0. Title: Schedules4U

1. Who: Merrick Oleszek (Github: @Merricko24), Xiaobo Gonaver (Github: @x1aobog), Adlai Kohn (@adko9071), Calvin Hawks (@CalvinHawks), Jaqueline Serrano (@Jackie995), Coleman Caldwell (@ccald27)

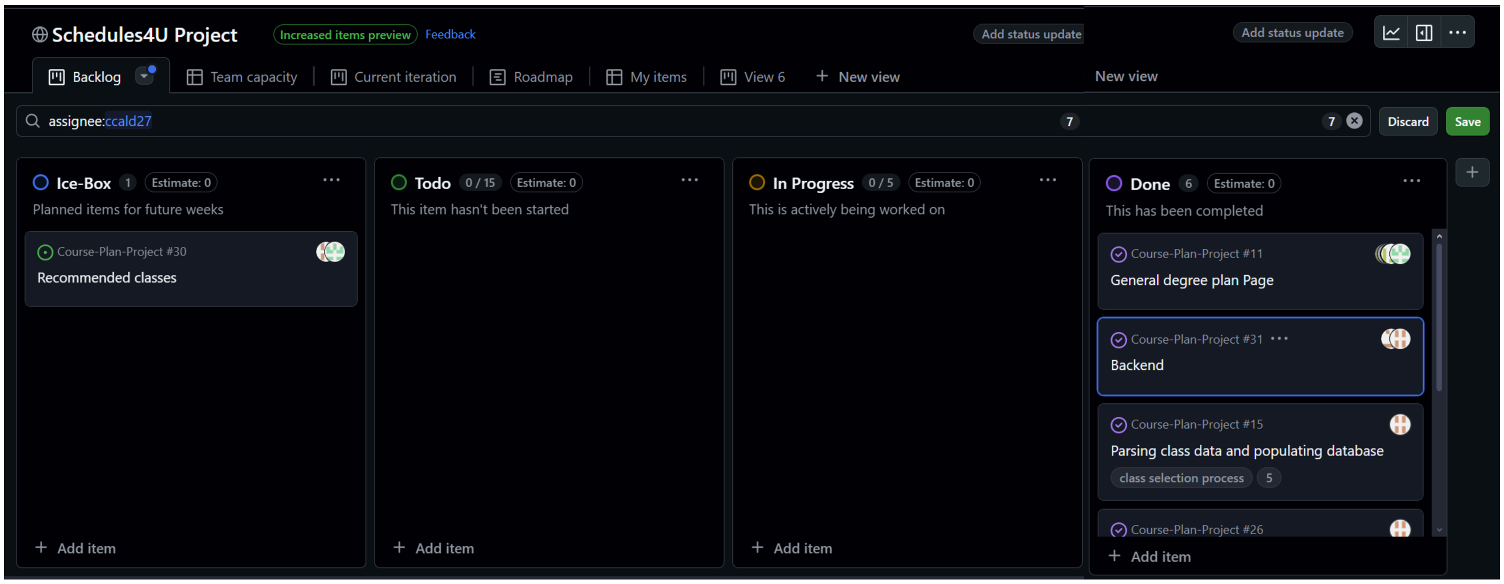
2. Project Description:

Schedules4U is a dynamic academic planning tool designed for computer science students and advisors at the University of Colorado Boulder. The platform streamlines the process of creating personalized and effective semester schedules, promoting greater student success through an easy-to-use, intuitive interface. Students can quickly build their course schedules with minimal effort, while also being matched with an advisor who provides tailored guidance throughout their academic journey. Advisors can view and manage student plans, offer course recommendations through personalized notes forwarded to each student, and ensure students stay on track to achieve their academic studies and career goals.

To build Schedules4U, we utilized a custom web scraper to gather up-to-date computer science class data, such as class descriptions, credits, and times, directly from university resources, ensuring the information is current and accurate. The website was developed using HTML, CSS, and JavaScript to create a responsive and user-friendly experience. By combining real-time data with interactive design, Schedules4U fosters a more proactive and collaborative approach to academic planning. The system enhances both the scheduling and advising process, creating a more efficient, supportive environment for students and advisors alike. Ultimately, Schedules4U aims to make academic planning accessible, straightforward, and empowering for the CU Boulder computer science community.

