Upside Down University

Scheduling Admin Tool

Jacob Woodworth

Danielle Smalley

Project Goal:

Design and build a web app that is used by university staff to manage student enrollments into courses.

Requirements Gathering:

1. We were not given a mockup or template to use. We were given full creative control over the template and appearance of the application.
2. The roles using this application were:
    a. Anonymous (Visitors)
    b. Scheduling
    c. Admin
3. This application is NOT public facing, so it is for internal use only.
4. This is a dynamic, data-driven app. Data is coming from a database that we built based on schema provided to us.
5. This project did require Identity
6. Timeframe for sprint: Project start 8/3 after lunch through 8/6 11:30am
7. An identity matrix was provided that laid out the controllers, views, and what each user role should be able to access.
8. The functionality details given were:
    a. There is a Students table that connects to Courses table, which connects to Enrollments table.
    b. The users authorized to see the info should be able to see a student enrolled in a course.

Technologies Used to Solve The Problem:

HTML5

CSS3

JavaScript

C#

ASP.Net

MVC

Entity Frameworks

Sequel Server Management Studio

Identity Samples

jQuery / jQuery DataTables

Bootstrap

Debugging

Pair Programming

Trello

Zoom

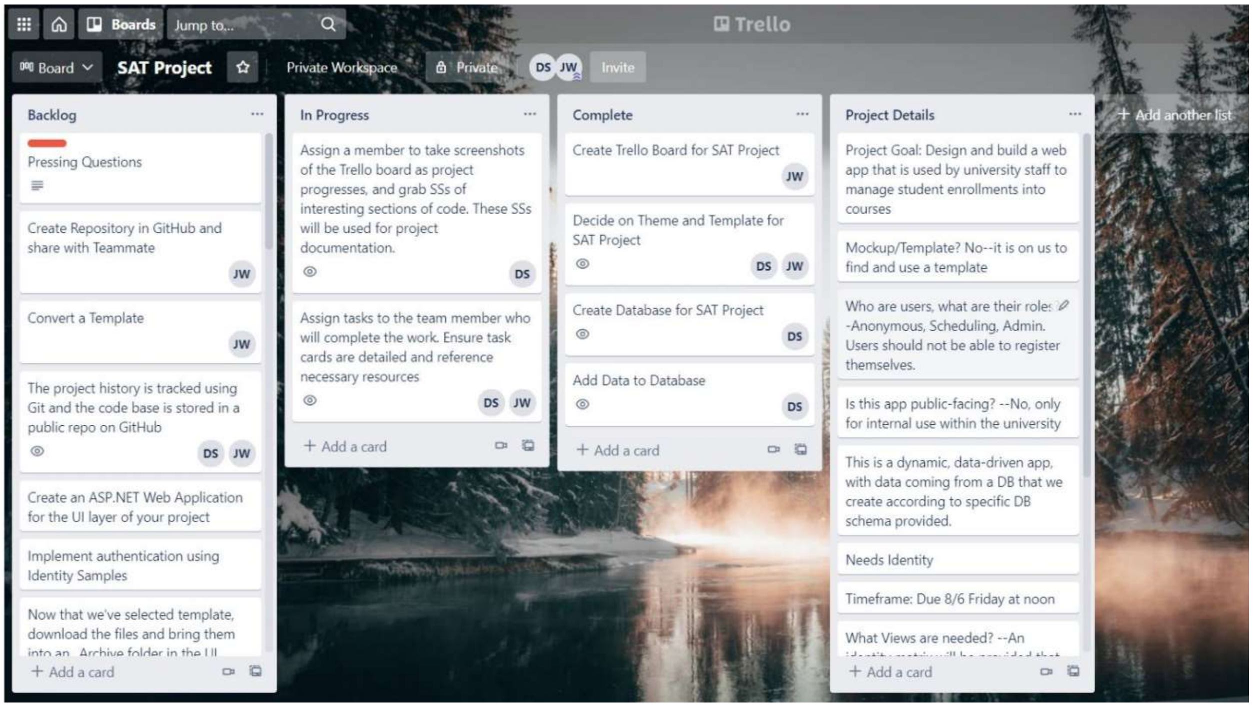
Discord

Creately

Filezilla

SmarterASP

<u>Trello Board Progression From Start To Finish:</u>



Backlog

- Pressing Questions
- Create Repository in GitHub and share with Teammate — JW
- Convert a Template — JW
- The project history is tracked using Git and the code base is stored in a public repo on GitHub — DS JW
- Create an ASP.NET Web Application for the UI layer of your project
- Implement authentication using Identity Samples
- Now that we've selected template, download the files and bring them into an Archive folder in the UI
- + Add a card

In Progress

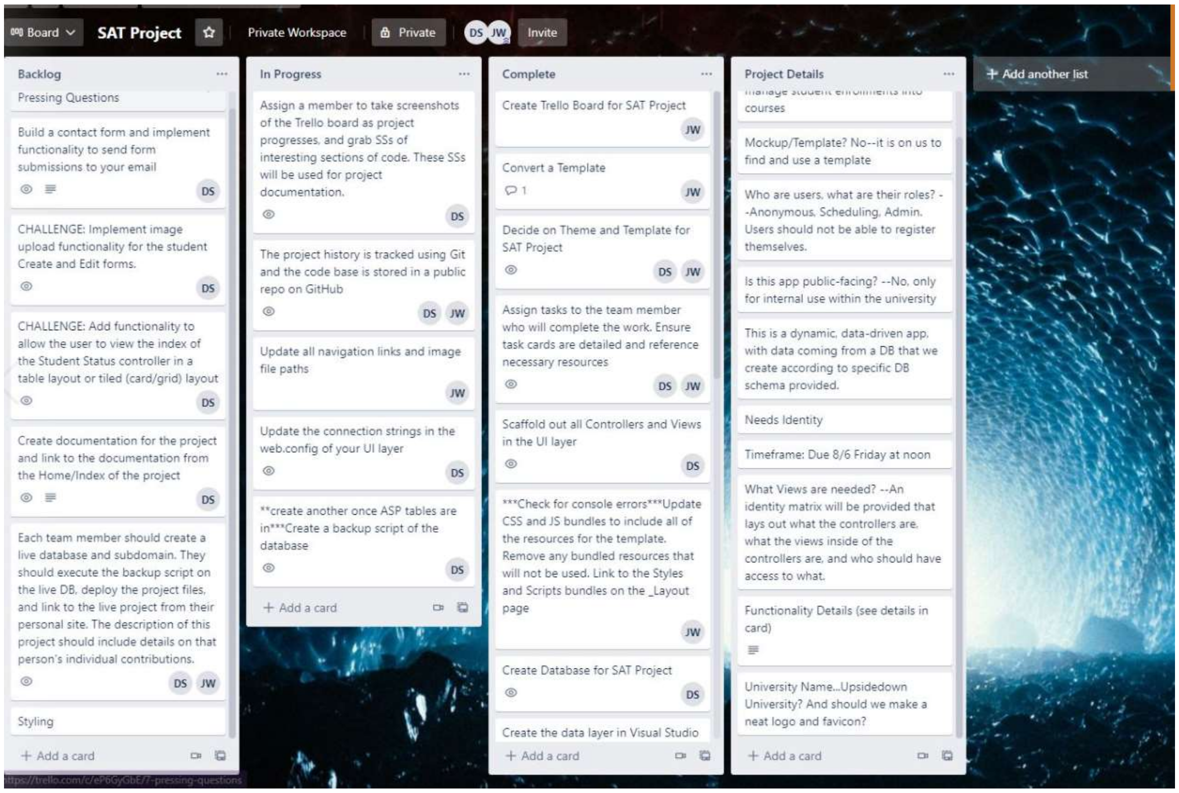
- Assign a member to take screenshots of the Trello board as project progresses, and grab SSs of interesting sections of code. These SSs will be used for project documentation. — DS
- Assign tasks to the team member who will complete the work. Ensure task cards are detailed and reference necessary resources — DS JW
- + Add a card

