Scheduling Administration

Project Documentation

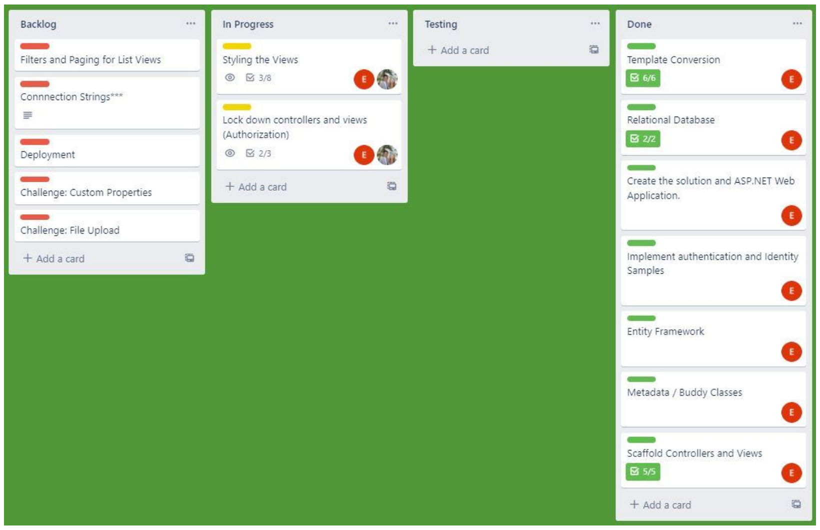
Objective: Create a web application that will allow the staff of a vocational school to handle the enrollments of students in their courses.

Technology Used: The application is built in **the .NET Framework**, Microsoft's stack of development technologies.

- The application was designed using **the MVC Architecture**, which stands for Models, Views, and Controllers. Models give structure to the data for our application to interact with. Views dynamically render pages. Controllers handle the logic and functionality.
- The application utilizes **Entity Framework** as our Object Relational Mapper, which enables us to work with our database.
- The application utilizes **Identity Samples** to handle the authentication of users and the different tiers of authorization among those users.

Process:

To keep track of our tasks throughout the development process, we used a Trello Board shared between developers. Using the scrum project management philosophy, each "card" is a sprint where we set short term goals/todos, wrote the code, tested it, and then moved it to the Done list when finished. We also utilized Pair Programming. As you can see below, Enrique accomplished much of the preliminary work as the Driver with Romeo as the Navigator. The remaining tasks were completed alternating between the two of us.



Backlog
- Filters and Paging for List Views
- Connnection Strings***
- Deployment
- Challenge: Custom Properties
- Challenge: File Upload

In Progress
- Styling the Views — 3/8
- Lock down controllers and views (Authorization) — 2/3

Testing

Done
- Template Conversion — 6/6
- Relational Database — 2/2
- Create the solution and ASP.NET Web Application.
- Implement authentication and Identity Samples
- Entity Framework
- Metadata / Buddy Classes
- Scaffold Controllers and Views — 5/5

A screenshot of our Trello Board halfway thru the project.

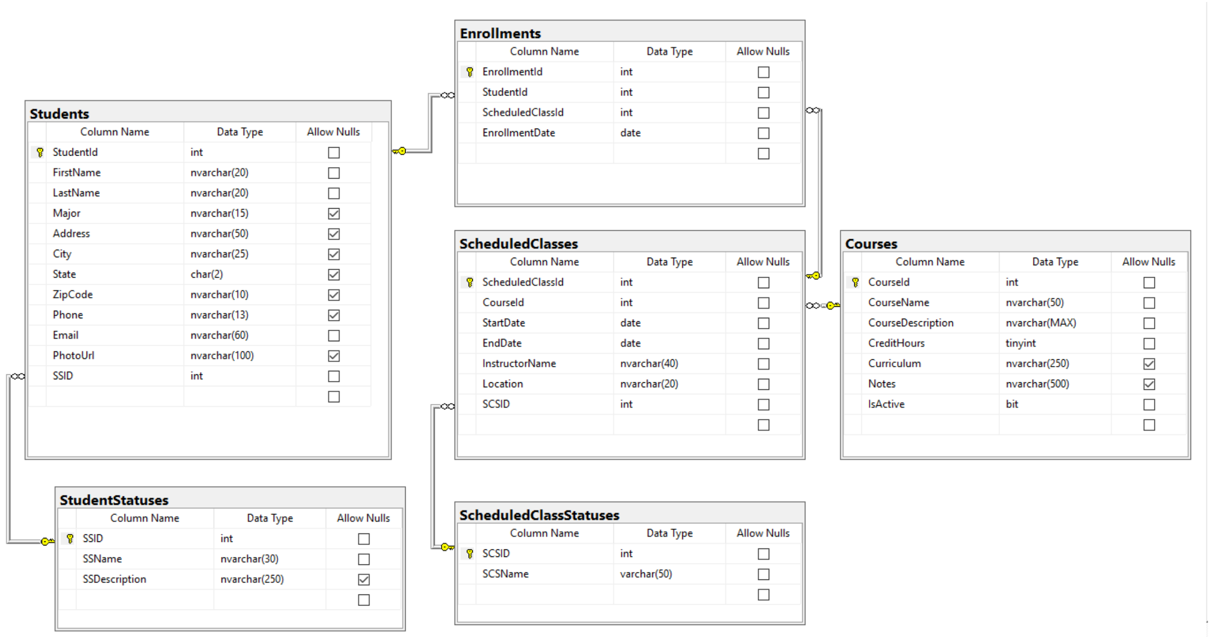
DATA MANAGEMENT

For this application, we designed a relational database in SQL Server Management Studio.