3. Project Tracker:

GitHub project board: https://github.com/users/Merricko24/projects/2


Schedules4U Project

assignee:ccald27 7 7

Ice-Box 1 Estimate: 0
Planned items for future weeks
- Course-Plan-Project #30 Recommended classes

Todo 0 / 15 Estimate: 0
This item hasn't been started

In Progress 0 / 5 Estimate: 0
This is actively being worked on

Done 6 Estimate: 0
This has been completed
- Course-Plan-Project #11 General degree plan Page
- Course-Plan-Project #31 Backend
- Course-Plan-Project #15 Parsing class data and populating database — class selection process 5
- Course-Plan-Project #26

4. Video:

[Video Demo Link (YouTube)](#)

[Video Demo Link (github)](#)

5. VCS:

Link: https://github.com/Merricko24/Course-Plan-Project/tree/main

6. Contributions:

Jaqueline - I contributed to the bareback register page using HTML and CSS. I added test data to the PostgreSQL database to test advising notes feature and ensure courses had prerequisites. I built a simple feature for students to add semester containers on the schedule page using HTML and JavaScript. I made small fixes to how classes display on the advisor page, ensuring database updates with JavaScript, Node.js, and Handlebars.I also implemented the display of advisor notes on the student page and made minor adjustments to the functionality on the advisor page using HTML, JavaScript, Node.js, PostgreSQL, and Handlebars.

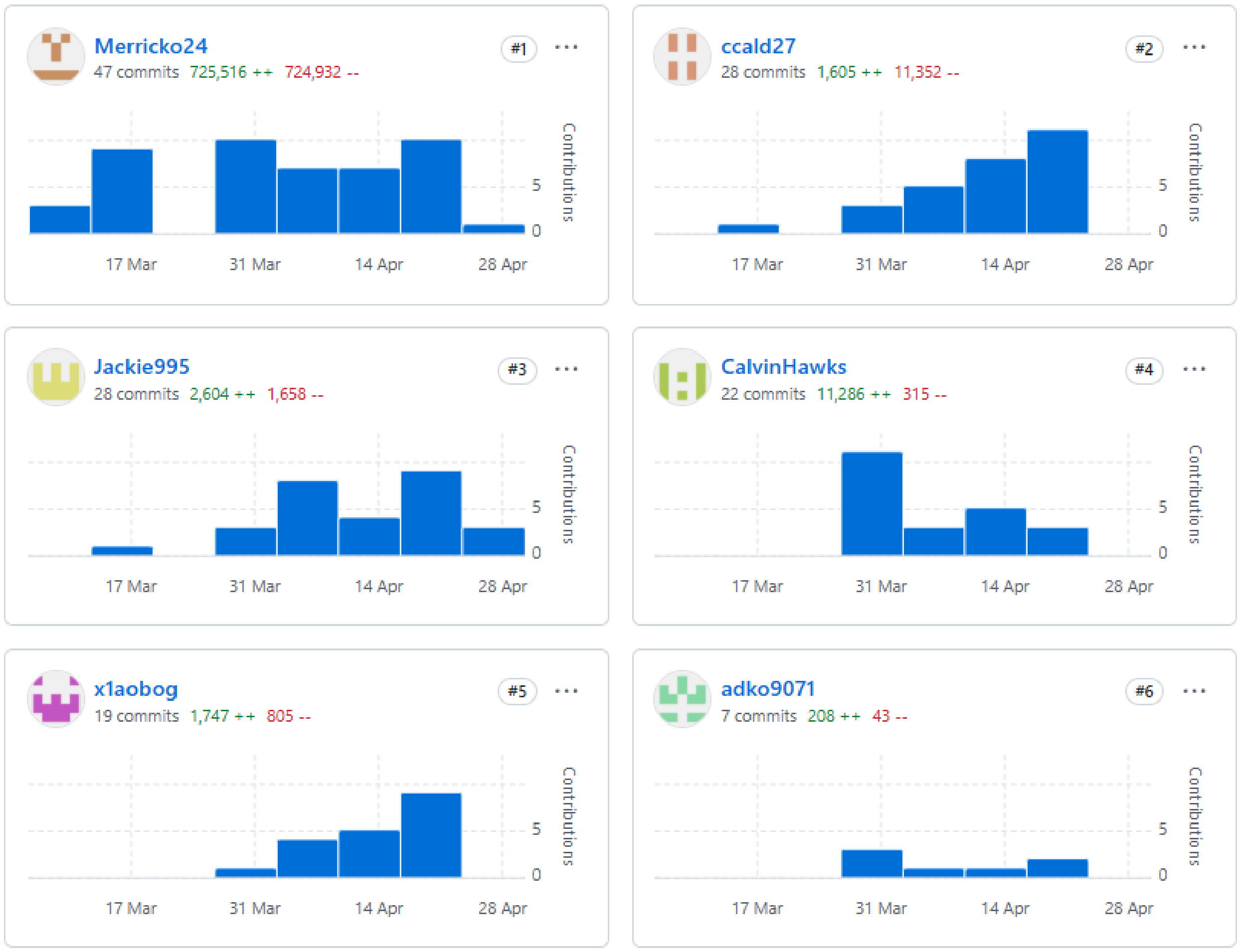
Xiaobo - I built the user interface for the advisor scheduling page using HTML and CSS. I worked on getting the advisor notes to sync with the student page in the HTML file and in JavaScript. I ended up getting this to work so that the advisor could send and save personalized notes to each of their respective students. I also helped with trying to integrate the database so that the advisor could access each student and their unique class schedule and personal information. I also built the logout page in HTML and routed it in JavaScript in our index.js file.