Complete

- Create Trello Board for SAT Project — JW
- Decide on Theme and Template for SAT Project — DS JW
- Create Database for SAT Project — DS
- Add Data to Database — DS
- + Add a card

Project Details

- Project Goal: Design and build a web app that is used by university staff to manage student enrollments into courses
- Mockup/Template? No--it is on us to find and use a template
- Who are users, what are their roles? -Anonymous, Scheduling, Admin. Users should not be able to register themselves.
- Is this app public-facing? --No, only for internal use within the university
- This is a dynamic, data-driven app, with data coming from a DB that we create according to specific DB schema provided.
- Needs Identity
- Timeframe: Due 8/6 Friday at noon
- What Views are needed? --An
- + Add a card



Backlog

- Pressing Questions
- Build a contact form and implement functionality to send form submissions to your email — DS
- CHALLENGE: Implement image upload functionality for the student Create and Edit forms. — DS
- CHALLENGE: Add functionality to allow the user to view the index of the Student Status controller in a table layout or tiled (card/grid) layout — DS
- Create documentation for the project and link to the documentation from the Home/Index of the project — DS
- Each team member should create a live database and subdomain. They should execute the backup script on the live DB, deploy the project files, and link to the live project from their personal site. The description of this project should include details on that person's individual contributions. — DS JW
- Styling
- + Add a card

In Progress

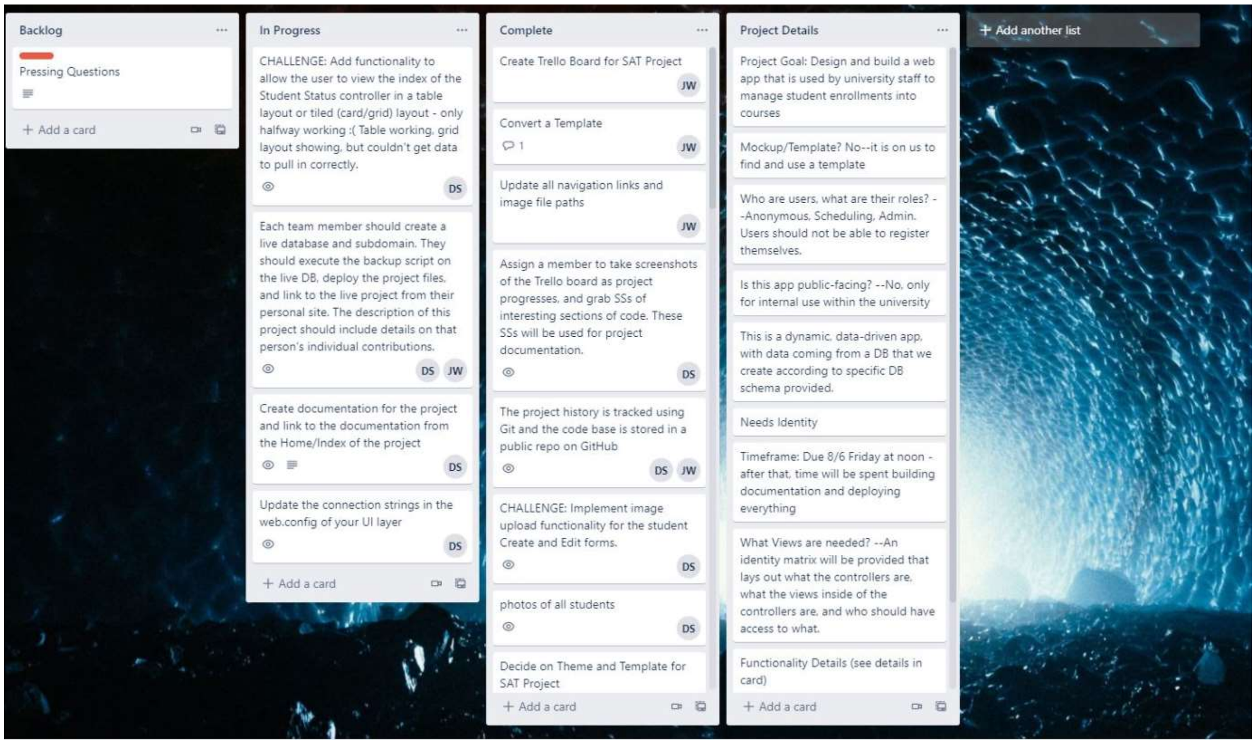
- Assign a member to take screenshots of the Trello board as project progresses, and grab SSs of interesting sections of code. These SSs will be used for project documentation. — DS
- The project history is tracked using Git and the code base is stored in a public repo on GitHub — DS JW
- Update all navigation links and image file paths — JW
- Update the connection strings in the web.config of your UI layer — DS
- **create another once ASP tables are in***Create a backup script of the database — DS
- + Add a card

