SAT

Created by David Muolo and Rachel Mantei

Requirements: Create a secure application for managing product data. Application is built to simulate an online class scheduling system. Administrators will have the ability to manage students, courses, scheduled classes, and enrollment. We were given 2 days to finish this project.

Anonymous, Admin, and Scheduling Access Log requested by Client

HomeController	Index	Contact
Anonymous	x	x
Scheduling	x	x
Admin	x	x

Courses	Active	Retired	Detail	Create	Edit	Delete
Anonymous						
Scheduling						
Admin	x	x	x	x	x	x

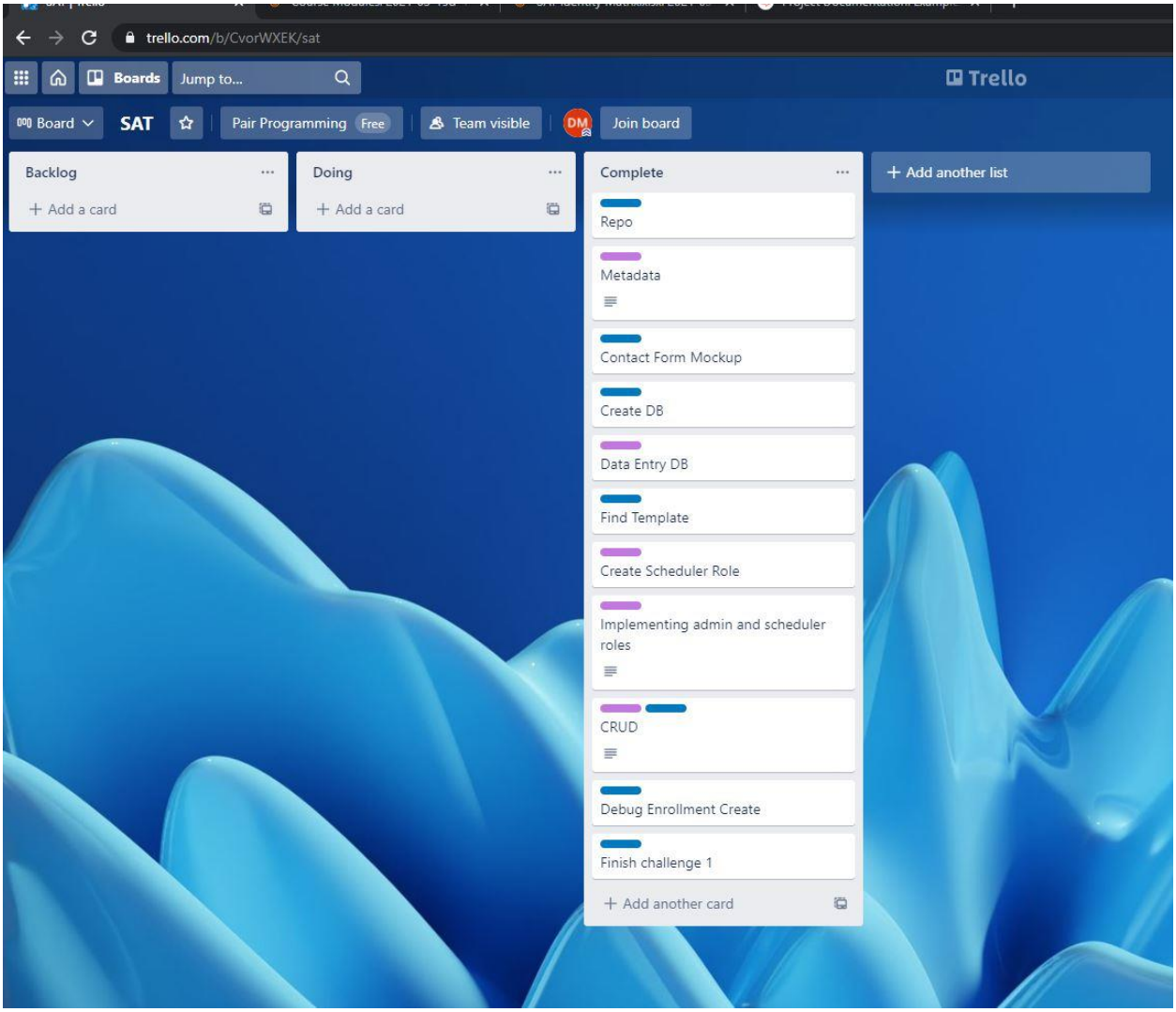
Enrollments	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x	x	x	x
Admin	x	x	x	x	x