Brief overview of Database tables:

- ***Courses*** – This table stores all of the courses offered by the school, including a unique ID, name and description, credit hours, and any curriculum or notes that need to be recorded. It also records whether a course is active or not.
- ***Scheduled Classes*** – This table keeps record of every time a course was taught, when it was taught, the instructor, and the location. It also references the Scheduled Class Statuses table to note whether that instance of the course is active, pending, complete, or cancelled.
- ***Students*** – This table stores all of the students that will, are, or have attended the school. Each is given a unique ID, and the table takes account of their name, major, address, contact information, and profile picture. It also references the Student Statuses table to track if a student is a prospective student (plans to attend but is not enrolled), a current student, a former student (withdrawn or dismissed), an alumni, or booted.
- ***Enrollments*** – This table handles enrolling each student in a scheduled class and records when that enrollment took place.

Please see the below image of the database for reference:



Enrollments

Column Name	Data Type	Allow Nulls
EnrollmentId	int	☐
StudentId	int	☐
ScheduledClassId	int	☐
EnrollmentDate	date	☐
		☐

Students

Column Name	Data Type	Allow Nulls
StudentId	int	☐
FirstName	nvarchar(20)	☐
LastName	nvarchar(20)	☐
Major	nvarchar(15)	☑
Address	nvarchar(50)	☑
City	nvarchar(25)	☑
State	char(2)	☑
ZipCode	nvarchar(10)	☑
Phone	nvarchar(13)	☑
Email	nvarchar(60)	☐
PhotoUrl	nvarchar(100)	☑
SSID	int	☐
		☐

ScheduledClasses

Column Name	Data Type	Allow Nulls
ScheduledClassId	int	☐
CourseId	int	☐
StartDate	date	☐
EndDate	date	☐
InstructorName	nvarchar(40)	☐
Location	nvarchar(20)	☐
SCSID	int	☐
		☐

Courses

Column Name	Data Type	Allow Nulls
CourseId	int	☐
CourseName	nvarchar(50)	☐
CourseDescription	nvarchar(MAX)	☐
CreditHours	tinyint	☐
Curriculum	nvarchar(250)	☑
Notes	nvarchar(500)	☑
IsActive	bit	☐
		☐

StudentStatuses

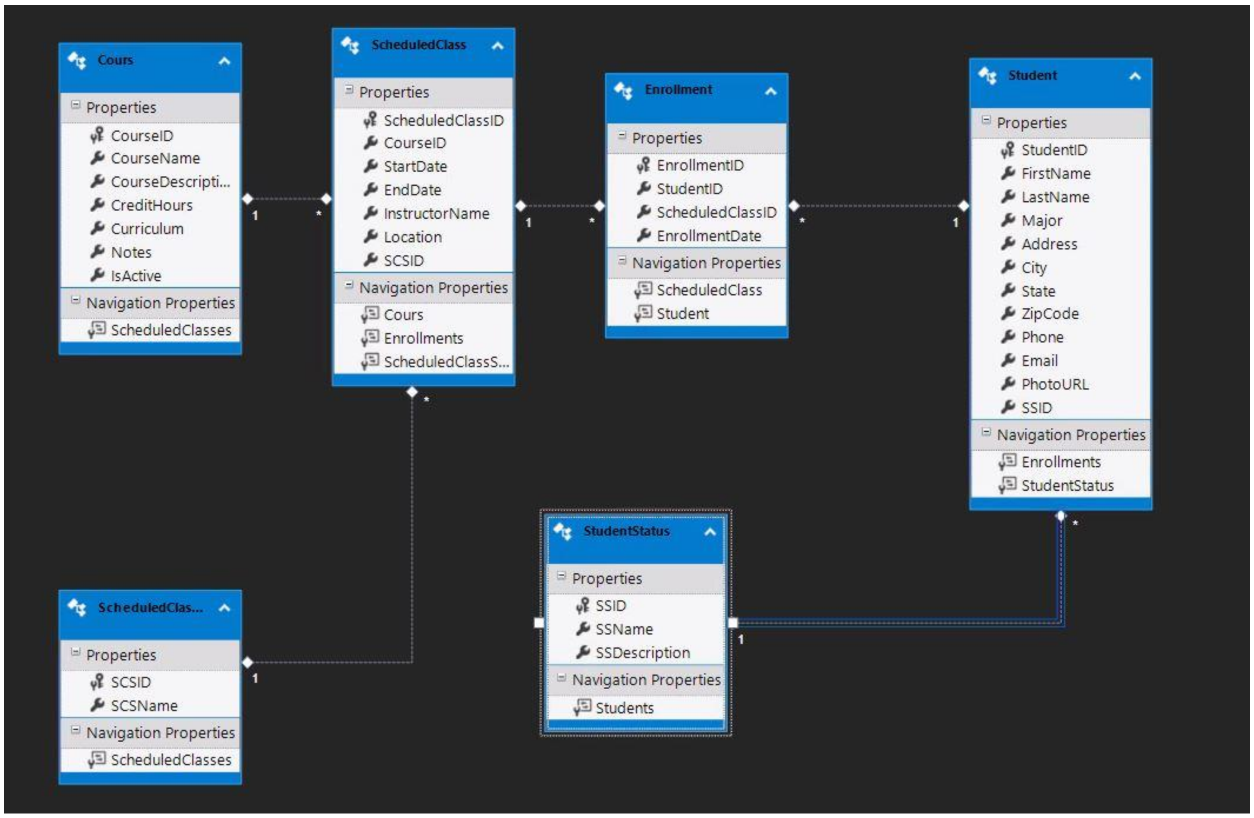
Column Name	Data Type	Allow Nulls
SSID	int	☐
SSName	nvarchar(30)	☐
SSDescription	nvarchar(250)	☑
		☐

