

Team 42

Client: Vicky Thorland-Oster, Tina Prouty

Adviser: Thomas Daniels

Team Members/Roles

Ryan Tullis – Team Lead

Dillon Gesy – Administrative Lead

Nicolas Figueroa Calderas – Frontend Developer

Charles “Joe” Hosier – Frontend Developer

Abdullahi Abdullahi – Backend Developer

sdmay24-42@iastate.edu

[Team Website](#)

Introduction/Background

Problem statement

Initially, the problem given to us was to create an optimal schedule given restraints. However, this was quickly deemed very difficult due to the very human process our clients go through to schedule the classes. Therefore, our goal has shifted to creating a class-scheduling visualization tool. Our end goal is to create a system that our clients can use each year to help aid them in scheduling.

Intended users and uses

The project is specifically intended to be used by the schedulers within the Department of Electrical and Computer Engineering. In Fall 2023, Vicky Thorland-Oster was in charge of scheduling. Due to Tina Prouty taking over in 2024, she is now the main client. Currently, the process Vicky and Tina do to schedule the classes is very time consuming given all the different requirements and restraints on when classes can and cannot be. In the future with more work done, this project could easily adapt to help other departments as well.

Related products

Our project closely resembles the Iowa State University Schedule of Classes, found [here](#). Although this is a great scheduler to base our project off of, there were two key problems.

- 1) It is made for students
 - 2) It uses Access+ to populate its data
-
- 1) Matters because students have a very different schedule than professors. For example, students have to schedule based around their flowchart, while professors just need to have a room and time scheduled.
 - 2) Is important because Iowa State University is shifting to Workday. Currently, all systems using or relying on Access+ will cease to work in the near future.

Although the Schedule of Classes will soon cease to be functional, it encaptures the general idea and execution of our project.

Revised Design

Requirements

The following is a list of functional requirements we determined with our client at the start of the Spring 2024 semester.

- **Requirement #1:** View/edit past schedules
 - Let our client be able to view past year's schedules with our program to cross reference when making their schedules
 - Let our client be able to import a past year's schedule with our program in order to create a new schedule
- **Requirement #2:** Filter visible classes
 - Let our client be able to filter the visualizer by multiple things:
 - Classes
 - Departments
 - Labs, Recitations
 - Days
- **Requirement #3:** Exportation
 - Give our client a compiled list of data of what they scheduled using our program
 - For example, exporting to Excel each class and its data, such as when it was scheduled, section number, professor, etc.
- **Requirement #4:** Importation
 - Allow our client to import schedule data
 - For example, being able to import data from Fall 2024. We will not have this data as we will be graduated, so our user needs to have a way to import it.
 - Allow our client to import new class times data
 - For example, classes in the past semester have started at 8:00. Now, they can start at 7:45.
 - Allow our client to import degree flowcharts
 - Being able to import degree flowcharts will help with conflict checking in Requirement #5

- **Requirement #5: Warnings**
 - Conflict check, through the flowcharts, whether or not a class can be scheduled at a time.

Below describes the non-functional requirements of our project.

- **Requirement #1: Usability**
 - Allow our client to navigate our system easily
- **Requirement #2: Scalability**
 - As the data grows larger and larger over the years, the system should also scale

Engineering standards

More standards also apply to our project. However, we found these to be the most relative to our project and end goal.

ISO/IEC 25010:2011 - The software made is of quality and has specific characteristics that entails the quality.

ISO/IEC/IEEE 29148:2018 - The software's requirements will be progressively elaborated on through the project's lifecycle. We will get closer to what the client wants over time.

Security concerns and countermeasures

Our data is completely safe and publicly available. There are no security concerns regarding our project due to it being a local application. If our application were hosted online like Iowa State University's Schedule of Classes, we would need to look into encryption.

Description of evolution since 491

Our concept from 491 used an entirely different tech stack - besides the SQLite database. Initially, we decided on Java using the Eclipse IDE with Google Web Toolkit. We then changed it to C# using the MAUI framework. The ideas remained similar, but our functionality under a time constraint shifted us from conflict checking to schedule editing. Below are figures of our very concept to what it is now.

491 Design 0

M	T	W	TR	F	
4 classes	5 classes	4 classes	4 classes	4 classes	8am
• • •	• • •	• • •	• • •	• • •	
• • •	• • •	• • •	• • •	• • •	
					8pm

Search by: Dropdown
Instructor
Class

Conflicts:

SE 491 & SE 421
SE 317 & SE 309

•
•
•
•

492 Design



Implementation Details

Detailed design

Our project has multiple pages and ways to interact with the project. We have everything broken up into Data, Models, ViewModels, and Views. A majority of our project relies on the ViewModels and Views, since it is a visualizer.

