E.A.T

Educational Administration Tool

Purpose

To help educational facilities create, track, manage their Students, Courses, and Enrollments

Tech Overview

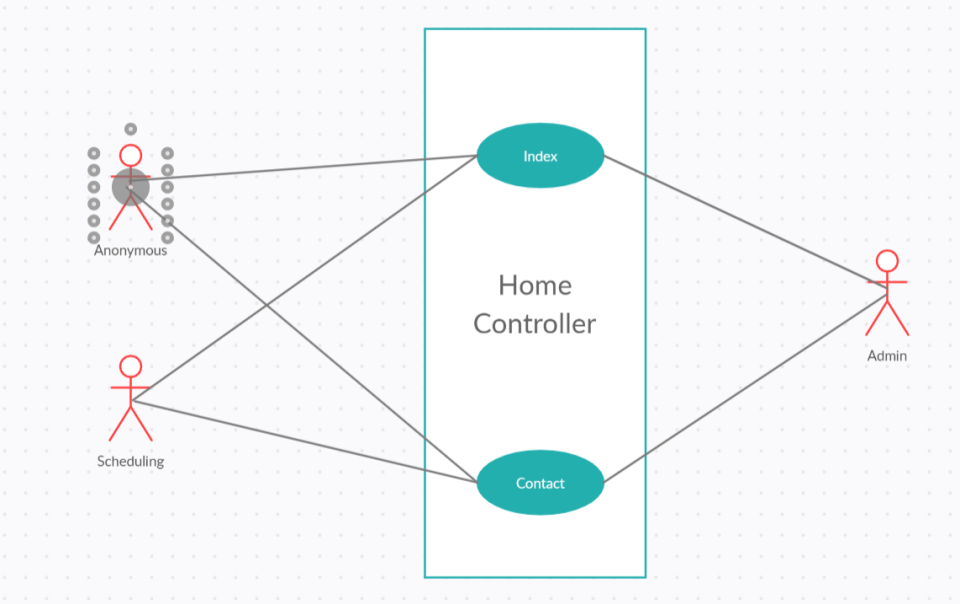
Utilized MVC, C#, HTML, CSS, JavaScript, SQL. Made in .NET framework using Visual Studio and SSMS. Used GitHub for Source Control. Made project using Extreme Coding (Pair Programming)

