

Class Scheduler

Lewis Callaway (Client Point of Contact), Carter Everts (Testing Lead), Michael Less (Tech Lead), Isaiah Ortiola (Component Designer), Brian Schomer (Project Manager), Simeon Steward (Project Lead)

Client: Vicky Thorland-Oster and Tina Prouty

Advisor: Dr. Tom Daniels

Problem Introduction

- **Project Objective:**

- Build a desktop application to streamline the scheduling of newly introduced classes for the ECpE Office

- **Background:**

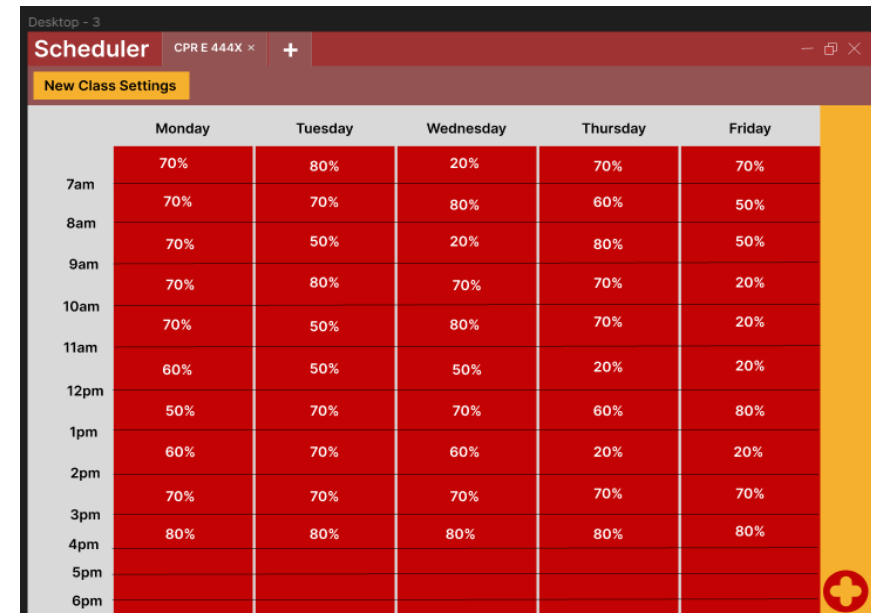
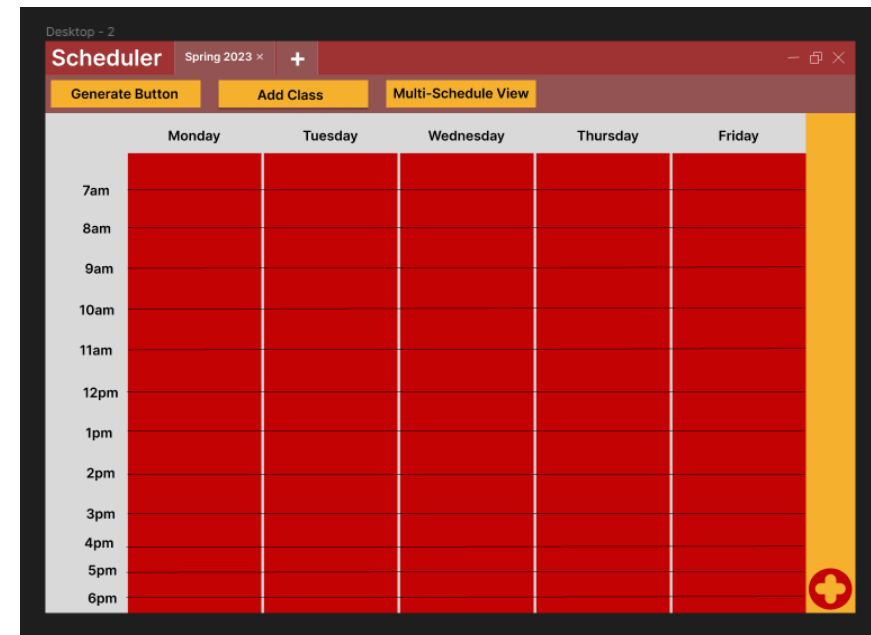
- Current System: Manual guess and check work by the advisors, using Excel and physical paper
- Current software like AccessPlus and Workday do not do this

- **Solution Plan:**

- Figure the best time for the new class based on how many target students are expected to be available
- Do this based on what semester the class is targeting and comparing the classes that semester of student is expected to take
- Our solution will be a best estimate, not exact
- Meant to help the advisors make informed decisions, not make it for them

Our Process

- Had vague understanding of the project
- Met with clients to better understand the problem
 - Got understanding of their problem
 - Began to brainstorm solutions
- Spent ample time planning our solution
- Met with clients again
 - Showed them our mock-up of the UI and application
 - Got their opinions on our plan



Our Process

- Began development and discovered an oversight
 - Algorithm to sort availability
- Met with faculty advisor for advice
- Began working on the algorithm and furthered development
- Got a working prototype and had a demo for the clients
- Got feedback
 - Fixed displayed times
 - Removed weekend slots from schedule
 - Standardized times
- Implemented their feedback and completed the application

Screenshot of the Class Scheduler application. The interface shows a grid of class settings for Monday, Wednesday, and Friday. The rows represent time slots from 7:45 am to 5:30 pm. Each cell in the grid contains the text "Add Constraints in Class Settings".