Description of functionality

Below describes the main/important functionality of each component. It will entail why it exists, what it is supposed to do, and the current implementation at this point.

Data: Import.cs

The import class does exactly what it says - imports data into our system. Other similar senior design projects have taken in data from Iowa State University's Schedule of Classes, described earlier. We knew this would not be robust and would eventually break our project due to the implementation of Workday. Therefore, we decided to ask the Registrar's office and our client for previous semester class data. However, since Workday is being implemented, both our client and the Registrar's office were completely taken up by Workday problems, so we never got access to this data. Since we don't have a file to reference what an imported file might look like, our import architecture isn't fully flushed out for user-use. To get something in our system, we imported COM S classes from an online PDF file, found [here](#). Due to the nature of copying data from a PDF file, we had to manually format and standardize the text in a .txt file - something we wouldn't need to do in the future once Workday is implemented. Below is an example of the process from taking it from a .pdf file to a .txt file.

PDF file

INTRO DATA STRUCTUR COM S 228 3 CREDITS
 Prereq: MINIMUM OF C- IN COM S 227; CREDIT OR CONCURRENT ENROLLMENT IN MATH 165
 Note: 2-PART COURSE. REGISTER FOR LETTERED SECTION FIRST AND THEN NUMBERED SECTION. STUDENTS WHO DON'T REGISTER FOR BOTH PARTS OF THE COURSE SHOULD DROP THE COURSE DUE TO INCOMPLETE REGISTRATION.
 Night exams scheduled. For dates go to <http://www.registrar.iastate.edu/students/exams/night-exams>.

9238100 A MWF 8:50-9:40
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238105 B MWF 3:20-4:10
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238110 C MWF 11:00-11:50
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238115 D MWF 2:15-3:05
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238120 E MWF 2:15-3:05
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238015 1 T 12:40-1:30
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238020 2 R 12:40-1:30
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238025 3 R 8:00-8:50
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238030 4 T 8:00-8:50
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238035 5 T 9:30-10:20
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238040 6 T 4:10-5:00
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238045 7 R 9:30-10:20
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238050 8 R 4:10-5:00
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238055 9 T 2:10-3:00
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

9238060 10 R 2:10-3:00
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

DISCRETE CMP STRUCT COM S 230 3 CREDITS
 Prereq: MINIMUM OF C- IN (COM S 227; MATH 165); ENGL 150
 Note: 2-PART COURSE. REGISTER FOR A LECTURE SEC FIRST AND THEN A LAB SEC. STUDENTS WHO DON'T REGISTER FOR BOTH LECTURE AND LAB SHOULD DROP THE COURSE DUE TO INCOMPLETE REGISTRATION.
 Night exams scheduled. For dates go to <http://www.registrar.iastate.edu/students/exams/night-exams>.

4005005 A TR 11:00-12:15
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

4005010 B TR 12:40-1:55
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

4005015 C TR 3:40-4:55
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

4005100 1 W 12:05-12:55
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

4005105 2 W 4:25-5:15
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

4005110 3 W 3:20-4:10
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

INDEPENDENT STUDY COM S 290 VAR. CREDITS
 Prereq: PERMISSION OF INSTRUCTOR
 Note: NO MORE THAN 6 CREDITS OF COM S 290 OR COM S 290H MAY BE APPLIED TOWARD GRADUATION
 Instructor permission required 1st week.
 Satisfactory-Fail Only.
 ***** SEE DEPT FOR REFERENCE NUMBERS

INDEP STUDY HONORS COM S 290H VAR. CREDITS
 Prereq: INSTR PERMISSION, HONORS PROG MEMBERSHIP
 Note: COURSE RESTRICTED TO STUDENTS IN THE UNIVERSITY HONORS PROGRAM ONLY. NO MORE THAN 6 CREDITS OF COM S 290 OR COM S 290H MAY BE APPLIED TOWARD GRADUATION.
 Instructor permission required 1st week.
 ***** SEE DEPT FOR REFERENCE NUMBERS

SOFTWR DEV PRACTCES COM S 309 3 CREDITS
 Prereq: MINIMUM OF C- IN (COM S 228; MATH 165)

4438005 1 MWF 9:55-10:45
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

4438010 2 MWF 11:00-11:50
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

4438015 3 MWF 2:15-3:05
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

DSN&ANALYSIS ALGRTH COM S 311 3 CREDITS
 Prereq: MINIMUM OF C- IN (COM S 228; MATH 166; ENGL 150); (COM S 230 OR CPR E 310)
 Note: 2-PART COURSE. REGISTER FOR LETTERED SECTION FIRST AND THEN NUMBERED SECTION. STUDENTS WHO DON'T REGISTER FOR BOTH PARTS OF THE COURSE SHOULD DROP THE COURSE DUE TO INCOMPLETE REGISTRATION.
 Night exams scheduled. For dates go to <http://www.registrar.iastate.edu/students/exams/night-exams>.