Complete

- Create Trello Board for SAT Project — JW
- Convert a Template — 1 — JW
- Decide on Theme and Template for SAT Project — DS JW
- Assign tasks to the team member who will complete the work. Ensure task cards are detailed and reference necessary resources — DS JW
- Scaffold out all Controllers and Views in the UI layer — DS
- ***Check for console errors***Update CSS and JS bundles to include all of the resources for the template. Remove any bundled resources that will not be used. Link to the Styles and Scripts bundles on the _Layout page — JW
- Create Database for SAT Project — DS
- Create the data layer in Visual Studio
- + Add a card

Project Details

- manage student enrollments into courses
- Mockup/Template? No--it is on us to find and use a template
- Who are users, what are their roles? --Anonymous, Scheduling, Admin. Users should not be able to register themselves.
- Is this app public-facing? --No, only for internal use within the university
- This is a dynamic, data-driven app, with data coming from a DB that we create according to specific DB schema provided.
- Needs Identity
- Timeframe: Due 8/6 Friday at noon
- What Views are needed? --An identity matrix will be provided that lays out what the controllers are, what the views inside of the controllers are, and who should have access to what.
- Functionality Details (see details in card)
- University Name...Upsidedown University? And should we make a neat logo and favicon?
- + Add a card



Backlog

- Pressing Questions

+ Add a card

In Progress

- CHALLENGE: Add functionality to allow the user to view the index of the Student Status controller in a table layout or tiled (card/grid) layout - only halfway working :(Table working, grid layout showing, but couldn't get data to pull in correctly. — DS
- Each team member should create a live database and subdomain. They should execute the backup script on the live DB, deploy the project files, and link to the live project from their personal site. The description of this project should include details on that person's individual contributions. — DS JW
- Create documentation for the project and link to the documentation from the Home/Index of the project — DS
- Update the connection strings in the web.config of your UI layer — DS

+ Add a card

Complete

- Create Trello Board for SAT Project — JW
- Convert a Template — 1 — JW
- Update all navigation links and image file paths — JW
- Assign a member to take screenshots of the Trello board as project progresses, and grab SSs of interesting sections of code. These SSs will be used for project documentation. — DS
- The project history is tracked using Git and the code base is stored in a public repo on GitHub — DS JW
- CHALLENGE: Implement image upload functionality for the student Create and Edit forms. — DS
- photos of all students — DS
- Decide on Theme and Template for SAT Project