Calvin- I contributed the idea. I worked on the main schedule page for the student, getting the classes to display properly. I worked on the backend for getting the site to load properly with docker/postgres. A big contribution was implementing the D3 library for the drag and drop feature to be able to move classes around in the student side and update the information in the database. I also implemented the class remove feature, making a button on each of the classes to be able to remove if from the users schedule.

Coleman- I contributed to most of the database work and connecting the database to the backend. I first focused on connecting the user login and registry routes to the database, and storing the user in the session regardless of where they were on the site. Additionally, I focused wherever needed to connect the database and backend routes to the physical pages of the student/advisor schedules to make or assist with things like bug fixes, class display, class search, class add. I worked lots on the class Add and Search, but got help from all of my peers and tried to help on other issues if there was urgent help needed. I helped somewhat on the Advisor Schedule and the class scraper, however was not able to finish assisting the advisor schedule or the scraper in time for presentation.

Merrick- I first focused on all of the visuals in the html in the register, login, home, and all of the partial pages. I also designed the logo we used in every page of the website. I did a good chunk of the stylesheet, making sure the buttons and nav bar and other features were the same across every page. Additionally, I helped with implementing the advisor notes and doing the popup that shows up with class info when hovering over any class on the schedule. Finally, my last two contributions were making sure students could not add more than one of the same class, and I did lab 13, getting render to work with our repository.

Adlai- I first focused on mocking up every webpage of the website into online wireframes, with particular attention towards the advisor aspects of the page. Then, I focused on the layout of the main schedules page, creating the first html display of the page. I tried to work on the logic for the colored lines between corresponding classes, but the libraries I was working with were too unfamiliar and I couldn't get them to work. I helped a little bit on the search classes html and logic, and I made the first implementation of the recommended class search inside the class search page, but due to time constraints I wasn't able to get the logic working on that and it was ultimately phased out of the final product.