2683005 A TR 8:00-9:15
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

2683010 B TR 11:00-12:15
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

2683015 C TR 12:40-1:55
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

2683100 1 W 7:45-8:35
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

2683105 2 W 9:55-10:45
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

2683110 3 W 2:15-3:05
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

CONSTR USER INTRFAC COM S 319 3 CREDITS
 8537005 1 MWF 8:50-9:40
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

INTR CPTR ARCH&MACH COM S 321 3 CREDITS
 Prereq: C- OR BETTER IN COM S 228, C- OR BETTER IN MATH 165, COM S 230 OR CPR E 281, ENGL 250

2684005 1 MWF 12:05-12:55
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

2684010 2 MWF 12:05-12:55
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

ADV PROGRAMM TCHNO COM S 327 3 CREDITS
 Prereq: MINIMUM OF C- IN (COM S 228; MATH 165)
 Night exams scheduled. For dates go to <http://www.registrar.iastate.edu/students/exams/night-exams>.

2454005 1 TR 9:30-10:45
 MEETS: AUG 21-DEC 15
 Note: Course delivery is On-site instruction

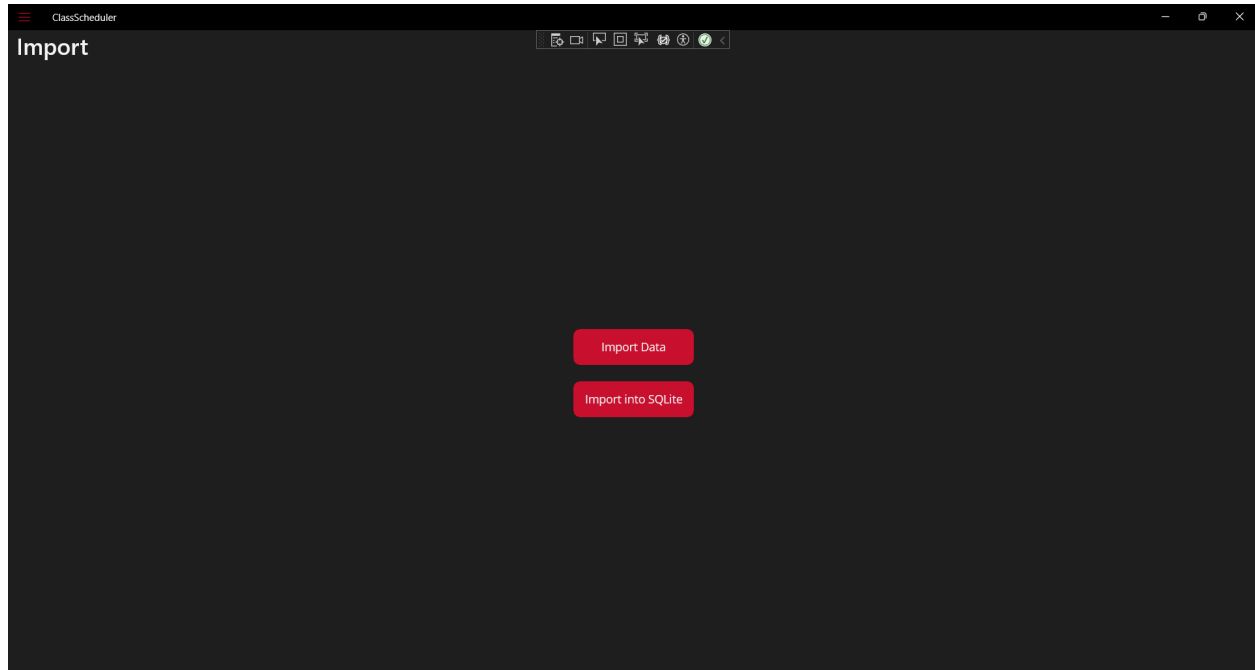
2454010 2 TR 9:30-10:45
 MEETS: AUG 21-DEC 15