Scheduled Classes	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x	x	x	
Admin	x	x	x	x	x

Students	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x			
Admin	x	x	x	x	x

Student Statuses	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling					
Admin	x	x	x	x	x

Trello Board



Required field given by client

Through the *normalization process* of building our database, we decided to store a student's first name and last name as two separate fields. This has caused the process of enrolling a student in a scheduled class slightly more difficult as when we enroll a student, we choose from a dropdown of first names. Since we don't store calculated fields in a database, we can create a custom property in our partial *(buddy)* class to represent the full name of a student instead of just their first name and apply metadata to the property.

1. Discuss with your pair programming group why calculated fields are not stored in a database.
2. Build a custom property that combines a student's first name and last name.
3. Utilize in the Student Index, Details, and Delete Views

When we used EF to build the controller and views, the drop-down list for Scheduled Classes displays the *InstructorName* by default, but that isn't very descriptive. Instead we could create a custom property that provides a summary of class information to enroll a student properly.

1. Build a custom property in the ScheduledClassMetadata file that combines the StartDate, CourseName and Location information.
2. Utilize the custom property in the EnrollementsController to populate the data into the dropdown list for ScheduledClassID.

Requirement given by client

Students will be grouped into small teams. Each group will complete the following in the SAT database:

- Add any remaining tables.
- Define the PK-FK relationships.
- Populate test data (3-10 records in each table).

Teams should specifically enter the following data for the Scheduled Class Status and Student Status.

Scheduled Class Status

SCSID	SCSName
1	Active
2	Pending
3	Completed
4	Cancelled

Student Status

SSID	SSName	SSDescription
1	Prospect	A student who has inquired about taking classes at the institution.
2	Current Student	A student who is actively participating in classes.
3	Former Student-Withdrawn	A student who has left the institution on their own accord.
4	Former Student-Dismissed	A student who has left the institution at our request.
5	Alumni	A student who has finished their coursework and left the institution.
6	Booted	This is what happens when you don't go to class.

Keep in mind that students will be providing documentation at the end of the week. It is suggested that you take any screenshots (using Windows "snipping tool") and save them (with a name that you'll know) to a specific location.

Suggested: Create a folder on your desktop for SAT Documentation OR open up a Word document and copy/paste the screenshots there (while returning later to add summaries and additional documentation).