ScheduledClassStatuses

Column Name	Data Type	Allow Nulls
SCSID	int	☐
SCSName	varchar(50)	☐
		☐

After the database was built, we used Entity Framework to hook up the database to our application. This allows the data to be created, read, added, or deleted from the tables.

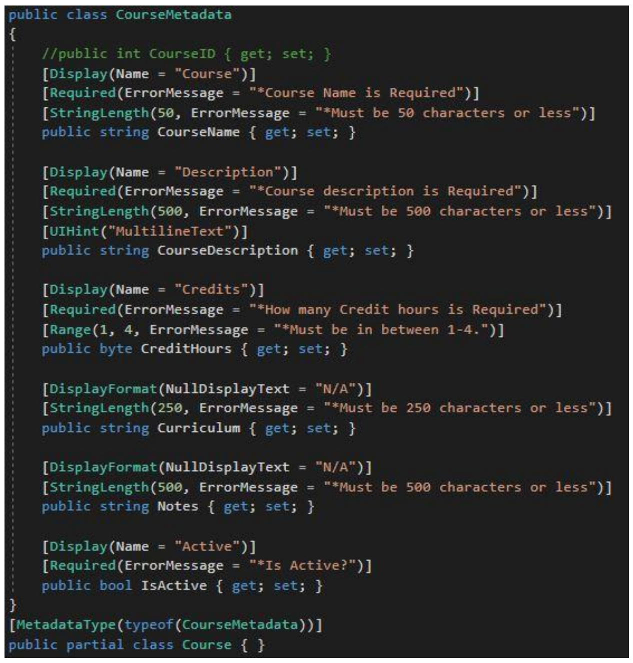
Below is a screenshot of our Entity Framework Diagram, matching the database:



To finish implementing the datatables in our application, we added metadata describing the different characteristics of our data.

For example, see the metadata to the right:

- We added Display names which take the coded names of our properties (e.g. "CourseName") and change it to a display-friendly name (e.g. "Course").
- We specified the minimum and maximum lengths for text or values for numbers. For example "CourseDescription" is capped out at 500 characters, and "CreditHours" only takes integers 1 thru 4.
- We specified which properties are required and which are optional.



```
public class CourseMetadata
{
    //public int CourseID { get; set; }
    [Display(Name = "Course")]
    [Required(ErrorMessage = "*Course Name is Required")]
    [StringLength(50, ErrorMessage = "*Must be 50 characters or less")]
    public string CourseName { get; set; }

    [Display(Name = "Description")]
    [Required(ErrorMessage = "*Course description is Required")]
    [StringLength(500, ErrorMessage = "*Must be 500 characters or less")]
    [UIHint("MultilineText")]
    public string CourseDescription { get; set; }

    [Display(Name = "Credits")]
    [Required(ErrorMessage = "*How many Credit hours is Required")]
    [Range(1, 4, ErrorMessage = "*Must be in between 1-4.")]
    public byte CreditHours { get; set; }

    [DisplayFormat(NullDisplayText = "N/A")]
    [StringLength(250, ErrorMessage = "*Must be 250 characters or less")]
    public string Curriculum { get; set; }

    [DisplayFormat(NullDisplayText = "N/A")]
    [StringLength(500, ErrorMessage = "*Must be 500 characters or less")]
    public string Notes { get; set; }

    [Display(Name = "Active")]
    [Required(ErrorMessage = "*Is Active?")]
    public bool IsActive { get; set; }
}
[MetadataType(typeof(CourseMetadata))]
public partial class Course { }
```

AUTHENTICATION/AUTHORIZATION

Our MVC Application utilizes IdentitySamples to handle authentication of users and the levels of authorization within the application.