txt file

DISCRETE CMP STRUCT COM S 230 3 CREDITS
4005005 A TR 11:00-12:15
MEETS: AUG 21-DEC 15
Note: Course delivery is On-site instruction
4005010 B TR 12:40-1:55
MEETS: AUG 21-DEC 15
Note: Course delivery is On-site instruction
4005015 C TR 3:40-4:55
MEETS: AUG 21-DEC 15
Note: Course delivery is On-site instruction
4005100 1 W 12:05-12:55
MEETS: AUG 21-DEC 15
Note: Course delivery is On-site instruction
4005105 2 W 4:25-5:15
MEETS: AUG 21-DEC 15
Note: Course delivery is On-site instruction
4005110 3 W 3:20-4:10
MEETS: AUG 21-DEC 15
Note: Course delivery is On-site instruction

LINUX SYS ESENTIALS COM S 252 3 CREDITS
3087005 1 ARRANGED
MEETS: AUG 21-DEC 15
Note: Course delivery is WWW
3087010 2 ARRANGED
MEETS: AUG 21-DEC 15
Note: Course delivery is WWW

Along with the Import functionality, comes the import page in Views, which looks like the following:



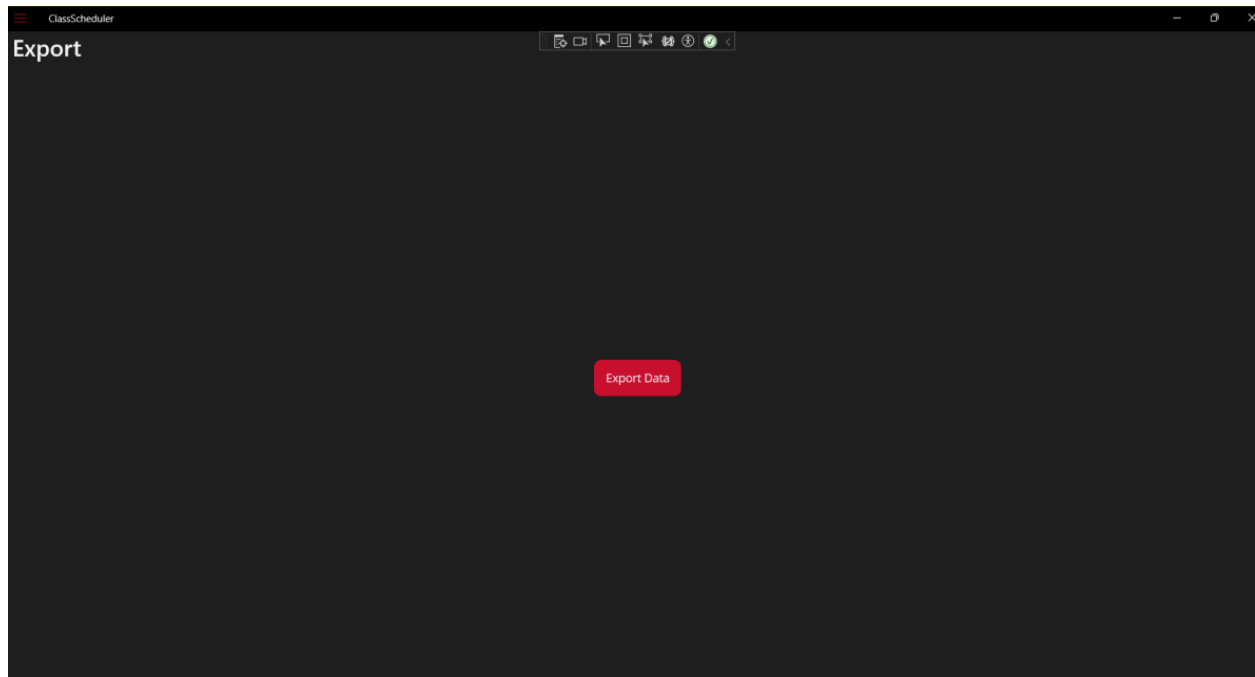
The import data function is supposed to take in a formatted file full of a semester's classes, given a file name. However, it currently only imports a specified .txt file. It is then supposed to display it in the Time Grid.

The Import into SQLite function imports the data directly into the local SQLite database. This is unimplemented at the moment.

Data: Import.cs

The export class also does exactly what you would expect it to do. It takes in information and exports it into a .txt file. This class takes in Course objects with Section objects and lists all of its relative information in a .txt file. Until we receive data from the Registrar's office, we have no idea about the formatting of information, so this is the next best thing.