+ Add a card

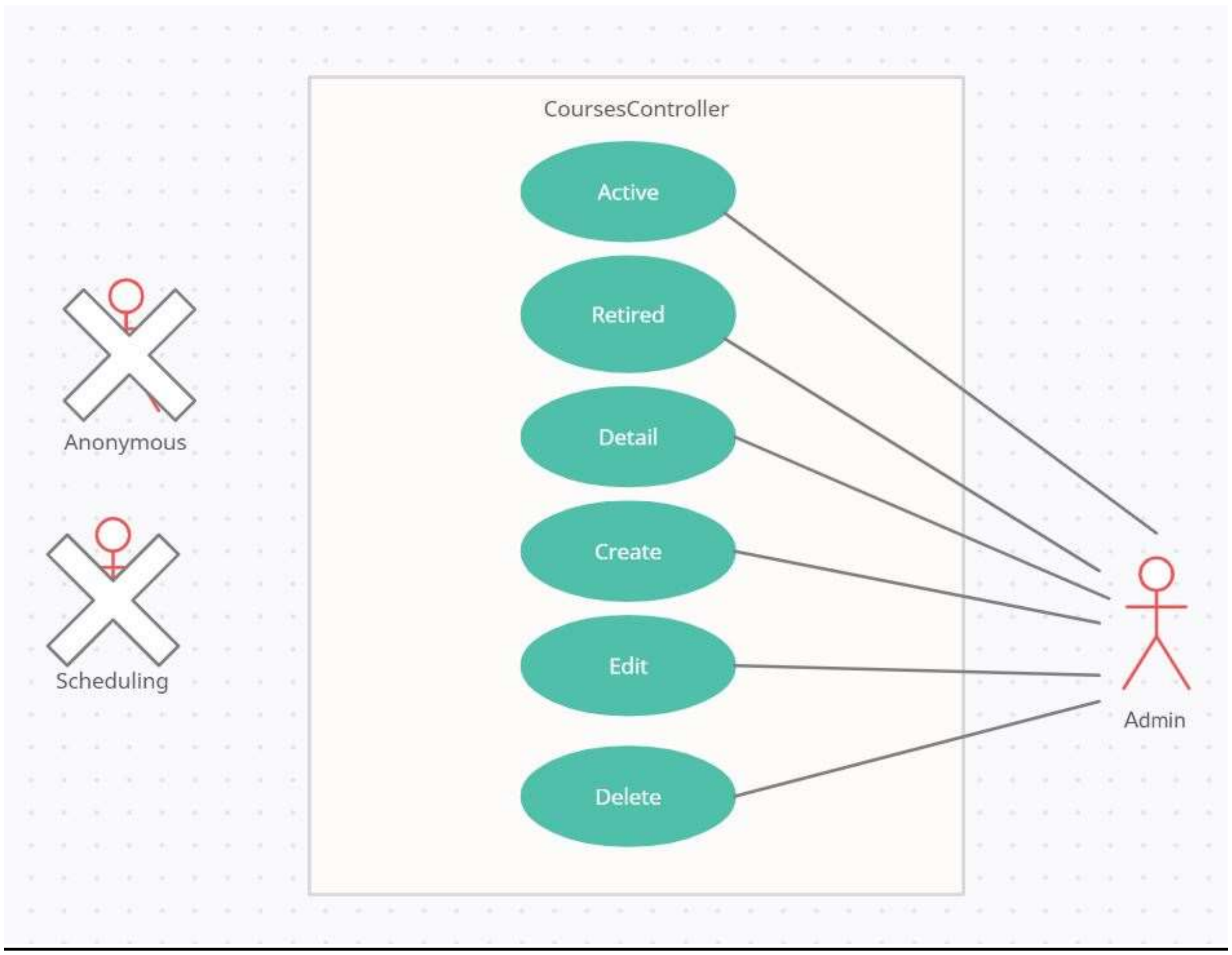
Project Details

- Project Goal: Design and build a web app that is used by university staff to manage student enrollments into courses
- Mockup/Template? No--it is on us to find and use a template
- Who are users, what are their roles? --Anonymous, Scheduling, Admin. Users should not be able to register themselves.
- Is this app public-facing? --No, only for internal use within the university
- This is a dynamic, data-driven app, with data coming from a DB that we create according to specific DB schema provided.
- Needs Identity
- Timeframe: Due 8/6 Friday at noon - after that, time will be spent building documentation and deploying everything
- What Views are needed? --An identity matrix will be provided that lays out what the controllers are, what the views inside of the controllers are, and who should have access to what.
- Functionality Details (see details in card)

