SAT Documentation

Friday, December 17, 2021   12:57 PM
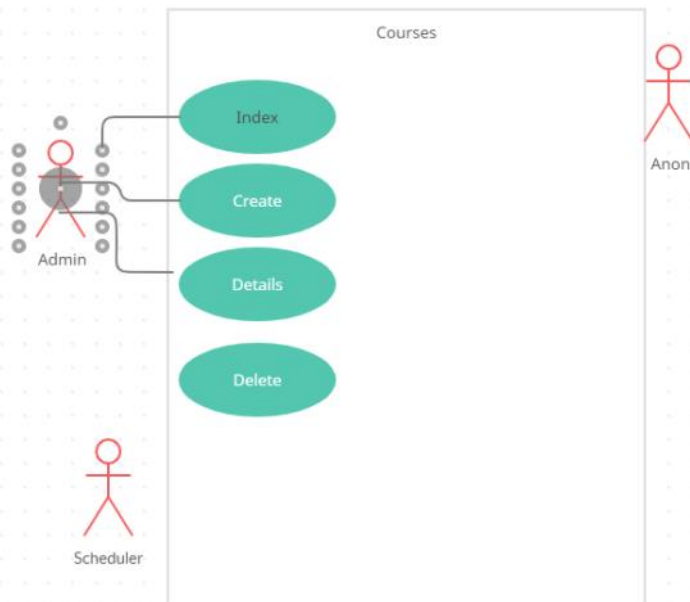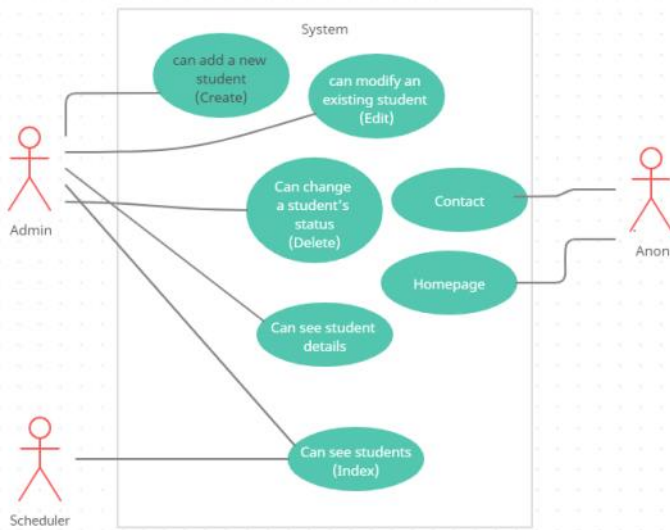
Joshua Owens & Amber Mills
Dev Team Jomber
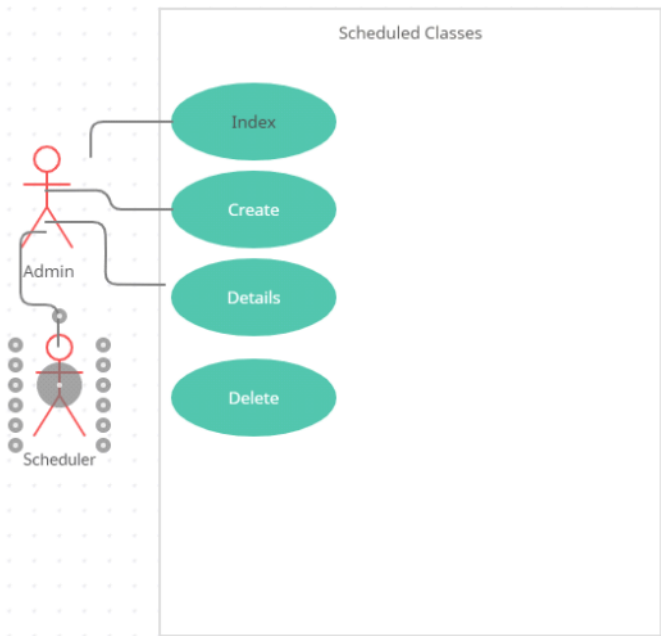This application is a scheduling administration tool. It can be used by three different users.
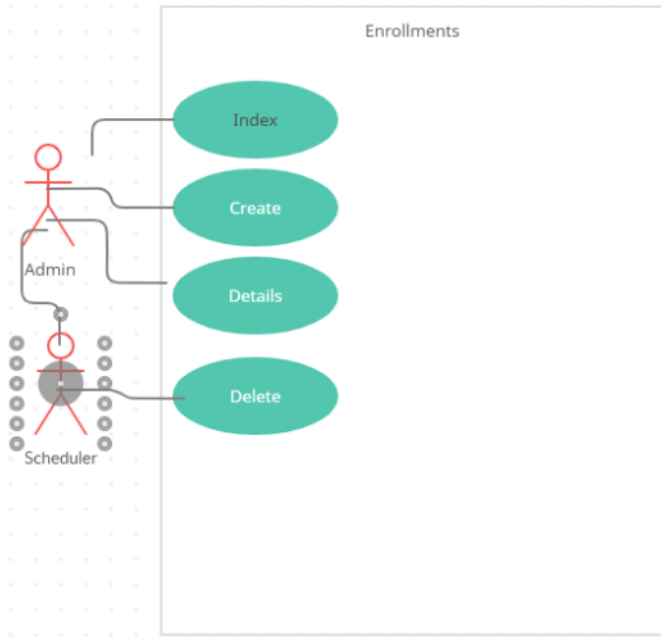The main functionality is for a scheduler to be able to add students to classes and update student information as needed