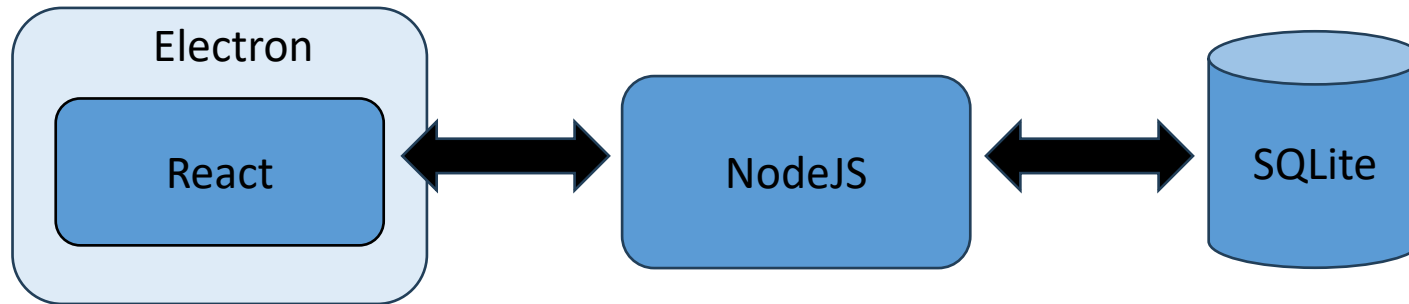
	Monday	Wednesday	Friday
7:45 am	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
8:50 am	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
9:55 am	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
11:00 am	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
12:05 pm	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
1:10 pm	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
2:15 pm	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
3:20 pm	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
4:25 pm	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings
5:30 pm	Add Constraints in Class Settings	Add Constraints in Class Settings	Add Constraints in Class Settings

Screenshot of the Class Scheduler application showing a grid of class settings with numerical values for Monday, Wednesday, and Friday. The rows represent time slots from 7:45 am to 5:30 pm. The values are displayed in the grid cells.

	Monday	Wednesday	Friday
7:45 am	100	100	100
8:50 am	100	100	100
9:55 am	100	100	100
11:00 am	100	100	66.67
12:05 pm	100	90	42.38
1:10 pm	100	90	75.71
2:15 pm	100	90	75.71
3:20 pm	100	90	90
4:25 pm	0	0	0
5:30 pm	100	75.71	100

Our Build

- We used React for our frontend
- NodeJS was used for the backend
- Electron was used as a wrapper to turn the webapp into a desktop app
- In memory SQLite Database to store class, section and cohort info