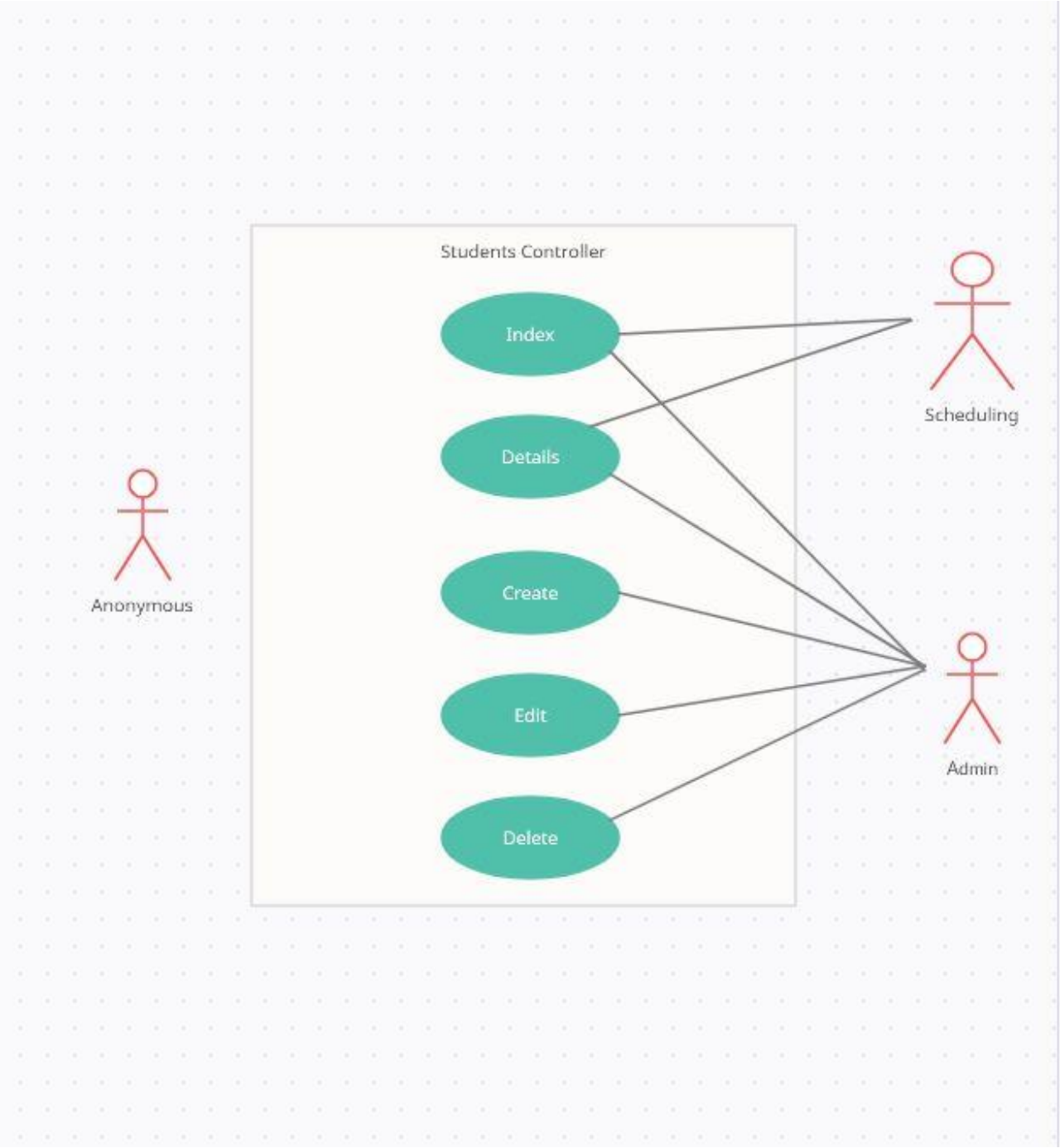
Each team will setup the SAT project, choose a template, and convert it. The steps below should be completed by the driver for this portion of the project.

- Select a template for the SAT project. Use the following template considerations as a guide.
  - Side nav (can be used for log in functionality)
  - Simple HTML/CSS structure.
- Create a new public repository on Github and invite your teammates to be collaborators.
- Clone that repo onto your machine and add a blank solution into the repo.
- Add an ASP.NET Web Application project for the UI layer. Select the MVC template and change authentication to individual user accounts.
- Run Identity Samples on the UI layer.
- Update the default connection string in the web.config file.
- Launch the project and login using the default admin credentials. Add new admin user, create *Scheduling* role, add *Scheduling* user
- Delete the default Admin after testing new Admin
- Commit and push changes
- Create the _Archive folder and add template files to the project.
- Convert the index.html to the _Layout.
- Commit and push changes
- Take some screenshots for documentation and update Trello accordingly.

Recommended: Use the BundleConfig for styling and scripts.

NOTE: All other team members will need to pull down the updated repo. They will need to launch the project and login with the default admin credentials. They will also need to add a new admin user, create the Scheduling role, add a scheduling user, and delete the default admin user (after testing the new admin user).

Scheduler access is also enabled when logging in under scheduler credentials. This allows access to the Index and Details version of Student Controller.



Currently the application is functioning and meets all requirements that includes CRUD functionality