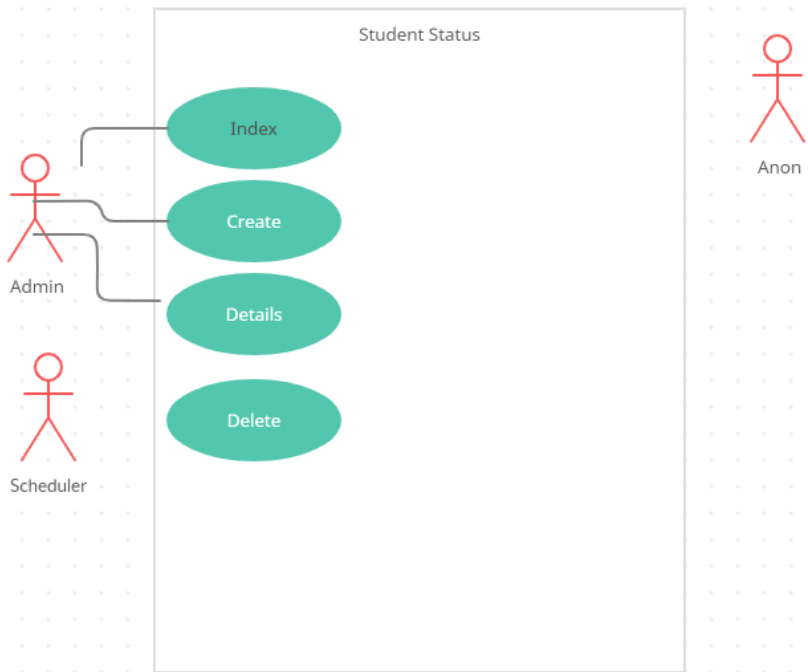The admin has access to maintaining the application

Anon || Admin || Scheduler

The anon has the least amount of access to the scheduling tool. They are only able to see the homepage or contact.
Certain users have access to certain pages