A brief summary of authorization levels:

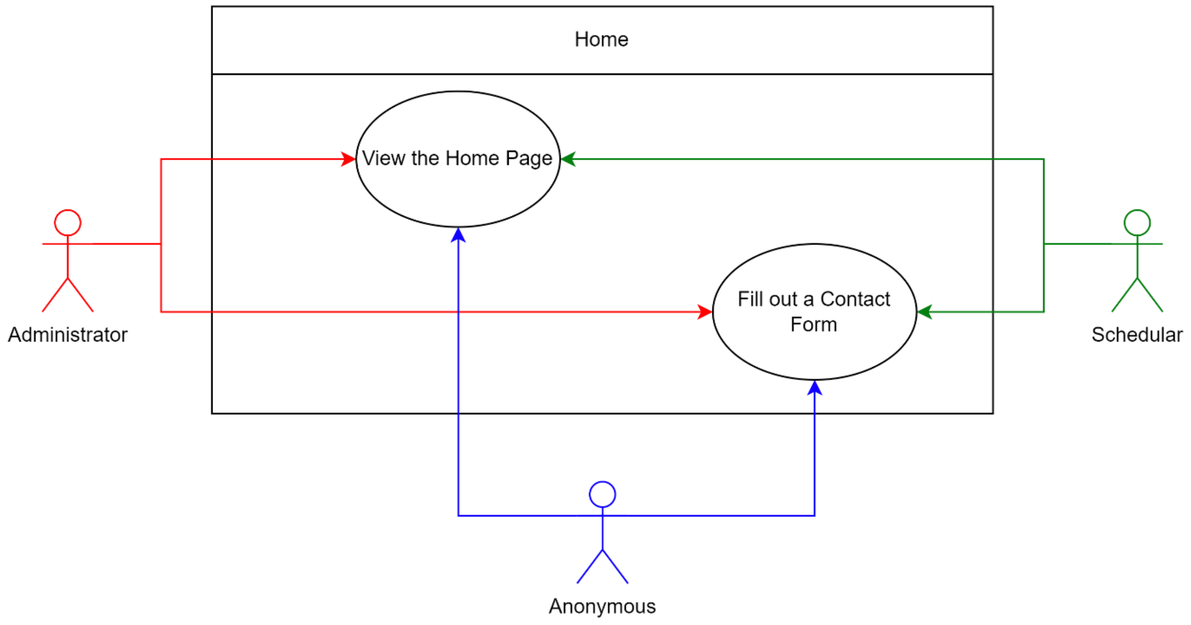
1. **Administrator** – Admin users have the ability to add, edit, or delete Students, Enrollments, Scheduled Classes, as well as Courses and Student Statuses. This is the highest level of authorization and should be handled with care.
2. **Schedular** – The main feature for this user is the ability to add, edit, or delete Enrollments. They can view students as they manage enrollments, but they cannot add, edit, or delete students from the record. Similarly, they can view, add, and edit courses on the schedule, but they cannot remove them. They do not have access to the Courses and Student Statuses.
3. **Anonymous** – Random visitors to the website will have the ability to view the home page and send in a contact form, but they cannot view any other pages.

An Identity Matrix was made available for reference.

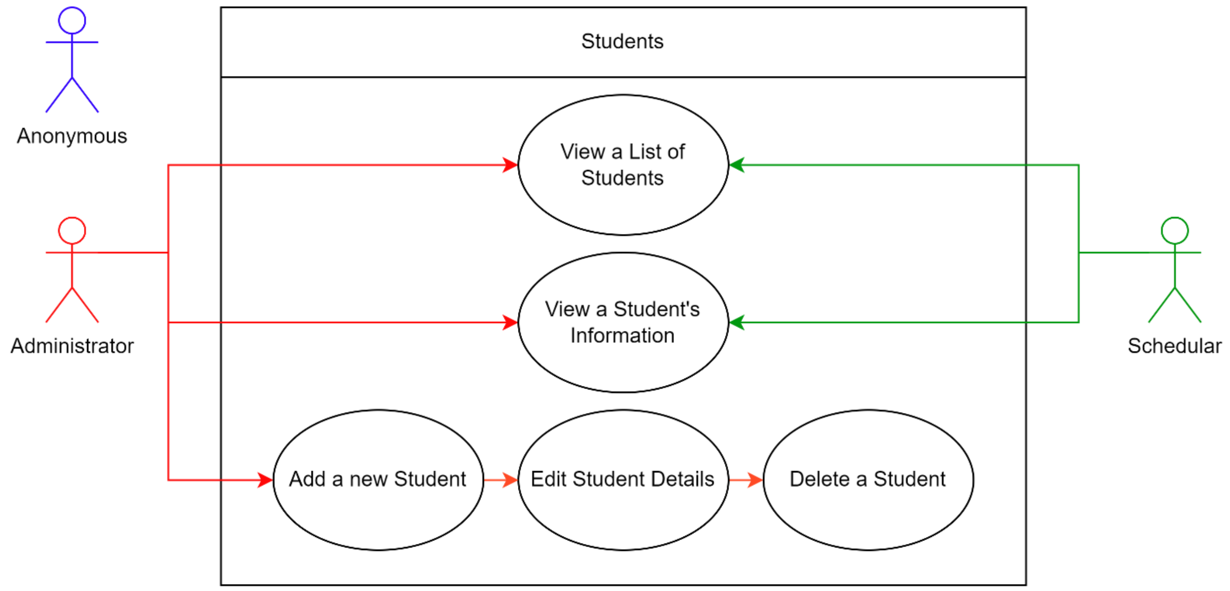
If a user attempts to access a page which they are not authorized to, they are redirected to log in.