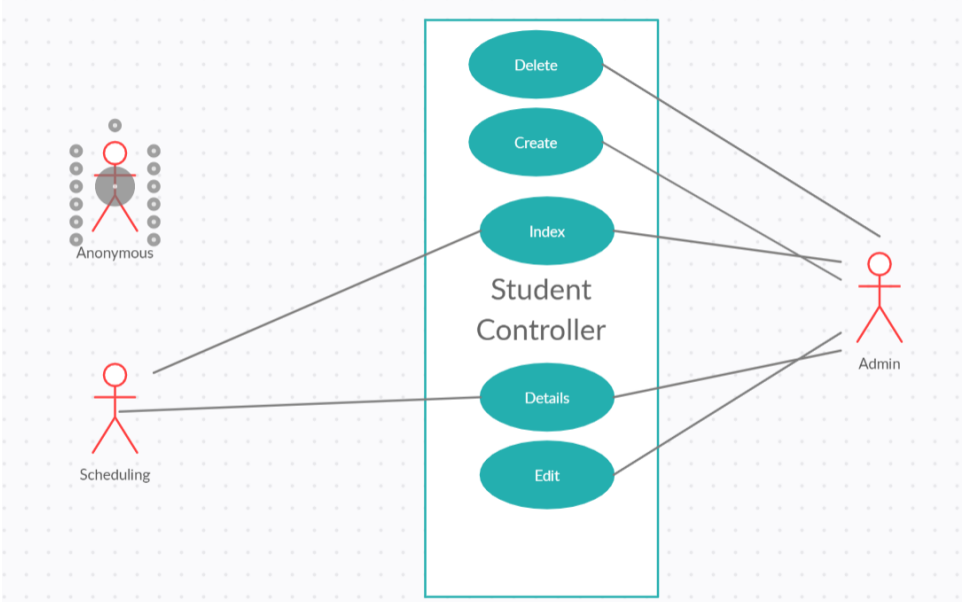
Roles and Explanations

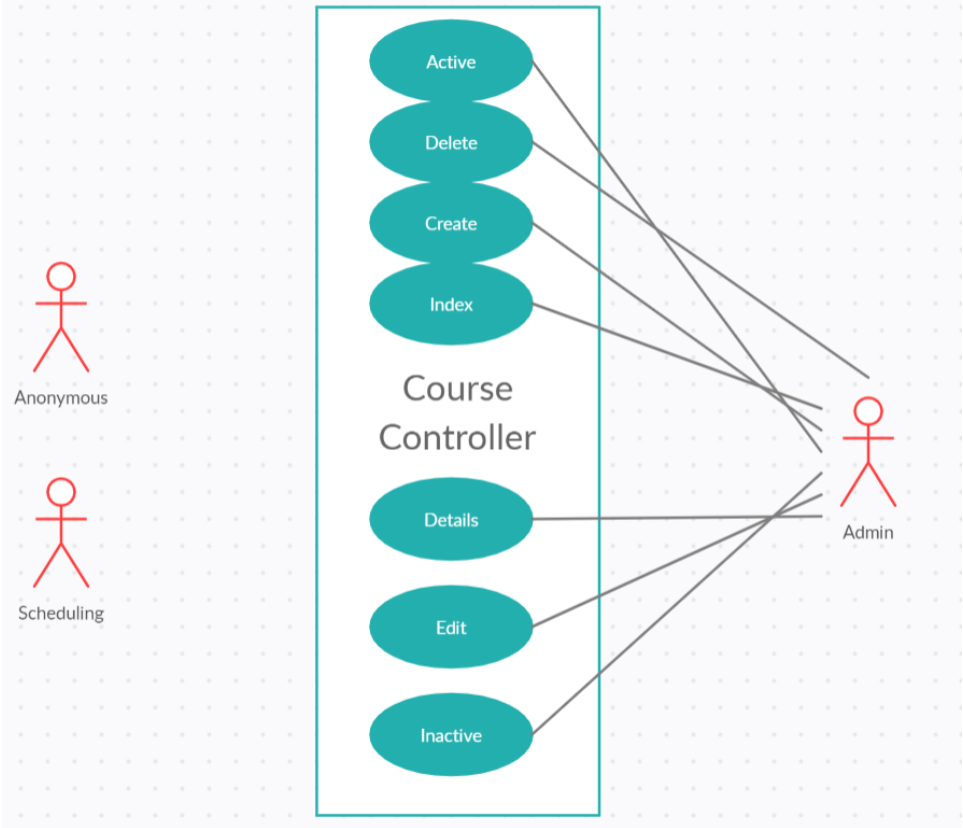
Anonymous – Represents the generic user only has access to the Home page and Contact page