Enrollments

- Index
- Create
- Details
- Delete

Anon

Admin

Scheduler

Scheduled Classes

- Index
- Create
- Details
- Delete

Anon

Admin

Scheduler

Students

- Index
- Create
- Details
- Delete

Admin

Scheduler

Anon

Student Status

- Index
- Create
- Details
- Delete

Admin
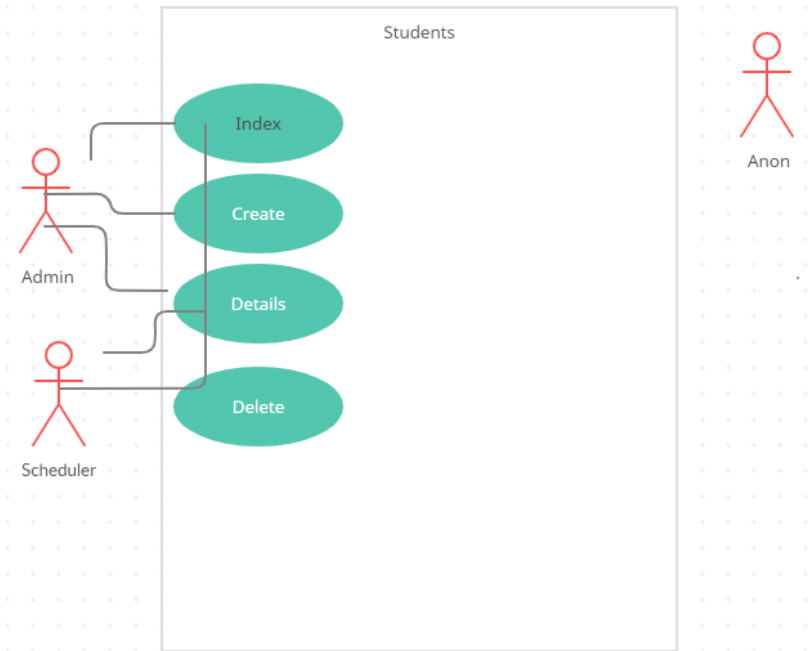
Scheduler

Anon

Admin
Scheduling

Anon

```
namespace SAT_Pair_Project.Utilities
{
    public class ImageUtility
    {

        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
```
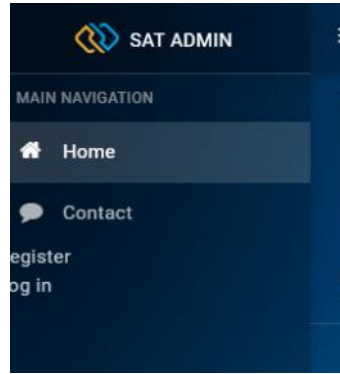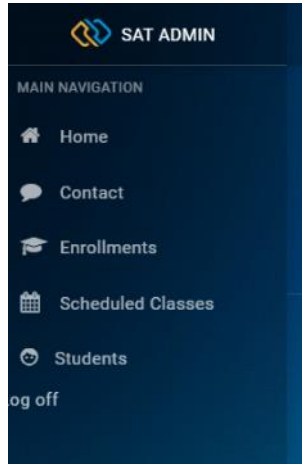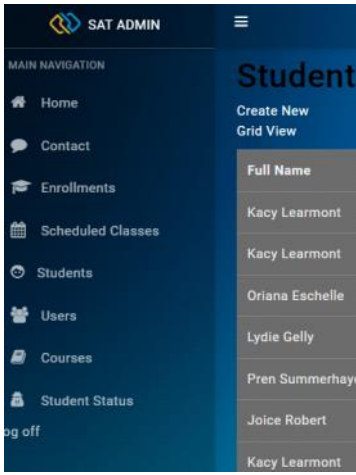
We updated students controller to implement file upload
We initiated under Create, Edit, Delete Views

We had to add a new view in the Students Controller
We created a Students Grid View to easily see student details

Enrollment page allows us to select the student and the class the student will be enrolled in

We locked down access to views in the Controllers
We implemented Authorize Roles to control access per role

During our process, one person was the driver and one person was the navigator

We added authorize statements to views that needed to be controlled

Amber Done

- Started Dummy Data
- Picked Template
- Added to Source Control
- source control
  ☑ 3/3
- Build simple 2 tier project structure
  ☑ 4/4
- Run identity samples in the UI layer
  ☑ 5/5
- Find and convert template
- EF in Data Layer
  ☑ 3/3
- Customize view in UI Layer
- img upload
  ☑ 1/1
- updated the nav in the _Layout
- contact view
- + Add a card

Josh Done

- Finished Tables
  JO
- Edited crappy Mockaroo data
- Create database
  ☑ 1/1
- Scaffold controllers and views
  ☑ 5/5
- Lock down the controllers
- Create metadata classes
  ☑ 2/2
- Lock down the views
- Displayed users in the nav
- Custom properties from metadata
  ☑ 2/2
- Implementations
  ☑ 2/3
- student grid view
- customizations
- + Add a card