Below is our export page:



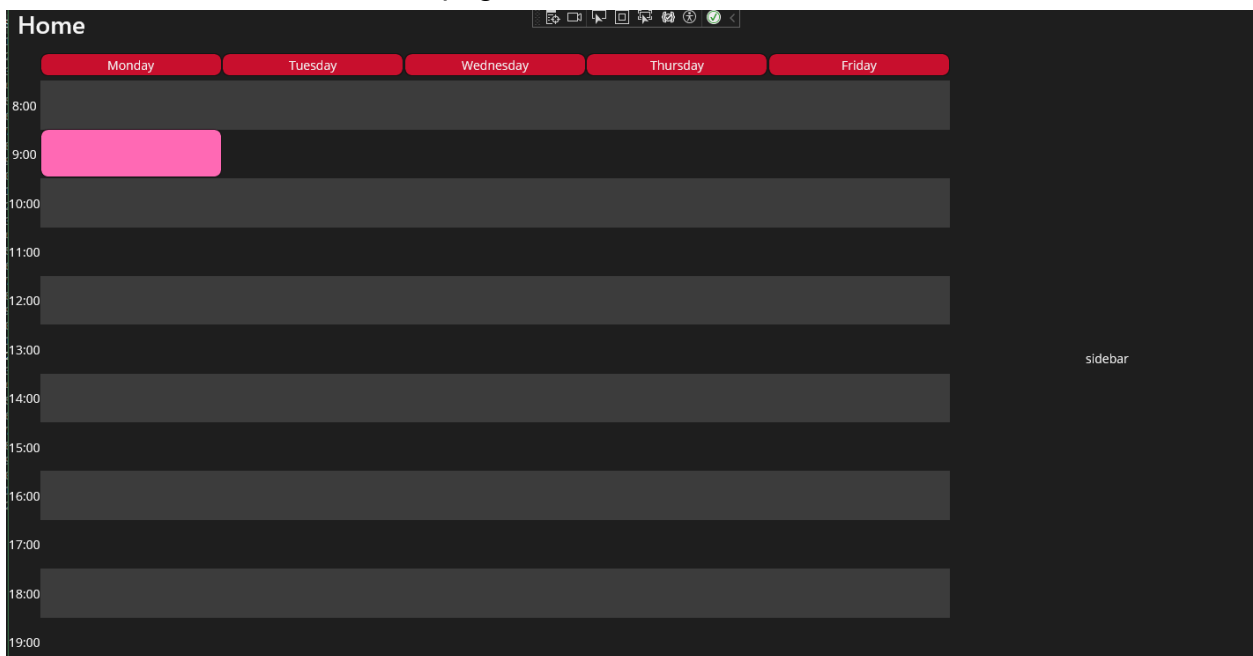
The following is our .txt file output after clicking the export data button:

```
1 Name: ORIENTATION Code: COM S 101 Credits: 0
2 Section Label: 1 Days Of Week: MW Start Time: 2:15 End Time: 3:05
3 Section Label: 2 Days Of Week: MW Start Time: 2:15 End Time: 3:05
4 Section Label: 3 Days Of Week: MW Start Time: 2:15 End Time: 3:05
5 Section Label: 4 Days Of Week: MW Start Time: 2:15 End Time: 3:05
6 Section Label: 5 Days Of Week: MW Start Time: 2:15 End Time: 3:05
7 Section Label: 6 Days Of Week: MW Start Time: 2:15 End Time: 3:05
8 Section Label: 7 Days Of Week: MW Start Time: 2:15 End Time: 3:05
9 Section Label: 8 Days Of Week: MW Start Time: 2:15 End Time: 3:05
10 Section Label: 9 Days Of Week: MW Start Time: 3:20 End Time: 4:10
11 Section Label: 10 Days Of Week: MW Start Time: 3:20 End Time: 4:10
12 Section Label: 11 Days Of Week: MW Start Time: 3:20 End Time: 4:10
13 Section Label: 12 Days Of Week: MW Start Time: 3:20 End Time: 4:10
14 Section Label: 13 Days Of Week: MW Start Time: 3:20 End Time: 4:10
15 Section Label: 14 Days Of Week: MW Start Time: 3:20 End Time: 4:10
16 Section Label: 15 Days Of Week: MW Start Time: 3:20 End Time: 4:10
17
18 Name: INTRO TO PROGRAMMNG Code: COM S 104 Credits: 2
19 Section Label: 1 Days Of Week: Arr. Start Time: End Time:
20 Section Label: 2 Days Of Week: Arr. Start Time: End Time:
21
22 Name: INTRO WEB PROGRAMMG Code: COM S 106 Credits: 3
23 Section Label: 1 Days Of Week: Arr. Start Time: End Time:
24
```

Views: Time Grid

This view is the main point of our project. This page's intent is to display courses on a time grid with the ability to filter which classes are shown. Clicking on a course displayed on the time grid brings a popup that provides information about the course. The grid interacts with the database and import function to display the courses.