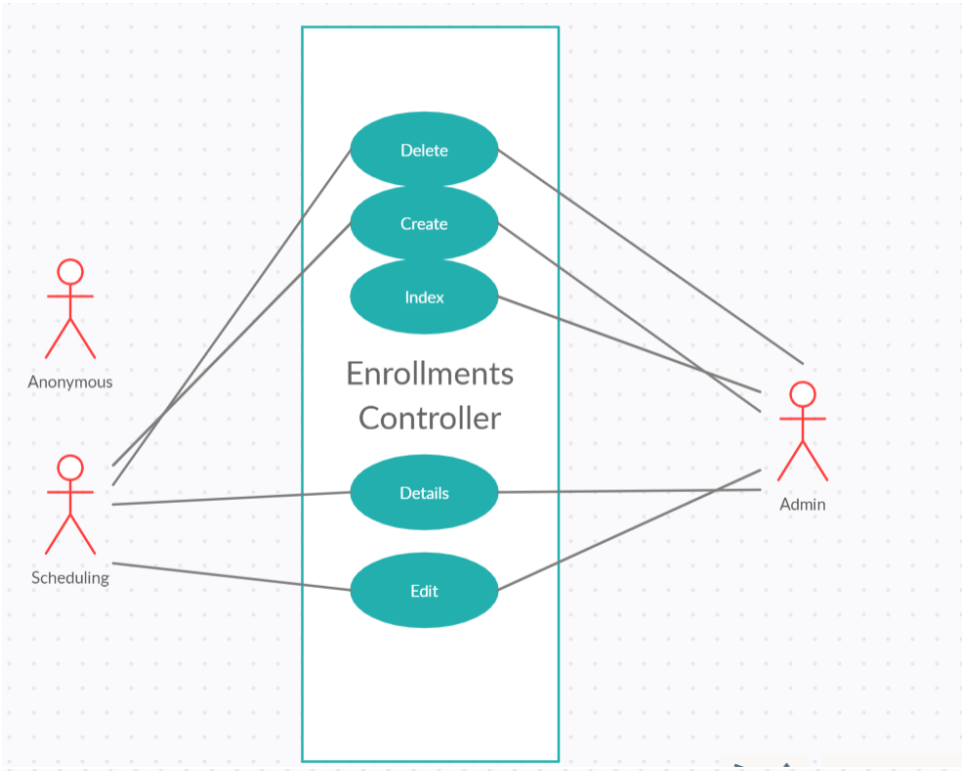
Scheduling – Represents an employee who has been given access to certain CRUD functionality and views.

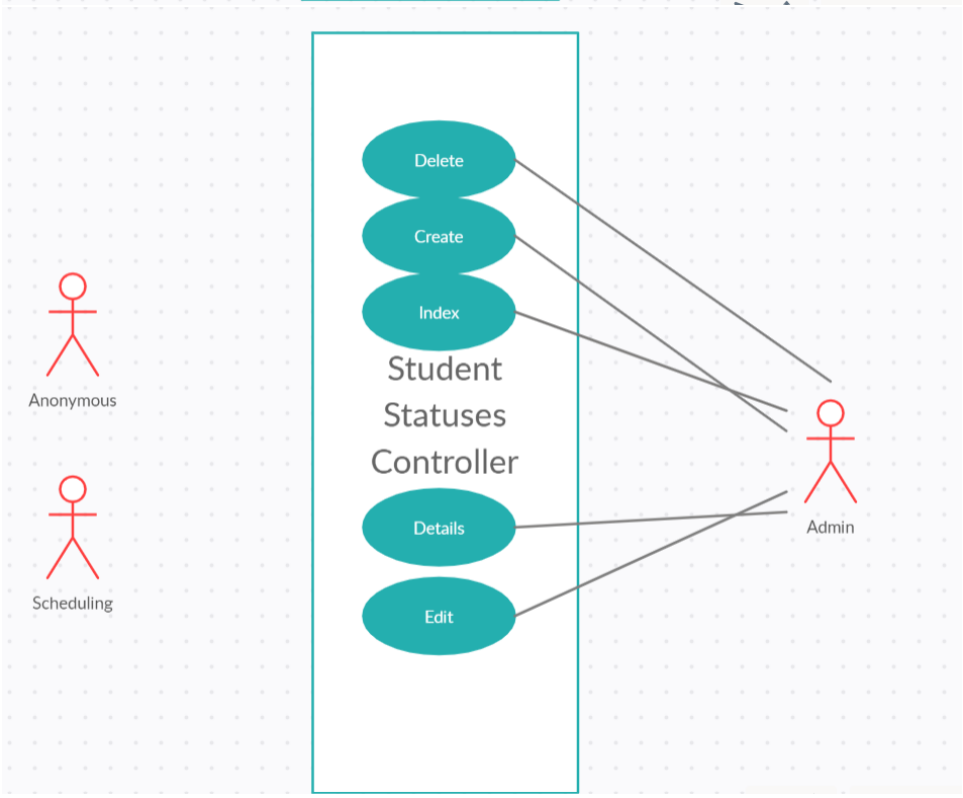
Admin – Given access to ALL CRUD functionality and views