Please see the following use case diagrams for a complete visual breakdown of authorization:

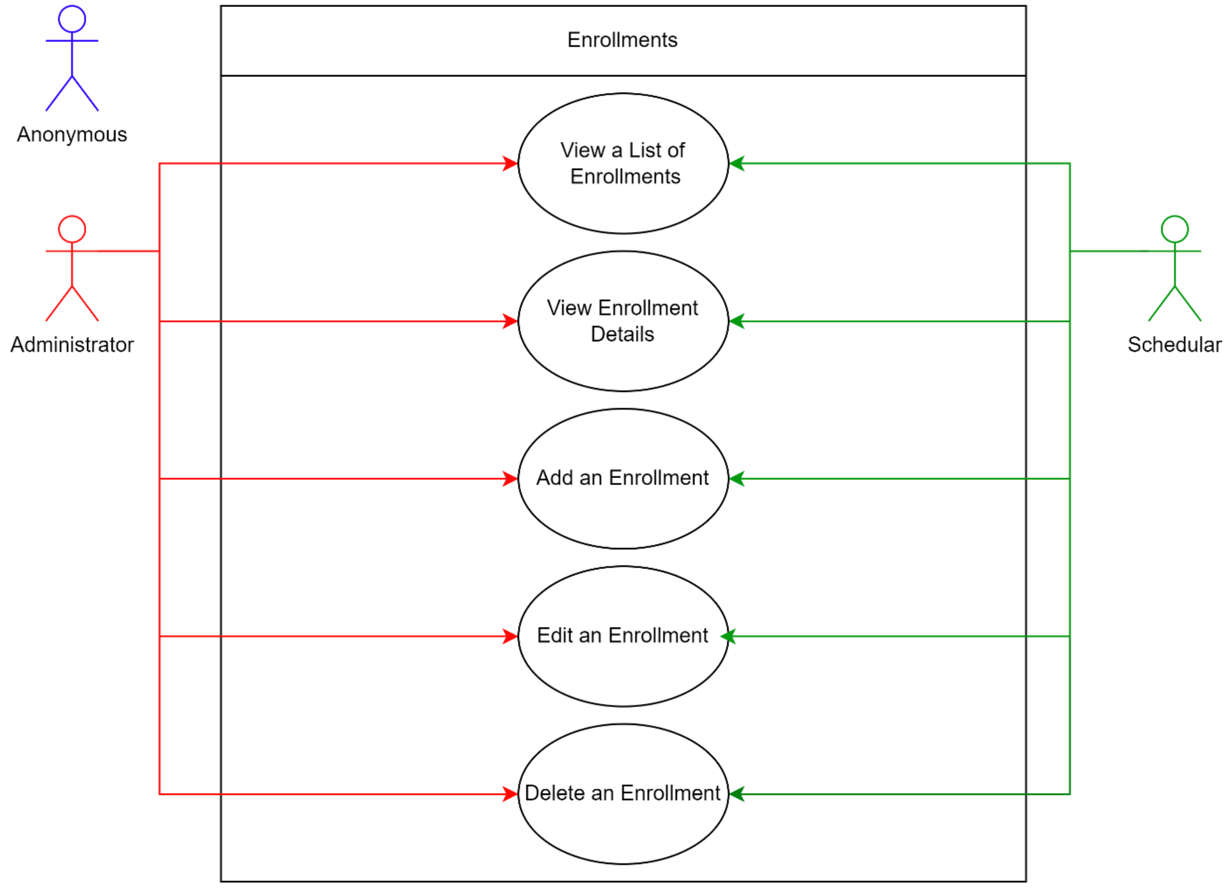
Home



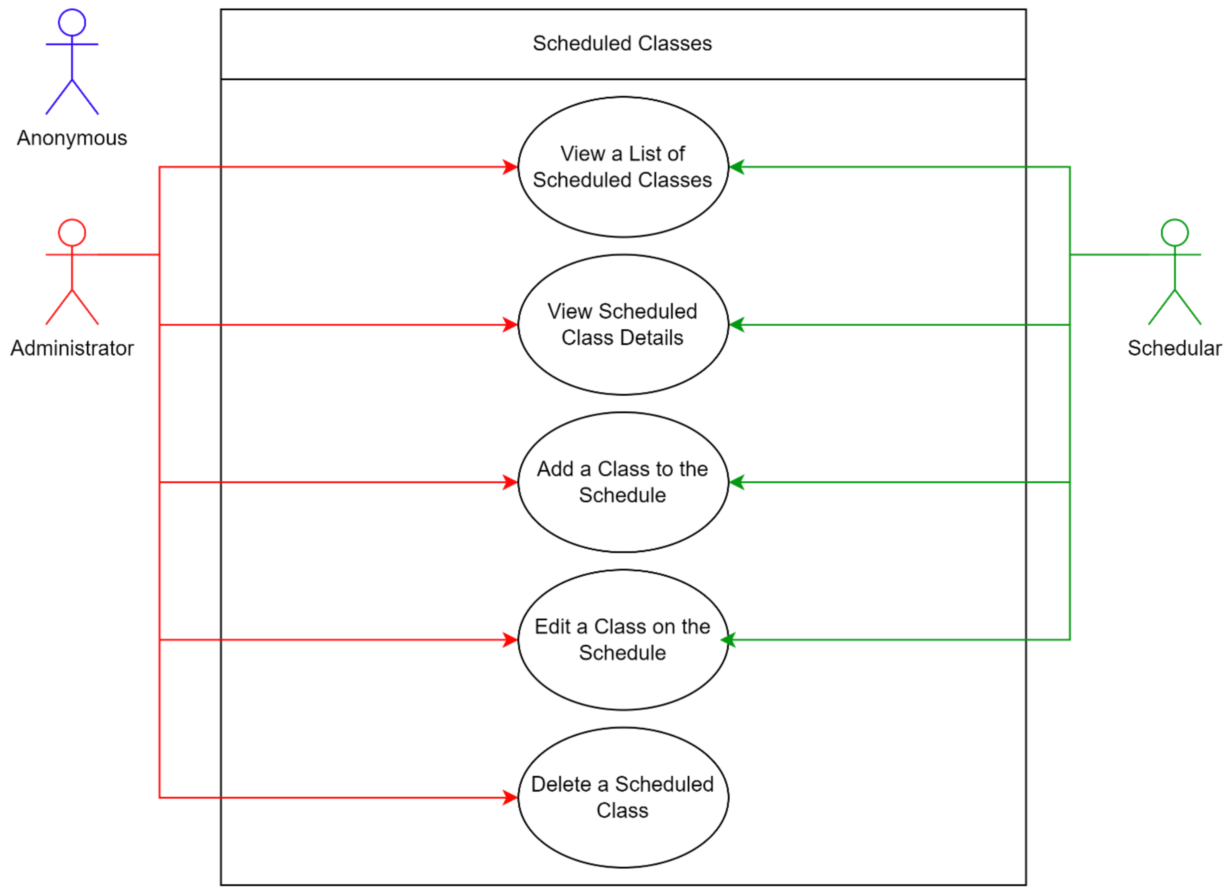
Students



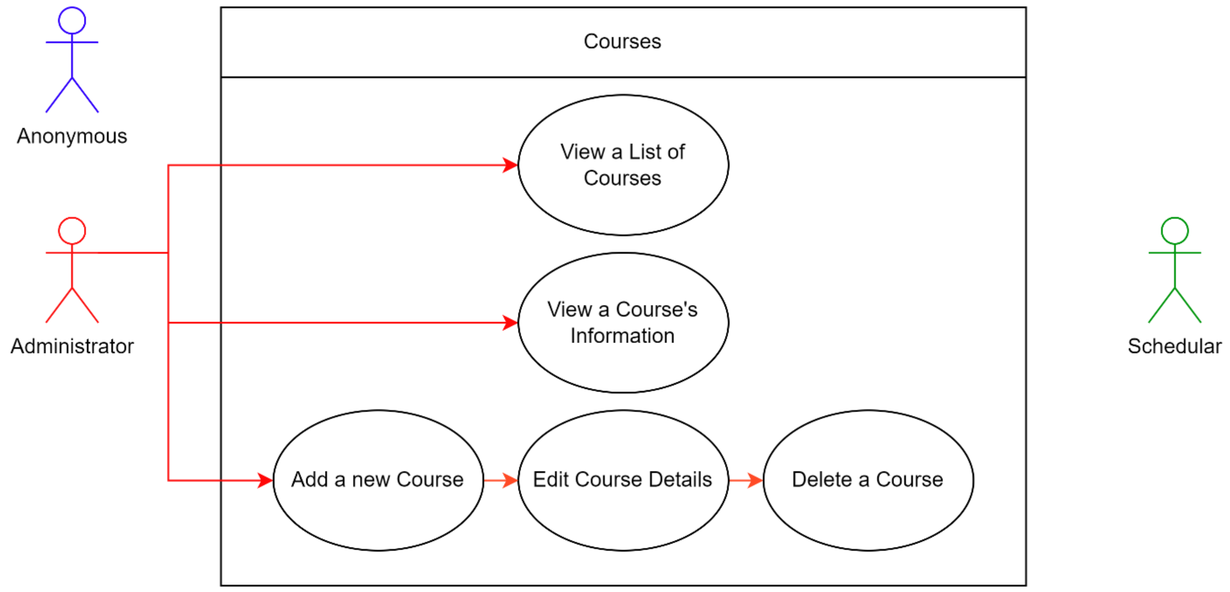
Enrollments



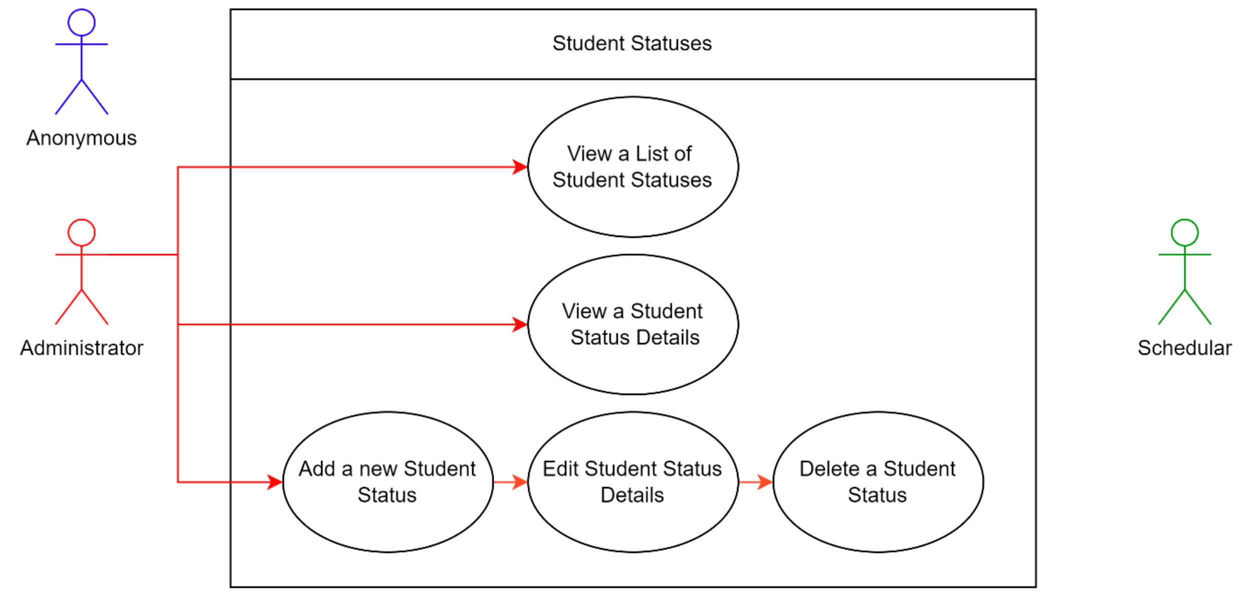
Class Schedule


Scheduled Classes
Anonymous
Administrator
Schedular
View a List of Scheduled Classes
View Scheduled Class Details