Below is our current Time Grid page:

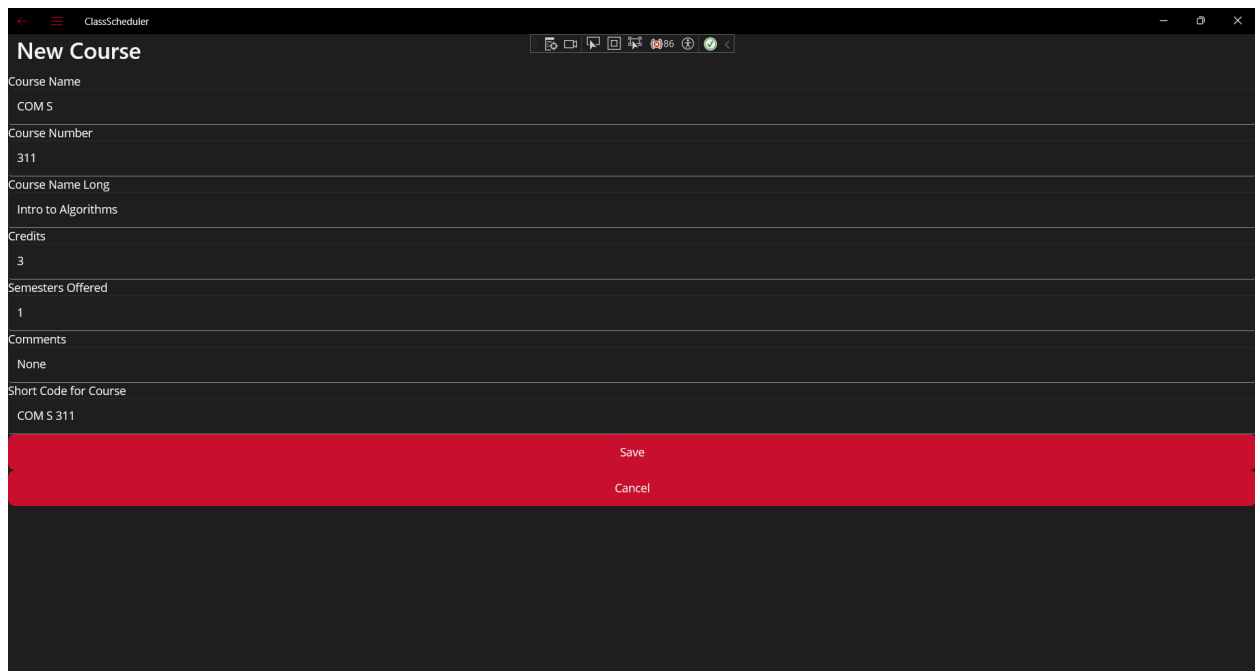
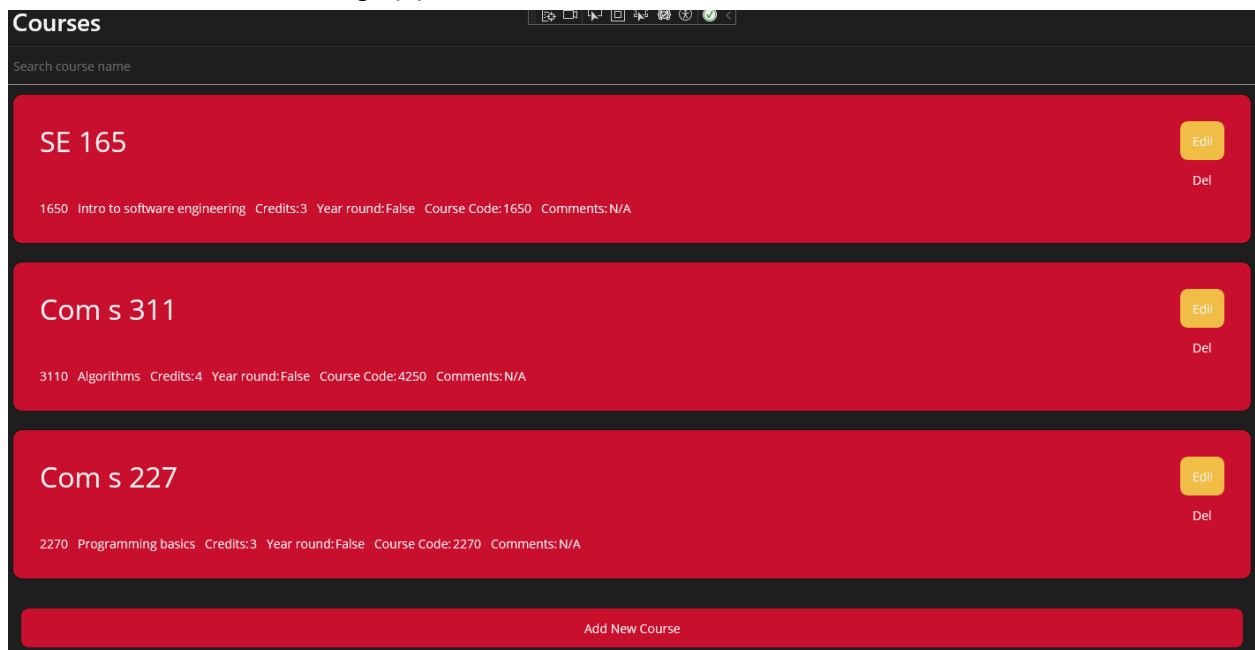