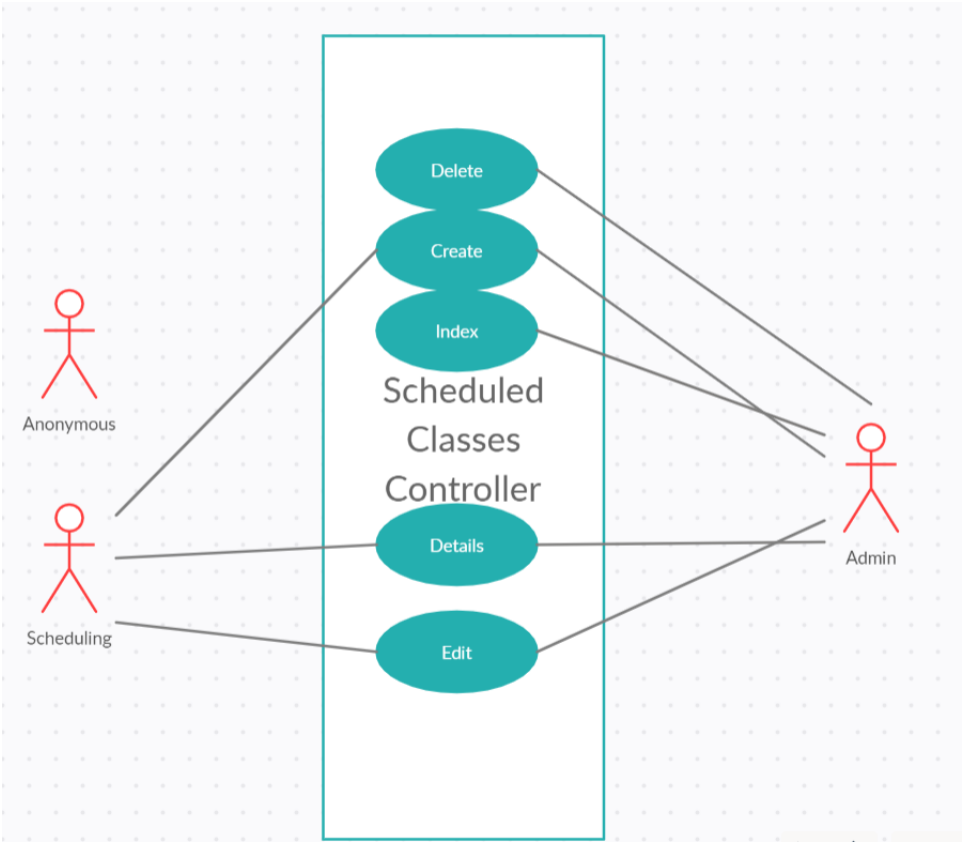



Delete
Create
Index
Student
Controller
Details
Edit
Anonymous
Scheduling
Admin

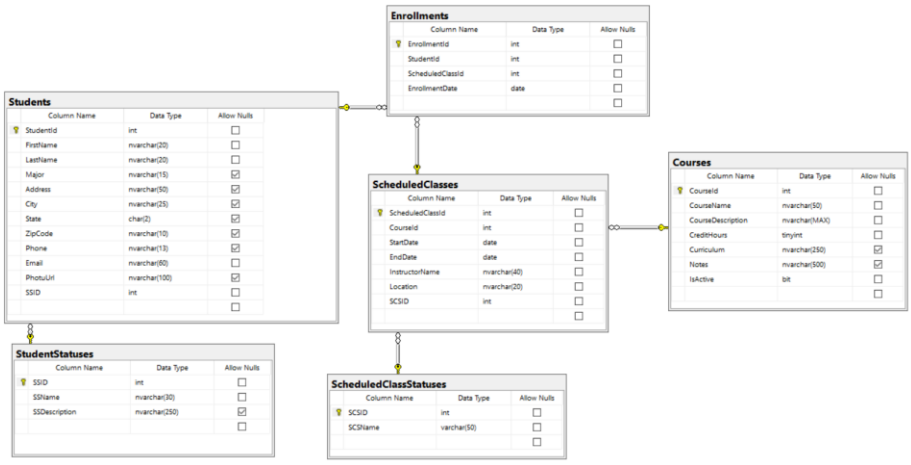

Active
Delete
Create
Index
Course
Controller
Details
Edit
Inactive
Anonymous
Scheduling
Admin