+ Add a card

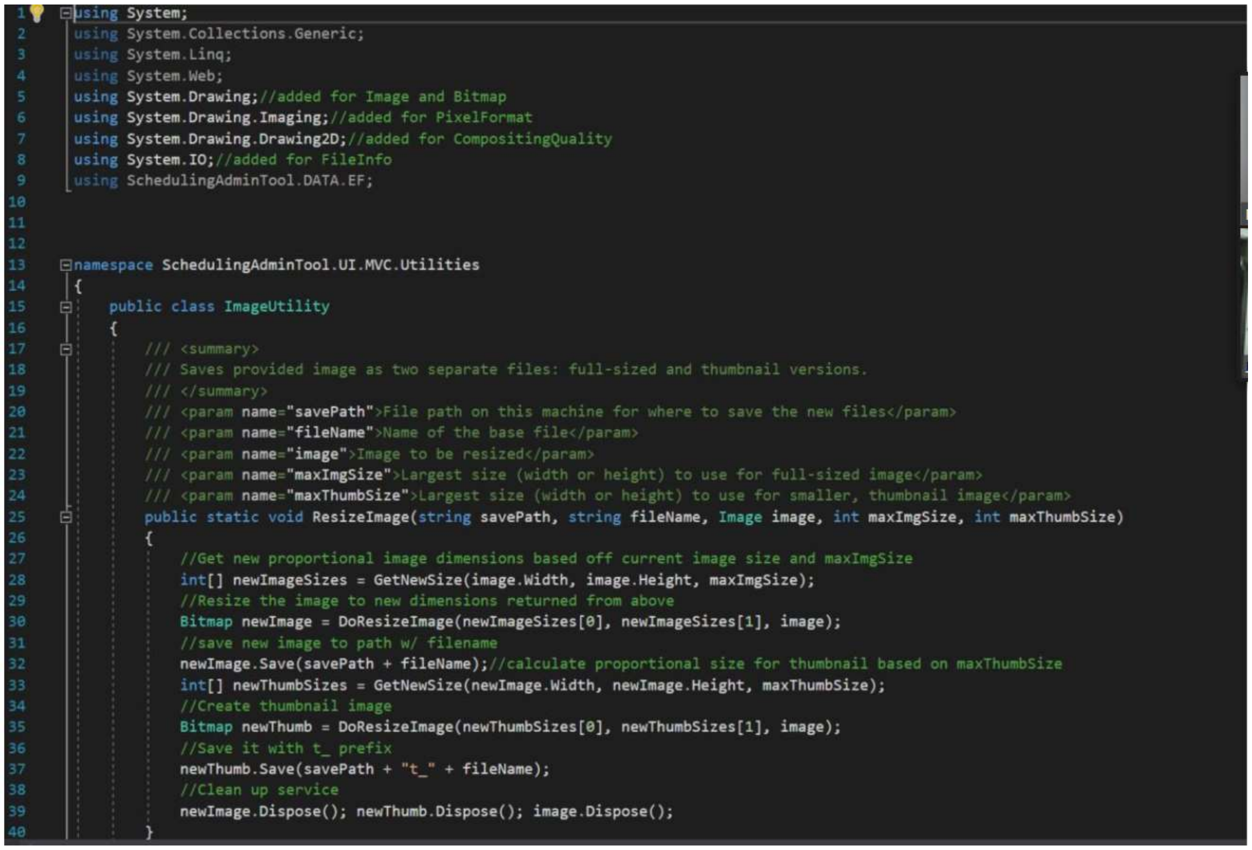
+ Add another list

Example of Using Creately Use Case Diagrams:

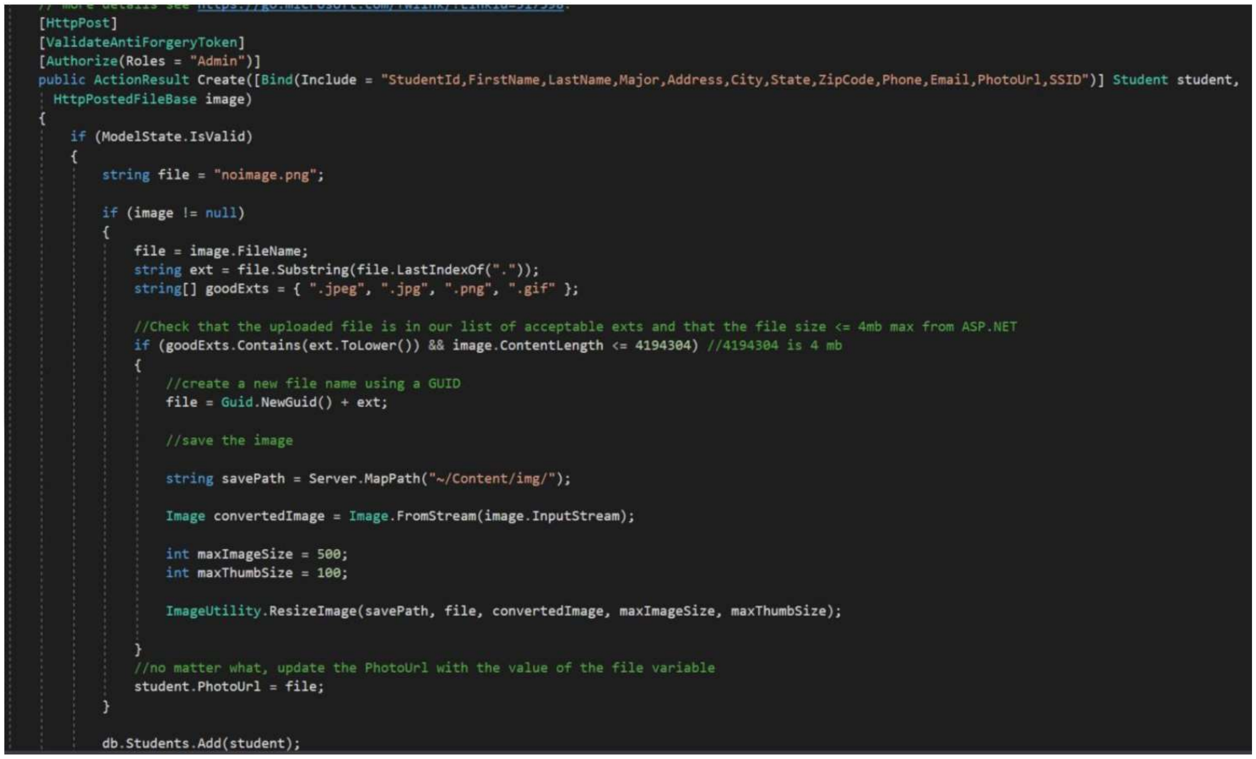


SAT Challenge Code:

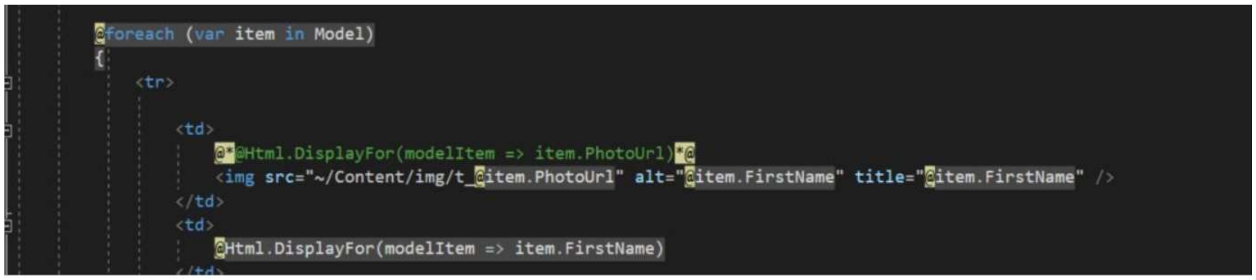
SAT Challenge #2 Lab - File Upload Screenshots