Merricko24 #1
47 commits 725,516 ++ 724,932 --

ccald27 #2
28 commits 1,605 ++ 11,352 --

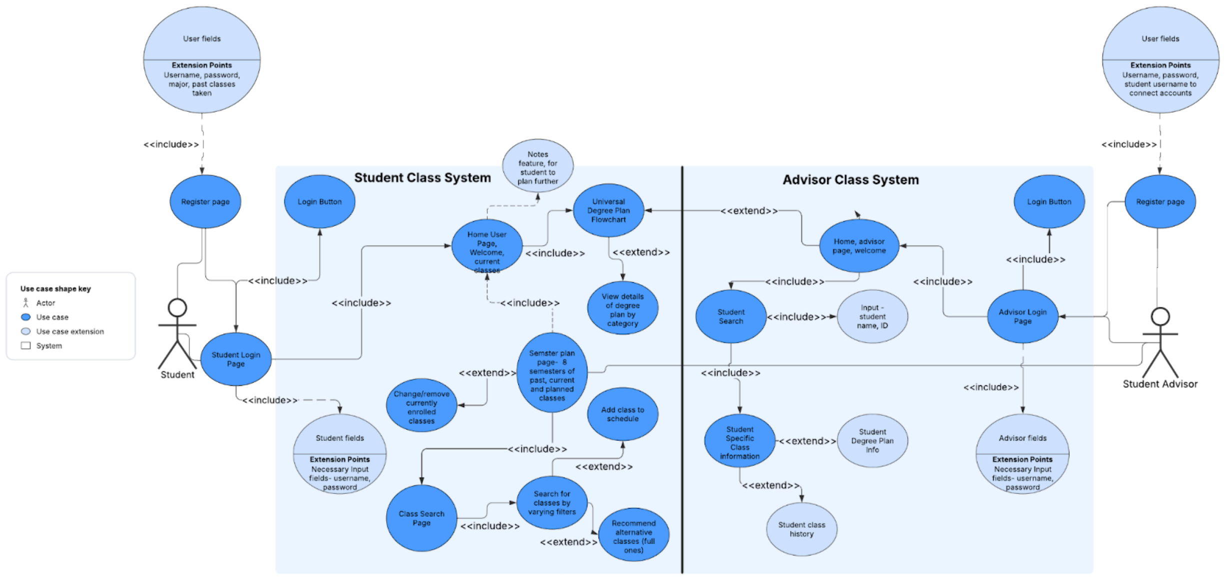
Jackie995 #3
28 commits 2,604 ++ 1,658 --