In reality, our time grid doesn't have much functionality. The filters are not yet implemented and our courses are currently not being displayed given information.

Views: Courses Page

This view stores all of our courses in list form. A course can be added, deleted, and edited on this page. The course delete button deletes the course from the database, as the add button adds it to the database. The edit button gives the user the ability to change any information about the course.

Below is our Courses Page(s):



Everything on the Courses Page is implemented.

Notes on implementation

As seen above, most of the implementations for our ideas on the project are incomplete. There are a few reasons we did not get as far as we wanted with the project, such as Workday being implemented, being unfamiliar with the tech stack, unclear work distribution, and other issues. Future work on this project should see this fully implemented in a semester or less.

Testing

Process

Due to an incomplete project, we were unable to test most of the functionality with the project. Most of the testing done was debugging to make sure an intended function works.

Results

Since a majority of the project is unimplemented, testing results only come from if the intended function works or not.

Broader Context

Generally, our outlook on these context areas haven't changed from our first design.

Public health, safety, and welfare:

Helps the Iowa State University community - specifically the schedulers. It gives the schedulers, our clients, more context when they schedule classes and

Global, cultural, and social:

We reflect all of Iowa State University's standard practices.

Environmental:

The only environmental concern is energy consumption, which a device requires to run.

Economic:

There is no economic impact.

Conclusions

Review progress

Overall, I think we as a team and individuals are disappointed with our total progress on the project. Although we had good ideas and had everything laid out, we had trouble executing our plans. A majority of our completed work is just the ground-work/basis of our total project.

Discuss value your design provides toward the problem

With a little work, information regarding past semesters can be shown to our clients to reference when scheduling classes.

Potential future steps

Future steps would be to allow them to filter and edit classes as they see fit. Populating the database with past semesters and giving the user full control on which data is used on the time grid would be super useful to reference.