Add a Class to the Schedule
Edit a Class on the Schedule
Delete a Scheduled Class

Courses



Student Statuses

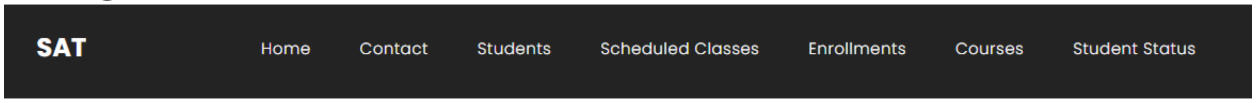


Authorization Example

Certain buttons are hidden for schedulars which are available to admins.

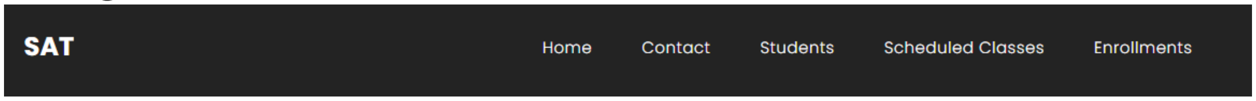
See the following examples of differences between user interfaces for different users:

Navigation Bar (Administrator View)



When an administrator is logged in, they can quickly access the Courses and Student Status pages.

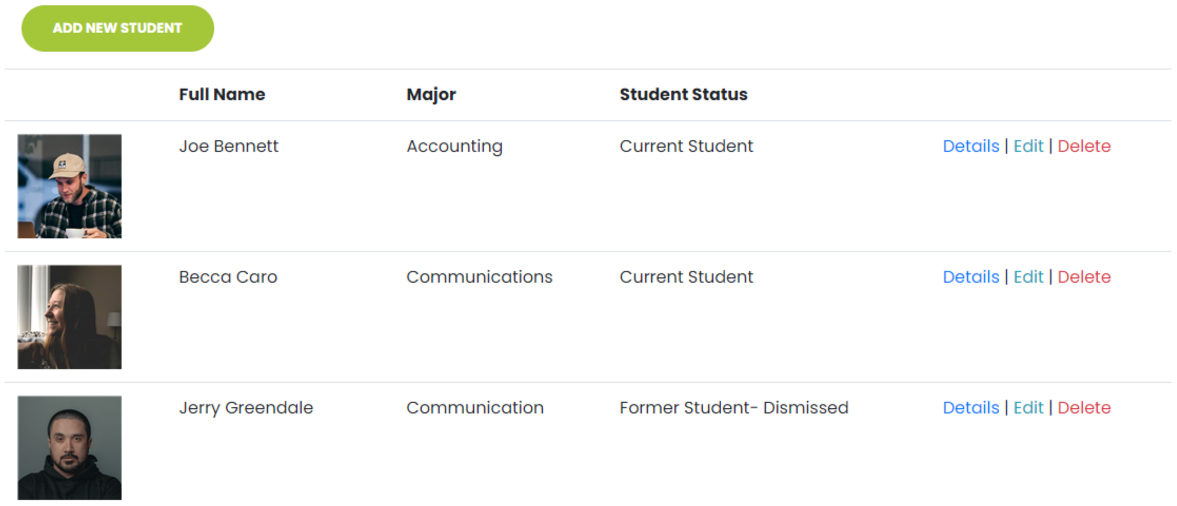
Navigation Bar (Schedular View)



Since schedulars are not permitted to access the Courses or Student Status tables, they are hidden from the navbar.

If schedulars attempted to access these pages anyway, they would be redirected to the login page, essentially prompting them to provide admin-level credentials.