CalvinHawks #4
22 commits 11,286 ++ 315 --

x1aobog #5
19 commits 1,747 ++ 805 --

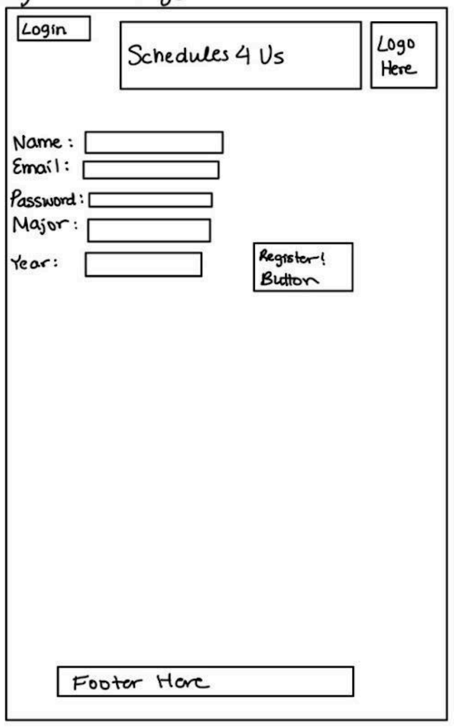
adko9071 #6
7 commits 208 ++ 43 --

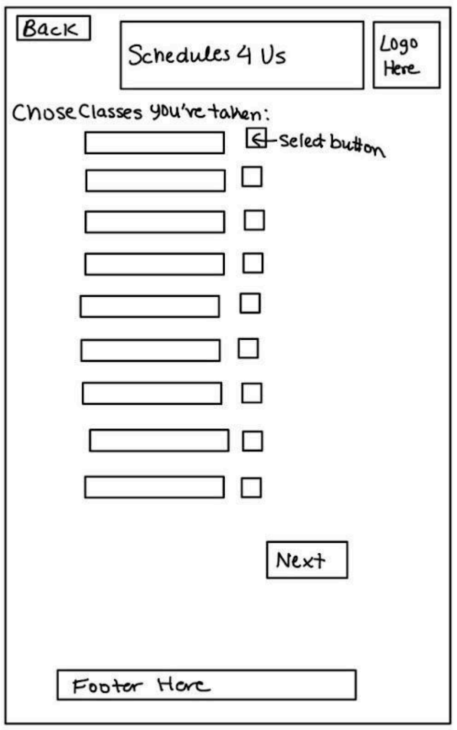
7. Use Case Diagram:

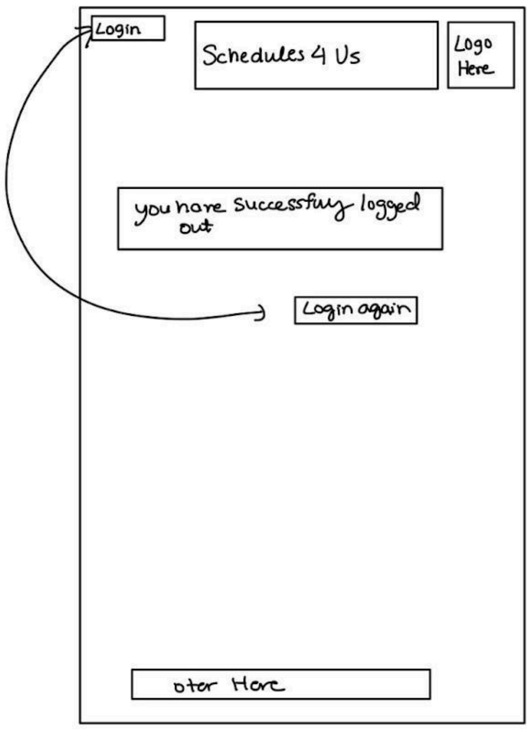

User fields
Extension Points
Username, password, major, past classes taken
<<include>>
Register page
Student
Use case shape key
Actor
● Use case
● Use case extension
▭ System
Student Class System
Login Button
Notes feature, for student to plan further
Home User Page, Welcome, current classes
Universal Degree Plan Flowchart
View details of degree plan by category
Student Login Page
Semster plan page- 8 semesters of past, current and planned classes
Change/remove currently enrolled classes
Add class to schedule
Student fields
Extension Points
Necessary Input fields- username, password
Class Search Page
Search for classes by varying filters
Recommend alternative classes (full ones)
<<include>>
<<extend>>
Advisor Class System
Home, advisor page, welcome
Student Search
Input - student name, ID
Advisor Login Page
Login Button
Register page
User fields
Extension Points
Username, password, student username to connect accounts
Student Specific Class information
Student Degree Plan Info
Student class history
Advisor fields
Extension Points
Necessary Input fields- username, password
Student Advisor