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Drawing;//added for Image and Bitmap
using System.Drawing.Imaging;//added for PixelFormat
using System.Drawing.Drawing2D;//added for CompositingQuality
using System.IO;//added for FileInfo
using SchedulingAdminTool.DATA.EF;



namespace SchedulingAdminTool.UI.MVC.Utilities
{
    public class ImageUtility
    {
        /// <summary>
        /// Saves provided image as two separate files: full-sized and thumbnail versions.
        /// </summary>
        /// <param name="savePath">File path on this machine for where to save the new files</param>
        /// <param name="fileName">Name of the base file</param>
        /// <param name="image">Image to be resized</param>
        /// <param name="maxImgSize">Largest size (width or height) to use for full-sized image</param>
        /// <param name="maxThumbSize">Largest size (width or height) to use for smaller, thumbnail image</param>
        public static void ResizeImage(string savePath, string fileName, Image image, int maxImgSize, int maxThumbSize)
        {
            //Get new proportional image dimensions based off current image size and maxImgSize
            int[] newImageSizes = GetNewSize(image.Width, image.Height, maxImgSize);
            //Resize the image to new dimensions returned from above
            Bitmap newImage = DoResizeImage(newImageSizes[0], newImageSizes[1], image);
            //save new image to path w/ filename
            newImage.Save(savePath + fileName);//calculate proportional size for thumbnail based on maxThumbSize
            int[] newThumbSizes = GetNewSize(newImage.Width, newImage.Height, maxThumbSize);
            //Create thumbnail image
            Bitmap newThumb = DoResizeImage(newThumbSizes[0], newThumbSizes[1], image);
            //Save it with t_ prefix
            newThumb.Save(savePath + "t_" + fileName);
            //Clean up service
            newImage.Dispose(); newThumb.Dispose(); image.Dispose();
        }
```



```
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Admin")]
public ActionResult Create([Bind(Include = "StudentId,FirstName,LastName,Major,Address,City,State,ZipCode,Phone,Email,PhotoUrl,SSID")] Student student,
  HttpPostedFileBase image)
{
    if (ModelState.IsValid)
    {
        string file = "noimage.png";

        if (image != null)
        {
            file = image.FileName;
            string ext = file.Substring(file.LastIndexOf("."));
            string[] goodExts = { ".jpeg", ".jpg", ".png", ".gif" };

            //Check that the uploaded file is in our list of acceptable exts and that the file size <= 4mb max from ASP.NET
            if (goodExts.Contains(ext.ToLower()) && image.ContentLength <= 4194304) //4194304 is 4 mb
            {
                //create a new file name using a GUID
                file = Guid.NewGuid() + ext;

                //save the image

                string savePath = Server.MapPath("~/Content/img/");

                Image convertedImage = Image.FromStream(image.InputStream);

                int maxImageSize = 500;
                int maxThumbSize = 100;

                ImageUtility.ResizeImage(savePath, file, convertedImage, maxImageSize, maxThumbSize);

            }
            //no matter what, update the PhotoUrl with the value of the file variable
            student.PhotoUrl = file;
        }

        db.Students.Add(student);
```



```
@foreach (var item in Model)
{
    <tr>

        <td>
            @*@Html.DisplayFor(modelItem => item.PhotoUrl)*@
            <img src="~/Content/img/t_@item.PhotoUrl" alt="@item.FirstName" title="@item.FirstName" />
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.FirstName)
```