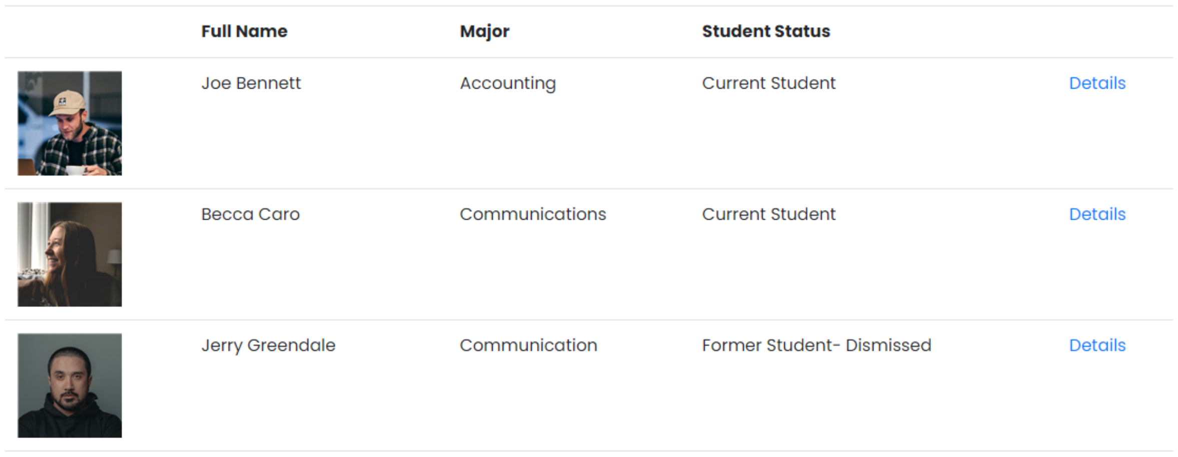
Students (Administrator View)



	Full Name	Major	Student Status	
	Joe Bennett	Accounting	Current Student	Details | Edit | Delete
	Becca Caro	Communications	Current Student	Details | Edit | Delete
	Jerry Greendale	Communication	Former Student- Dismissed	Details | Edit | Delete

Administrators have access to Add, Edit or Delete students.

Students (Schedular View)



	Full Name	Major	Student Status	
	Joe Bennett	Accounting	Current Student	Details
	Becca Caro	Communications	Current Student	Details
	Jerry Greendale	Communication	Former Student- Dismissed	Details

Schedulars do not have access to the add, edit, or delete buttons.

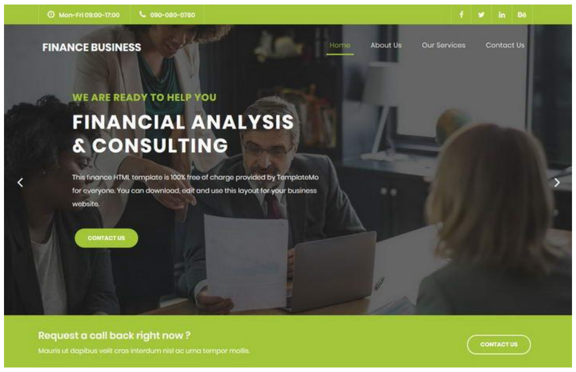
UI/UX Design Work

Template:

We selected the free Bootstrap template Finance Business from templatemo.

Key template features:

- Sans-serif font.
- Green-black-white color scheme.
- Responsive and animated navbar.



WE ARE READY TO HELP YOU

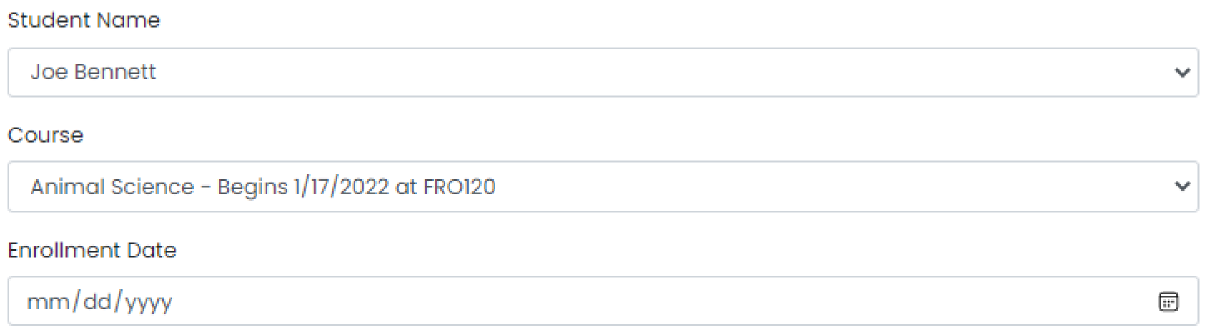
FINANCIAL ANALYSIS & CONSULTING

This finance HTML template is 100% free of charge provided by TemplateMo for everyone. You can download, edit and use this layout for your business website.

Request a call back right now ?

Mauris ut dapibus velit cras interdum nisl ac urna tempor mollis.

User-Friendly Forms:



Student Name	Joe Bennett
Course	Animal Science - Begins 1/17/2022 at FRO120
Enrollment Date	mm/dd/yyyy

As you can see in the screenshot above, enrolling a student in a course is as simple as selecting that student by full name, selecting the course (start date and location are noted for clarity), then selecting an enrollment date in a calendar pop-up.

Dynamic Page Banners

The above banner is rendered on every page with a dynamic title for each page.