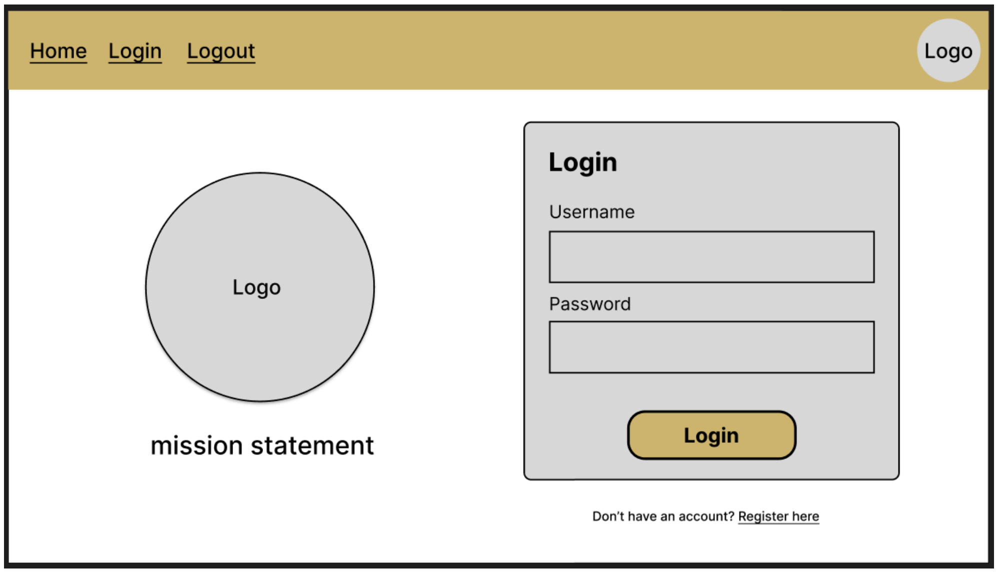
8. Wireframes:

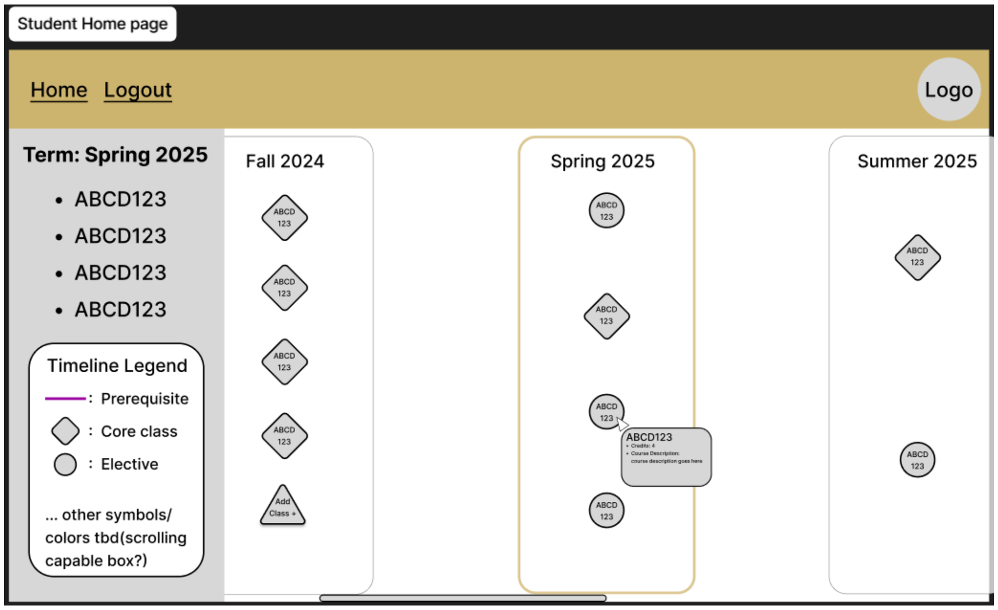
Registration Page

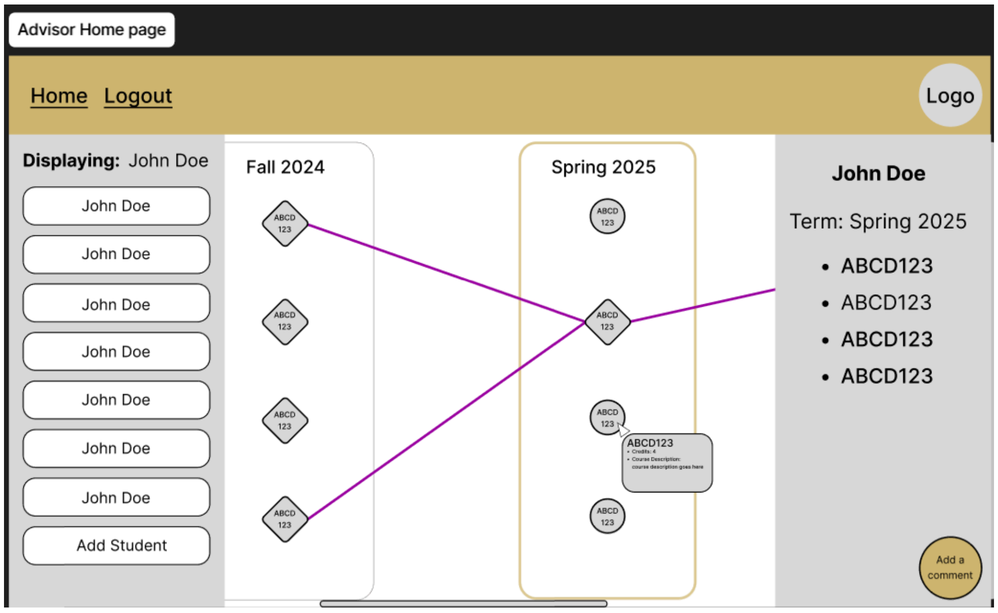







Home
Login
Logout
Logo
Logo
mission statement
Login
Username
Password
Login
Don't have an account? Register here


Student Home page
Home
Logout
Logo
Term: Spring 2025
• ABCD123
• ABCD123
• ABCD123
• ABCD123
Timeline Legend
: Prerequisite
: Core class
: Elective
... other symbols/ colors tbd(scrolling capable box?)
Fall 2024
ABCD 123
ABCD 123
ABCD 123
ABCD 123
Add Class +
Spring 2025
ABCD 123
ABCD 123
ABCD 123
ABCD123
• Credits: 4
• Course Description: course description goes here
ABCD 123
Summer 2025
ABCD 123
ABCD 123


Advisor Home page
Home
Logout
Logo
Displaying: John Doe
John Doe
John Doe
John Doe
John Doe
John Doe
John Doe
John Doe
Add Student
Fall 2024
ABCD 123
ABCD 123
ABCD 123
ABCD 123
Spring 2025
ABCD 123
ABCD 123
ABCD 123
ABCD123
• Credits: 4
• Course Description: course description goes here
ABCD 123
John Doe
Term: Spring 2025
• ABCD123
• ABCD123
• ABCD123
• ABCD123
Add a comment

9. Test results:

The three user acceptance tests were registering a username and password, logging in and logging out, and class registration.

The first use case tested user registration with username and password. My friend navigated to the registration page and provided their details, including name, identikey, and password. Upon submission, their account information was saved to the database, and they were able to log into the account afterward as expected. The only issue we encountered was with the identikey format: initially, we had not clearly specified that identikeys must consist of four letters followed by four numbers ("abcd1234"). To resolve this, we added an example format on the registration page to guide users in entering a valid identikey. The third use case tested the users ability to log in to view student schedules and leave notes. My friend logged in as an advisor, viewed the recently added classes on the advisor page, and successfully wrote and saved a note for the student. The functionality worked as expected, but there was no indicator that the note had been successfully saved to the database. To address this, we implemented a pop-up confirmation that now appears when an advisor saves a message. The third use case validated the logout functionality for both student and advisor accounts. My friend was able to log out completely from either type of account. After logging out, the application correctly provided a link for users to return to the login page. The use case was completed without any issues, and no major suggestions for improvements were made for this feature. In the fourth use case, the user searched for and added classes to their schedule. My friend opened the search modal, typed in a class name, and added it, which correctly updated her schedule. The functionality worked as intended. However, she suggested including time-slot information to visually map out when each class would take place during the week. Although this feature was not implemented in the current version, it would be a valuable enhancement for future development.

10. Deployment: Link to deployment environment or a written description of how the app was deployed and how one might access/run the app. The app must be live, working, and accessible to your TA.

The link below will bring you to our website where you can access our website in its entirety through the website render. Click here