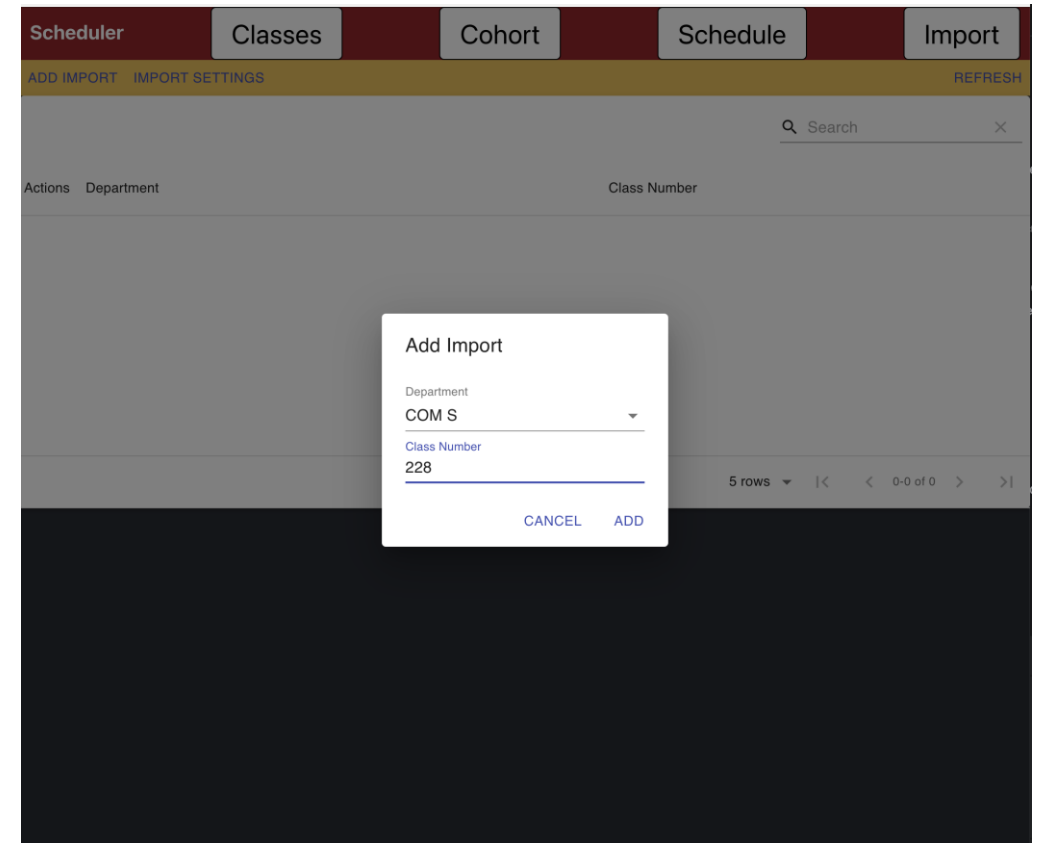


Importing Classes

- Since the algorithm takes in classes, we need a list of all the section times for a class
- Some classes can have upwards of 20 sections
- Labs and recitations also affect how scheduling work

Importing Classes

- All of the problems with classes can be solved with an automatic import
- Use `classes.iastate.edu` to get all of the sections
- Allow users to edit any of the information pulled back



Cohorts

- Definition: A cohort is a composite of a major and a semester number
- Each cohort has a list of required classes
- The user can specify what classes are required for a given cohort
- Make sure that all class times are passed to the backend for a given class
- Allow users to easily edit which classes required for a cohort

The screenshot displays a web application interface for managing cohorts. At the top, there are navigation tabs: Scheduler, Classes, Cohort, Schedule, and Import. Below these is a table with columns for Program, Semester #, and Classes. A modal window titled "Add Cohort" is open, showing a form with the following fields:

- Program: CPRE (selected)
- Semester #: 2 (with "Ex: 3" below it)
- Class: (dropdown menu)
- ADD CLASS button

Below the form, a list of "Cohort Classes" is shown:

COM S 228	DELETE
CPR E 185	DELETE
CPR E 230	DELETE
CPR E 281	DELETE
MATH 165	DELETE

At the bottom of the modal are "CANCEL" and "SUBMIT" buttons. The background table shows a row for "CYB E" in semester "2" with classes "COM S 228, MATH 165".

Time Grid

- All of this produces the time grid
- This gives an estimate for percentage of students available for each section
- The colors green, yellow, and red give quick visual feedback for which times are best

	Monday	Wednesday	Friday
7:45 am	81.8	81.8	81.8
8:50 am	100	100	100
9:55 am	100	100	100
11:00 am	100	100	100
12:05 pm	47.7	47.7	33.41
1:10 pm	47.7	47.7	33.41
2:15 pm	72.7	72.7	58.41
3:20 pm	50	50	50
4:25 pm	0	0	0
5:30 pm	100	85.71	100

The Algorithm



Purpose:

Finds what availability a cohort has for every section in our time grid



Assumption:

Students in cohorts will be evenly distributed between the sections of their classes

The Algorithm

- Psuedo-Code:

For each class in cohort:

For each section of that class:

Add conflict with weight $1 / \#$ of sections in that class

- The class conflict information is sent to the frontend for display

Value Provided by Design

- Meant to record scheduled courses for the ECpE office and enter in courses that make up a student's core (flowchart)
- A good clean UI that any user can operate
- Menus that allow creation of classes and search for available times.
- A clean indication of what times classes are conflicting.
- Users can export the data of the schedule into a file for later usage
- It uses a familiar resource (classes.iastate.edu) to pull information from for easy class importation.

The screenshot shows a web application window titled "Class Scheduler" with a menu bar (File, Edit, View, Window, Help) and a navigation bar with buttons for "Scheduler", "Classes", "Cohort", "Schedule", and "Import". Below the navigation bar is a tab labeled "Tab 1" and a yellow bar with "EXPORT TO CSV" and "CLASS SETTINGS" options. The main content is a table with time slots on the y-axis and days of the week on the x-axis. The table data is as follows:

	Monday	Wednesday	Friday
7:45 am	100	100	100
8:50 am	100	100	100
9:55 am	100	100	100
11:00 am	100	100	66.67
12:05 pm	100	90	42.38
1:10 pm	100	90	75.71
2:15 pm	100	90	75.71
3:20 pm	100	90	90
4:25 pm	0	0	0
5:30 pm	100	75.71	100

DEMO