Delete
Create
Index
Enrollments Controller
Details
Edit
Anonymous
Scheduling
Admin


Delete
Create
Index
Student Statuses Controller
Details
Edit
Anonymous
Scheduling
Admin



Workflow Overview



Database Created in SSMS to meet client needs

```
<ul>
    <li class="active"><a href="index.html">Home</a></li>
    <li><a href="about.html">About</a></li>
    <li><a href="pricing.html">Pricing</a></li>
    @if (User.IsInRole("Admin") && Request.IsAuthenticated)
    {
        <li class="has-dropdown">
            <a href="blog.html">Admin Features</a>
            <ul class="dropdown">
                <li><a href=@Url.Action("Index", "Cours")>Courses</a></li>
                <li><a href=@Url.Action("Index", "Enrollments")>Enrollments</a></li>
                <li><a href=@Url.Action("Index", "ScheduledClasses")>Scheduled Classes</a></li>
                <li><a href=@Url.Action("Index", "Students")>Students</a></li>
                <li><a href=@Url.Action("Index", "StudentStatus")>Student Statuses</a></li>
            </ul>
        </li>
    }
    else if (User.IsInRole("Scheduling") && Request.IsAuthenticated)
    {
        <li class="has-dropdown">
            <a href="blog.html">Scheduling Features</a>
            <ul class="dropdown">
                <li><a href=@Url.Action("Index", "Enrollments")>Enrollments</a></li>
                <li><a href=@Url.Action("Index", "ScheduledClasses")>Scheduled Classes</a></li>
                <li><a href=@Url.Action("Index", "Students")>Students</a></li>
            </ul>
        </li>
    }
```

Functioning nav bar with Role based access

```
Delete.cshtml   Inactive.cshtml   Index.cshtml   ScheduledClassesController.cs
EAT.MVC.UI                    EAT.MVC.UI.Controllers.ScheduledClassesController
122             if (scheduledClass == null)
123             {
124                 return HttpNotFound();
125             }
126             return View(scheduledClass);
127         }
128
129         // POST: ScheduledClasses/Delete/5
130         [HttpPost, ActionName("Delete")]
131         [ValidateAntiForgeryToken]
132         [Authorize(Roles = "Admin")]
133         public ActionResult DeleteConfirmed(int id)
134         {
135             ScheduledClass scheduledClass = db.ScheduledClasses.Find(id);
136             if (scheduledClass.SCSID != 4)
137             {
138                 scheduledClass.SCSID = 4;
139             }
140             else if (scheduledClass.SCSID == 4)
141             {
142                 scheduledClass.SCSID = 1;
143             }
144             //db.ScheduledClasses.Remove(scheduledClass);
145             db.SaveChanges();
146             return RedirectToAction("Index");
147         }
148
149         protected override void Dispose(bool disposing)
150         {
151             if (disposing)
152             {
125 %
```

Soft Delete for scheduled classes (Active to Inactive and vice versa)

```
<td>
    @Html.DisplayFor(modelItem => item.StudentStatus.SSName)
</td>
<td>
    @Html.ActionLink("Details", "Details", new { id = item.StudentId }) |
    @if (User.IsInRole("Admin"))
    {
        @Html.ActionLink("Edit", "Edit", new { id = item.StudentId })
        @Html.ActionLink("Delete", "Delete", new { id = item.StudentId })
    }

</td>
```

Lockdown Certain actions in the Index Views based on Roles

```
[MetadataType(typeof(ScheduledClassMetadata))]
public partial class ScheduledClass
{
    public string ClassInfo { get { return $"{StartDate} {Cours.CourseName} {Location}"; } }
}
```

Custom property Metadata to display Course info

```
student, HttpPostedFileBase studentImage)
{
 if (ModelState.IsValid)
 {
    #region Image upload
    string file = "DefaultImage.jpg";
    if (studentImage != null)

        file = studentImage.FileName;
        string ext = file.Substring(file.LastIndexOf('.'));
        string[] goodExts = { ".jpeg", ".jpg", ".png", ".gif" };
        if (goodExts.Contains(ext))
        {
            if (studentImage.ContentLength <= 4194304)
            {
            file = Guid.NewGuid() + ext;

            string savePath = Server.MapPath("~/Content/images/StudentPhotos/");
            Image convertedImage = Image.FromStream(studentImage.InputStream);
            int maxImageSize = 500;
            int maxThumbSize = 100;
            ImageService.ResizeImage(savePath, file, convertedImage, maxImageSize, maxThumbSize);
            }


        }
    student.PhotoUrl = file;
```

