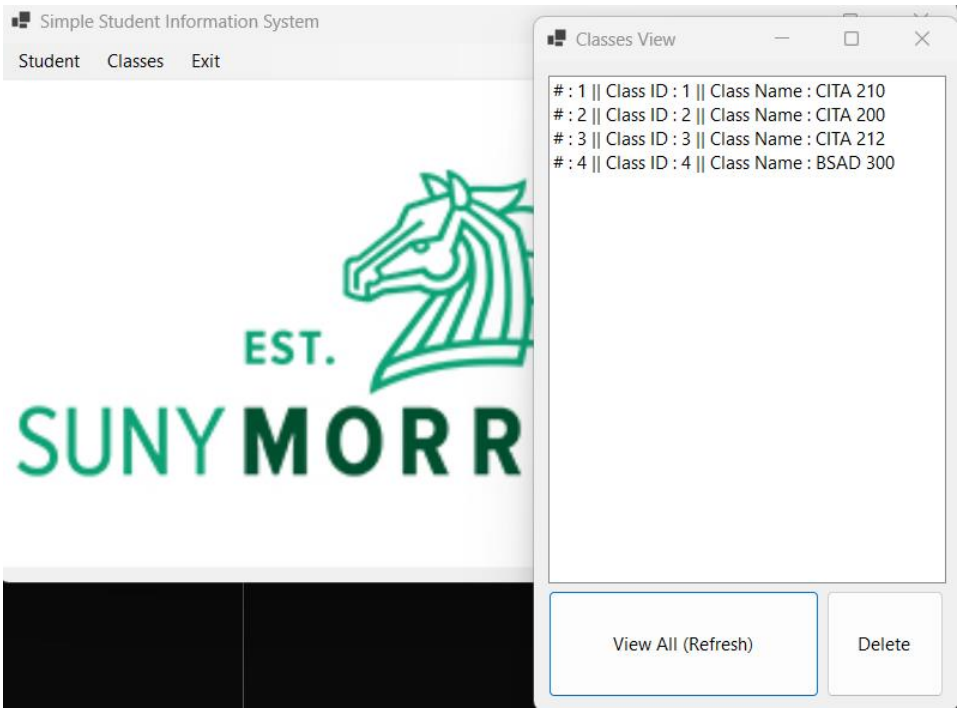
Enroll Student:



Add Class:



View Class:



Code:

```
using System.Collections.Generic;

namespace CITA_218_Final_Project
{
    /*
     * Comments provided by ChatGPT
     * Programmed by Group 2
     */

    // FormHome class represents the main form of the application
    16 references
    public partial class FormHome : Form
    {
        // Instances of other forms
        FormStudentAdd varFormStudentAdd;
        FormStudentView varFormStudentView;
        FormStudentSearch varFormStudentSearch;
        FormStudentEnroll varFormStudentEnroll;
        FormClassesAdd varFormClassesAdd;
        FormClassesView varFormClassesView;

        // Lists to store student and class information
        public List<int> studentId = new List<int>();
        public List<string> studentName = new List<string>();
        public List<int> classId = new List<int>();
        public List<string> className = new List<string>();
        public List<List<string>> registrar = new List<List<string>>();

        // Constructor initializes the main form and adds sample data
        1 reference
        public FormHome()
        {
            InitializeComponent();

            // Sample data for students and their enrolled classes
            studentId.Add(1);
            studentName.Add("Grace");
            registrar.Add(new List<string>());

            studentId.Add(2);
            studentName.Add("Aidan");
            registrar.Add(new List<string>());

            studentId.Add(3);
            studentName.Add("Matt");
            registrar.Add(new List<string>());

            classId.Add(1);
            className.Add("CITA 218");

            classId.Add(2);
            className.Add("CITA 200");

            classId.Add(3);
            className.Add("CITA 212");
        }
```

```
        // Event handler for opening the FormStudentView
        1 reference
        private void FormStudentView_Click(object sender, EventArgs e)
        {
            varFormStudentView = new FormStudentView(this);
            varFormStudentView.Show();
        }

        // Event handler for opening the FormStudentSearch
        1 reference
        private void FormStudentSearch_Click(object sender, EventArgs e)
        {
            varFormStudentSearch = new FormStudentSearch(this);
            varFormStudentSearch.Show();
        }

        // Event handler for opening the FormStudentEnroll
        1 reference
        private void FormStudentEnroll_Click(object sender, EventArgs e)
        {
            varFormStudentEnroll = new FormStudentEnroll(this);
            varFormStudentEnroll.Show();
        }

        // Event handler for opening the FormClassesAdd
        1 reference
        private void FormClassesAdd_Click(object sender, EventArgs e)
        {
            varFormClassesAdd = new FormClassesAdd(this);
            varFormClassesAdd.Show();
        }

        // Event handler for opening the FormClassesView
        1 reference
        private void FormClassesView_Click(object sender, EventArgs e)
        {
            varFormClassesView = new FormClassesView(this);
            varFormClassesView.Show();
        }

        // Event handler for the Exit button to close the application
        1 reference
        private void Exit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        // Reference for the next student ID
        public int studentIdRef = 4;

        // Reference for the next class ID
        public int classIdRef = 4;
    }
}
```

Layout:

The code is made using lists to store data and an interface that allows the user to select different options of what they want to add to the system. The buttons and options are accessed through a menu strip on the top of the form. From there, depending on the option, text boxes are given to enter the necessary information to add to the system.

Steps:

-GitHub repo was made to track changes and files

-Defining the requirements of the system and that needs to be viewable/searchable

-Creating base layout of the menu strip and button dropdowns

-Give code to each drop down button in the menu strip