Image upload functionality and making a thumbnail of uploaded image

```
[Authorize(Roles = "Admin")]
public ActionResult Inactive()
{
    return View(db.Courses.Where(c => !c.IsActive).ToList());
}
```

Returning certain Courses based on if the course is Active or not

```
namespace EAT.MVC.UI.Controllers
{
    public class StudentsController : Controller
    {
        private SATEntities db = new SATEntities();

        // GET: Students
        [Authorize(Roles = "Admin, Scheduling")]
        public ActionResult Index()
        {
            var students = db.Students.Include(s => s.StudentStatus);
            return View(students.ToList());
        }

        // GET: Students/Details/5
        [Authorize(Roles = "Admin, Scheduling")]
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Student student = db.Students.Find(id);
            if (student == null)
            {
                return HttpNotFound();
            }
            return View(student);
        }
```

Locking down CRUD functionality based on user role in the controller

```
<li style="color:black">
        Hello @User.Identity.GetUserName()!

</li>
<li><a href="javascript:document.getElementById('logoutForm').submit()">Log off</a></li>
```

Custom Message shown while logged in

```
bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
            "~/Scripts/jquery-{version}.js",
            "~/Content/js/jquery.min.js",
            "~/Content/js/jquery.easing.1.3.js",
            "~/Content/js/bootstrap.min.js",
            "~/Content/js/jquery.waypoints.min.js",
            "~/Content/js/jquery.stellar.min.js",
            "~/Content/js/owl.carousel.min.js",
            "~/Content/js/jquery.flexslider-min.js",
            "~/Content/js/jquery.countTo.js",
            "~/Content/js/jquery.magnific-popup.min.js",
            "~/Content/js/magnific-popup-options.js",
            "~/Content/js/simplyCountdown.js",
            "~/Content/js/main.js"
            ));


bundles.Add(new StyleBundle("~/Content/css").Include(
          "~/Content/bootstrap.css",
          "~/Content/site.css",
          "~/Content/css/animate.css",
          "~/Content/css/icomoon.css",
          "~/Content/css/bootstrap.css",
          "~/Content/css/magnific-popup.css",
          "~/Content/css/owl.carousel.min.css",
          "~/Content/css/owl.theme.default.min.css",
          "~/Content/css/flexslider.css",
          "~/Content/css/pricing.css",
          "~/Content/css/style.css"
          ));
```

Bundled both the CSS and js to reduce load time

```
            </div>
        </div>
        <div class="row">
            <div class="row">...</div>
            <div class="col-md-6 animate-box">
                <div class="course">
                    <img src="~/Content/images/facepainting10.8.png" class="course-img" />
                    @*<a href="#" class="course-img" style="background-image: url(../../
                    Content/images/facepainting10.8.png);">
                </a>*@
                    <div class="desc">
                        <h3><a href="#">Face Painting</a></h3>
                        <p>A family favorite, everyone enjoys a great face painting! We use
                    organic materials so as not to harm our subjects. Brush techniques and
                    learning to get small children to sit still are skills you will take away
                     from this class.</p>
                        <span><a href=@Url.Action("Index", "Cours") class="btn btn-primary
                    btn-sm btn-course">Take A Course</a></span>
                    </div>
                </div>
            </div>
            <div class="col-md-6 animate-box">
                <div class="course">
                    <img src="~/Content/images/trapez.jpg" class="course-img" />
                    @*<a href="#" class="course-img" style="background-image: url(../../
```

Created own images and added styling to format them on the Home page

```
<dd>
    <img src="~/Content/images/StudentPhotos/@Html.DisplayFor(model => model.PhotoUrl)" />
</dd>
```

Created to show Student picture in the details