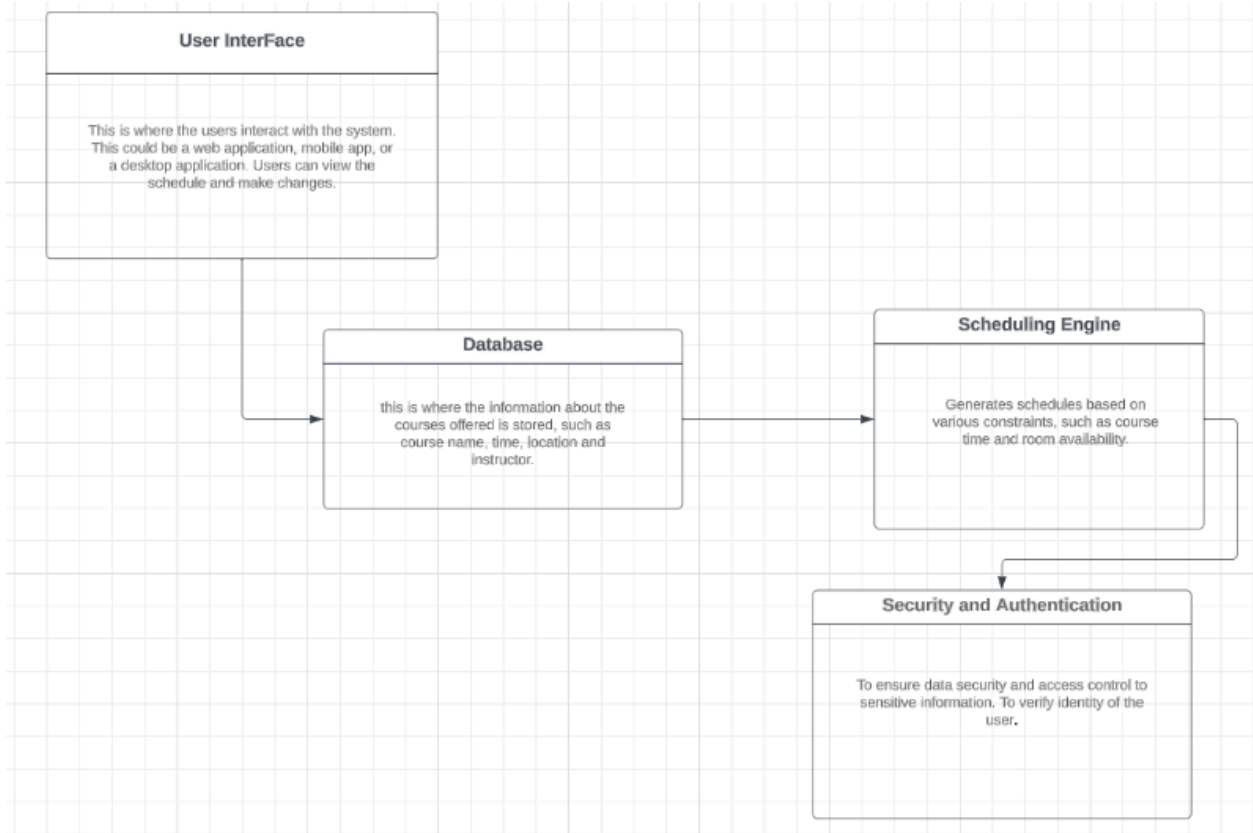
Conclusions

Appendix 1 - Operation Manual

To use our project, the user would upload the project to their computer using a flash drive. The flash drive would contain the project executable for the user to run. The main hub for all our separate pages is on the top right corner, with the 3 lines. Here, you can navigate to which page you want to do what you need. If just referencing past semesters, you should navigate to the import page and then to the time grid.

Appendix 2 - Alternative/initial version of design

An alternative version of this design is hosted on the web, like Iowa State University's Schedule of Classes. This takes a different approach and has different broader contexts, as it is hosted on a server. Here is an early design:



Another design had more focus on scheduling and their conflicts, focused on letting the user know where two classes couldn't be scheduled. An early drawn image is provided below:

Mon.	Tue.	Wed.	Thur.	Fri.	Time		
9 Classes	7 Classes	9 Classes	7 Classes	9 Classes	8:00	<div style="background-color: #f0f0f0; padding: 5px;"> <p>Filters</p> <p><input type="text"/> Professor <small>First Last</small></p> <p><input type="text"/> Class <small>e.g. SE 491</small></p> <p><input type="text"/> Section <small>e.g. 5 or B</small></p> <p><input type="text" value="2024"/> Year</p> <p><input type="radio"/> Fall <input type="radio"/> Spring</p> </div>	
10 Classes		10 Classes		10 Classes	10 Classes		9:00
13 Classes	9 Classes	13 Classes	9 Classes	13 Classes	10:00		
							11:00
6 Classes		6 Classes		6 Classes	12:00		<p>Conflicts</p> <p>SE 421 & SE 491 Starting time: MWF 12:00pm Conflict Weight: 8</p>
9 Classes		9 Classes		9 Classes	1:00		<p>CPR E 308 & SE 339 Starting time: MWF 1:05pm Conflict Weight: 4</p>
					2:00		
					3:00		
					4:00		
					5:00		
					6:00		
					7:00		
					8:00		

Appendix 3 - What we learned

Throughout the project, we learned a lot about what it means to develop a software project and some do's and don'ts. Although our project is very incomplete, we can still take away very valuable lessons learned from our time developing it.

- **Group communication and workflow is very important**
 - If one person starts to fall behind, sometimes the whole group can fall behind
 - Absences with no repercussions can lead to less attendees and overall work output
 - If someone is unresponsive, it can put a fast halt on any work
- **Using a system you are familiar with is a big deal**
 - Due to only a few somewhat familiar with MAUI, .NET, and C#, this led to people being very unsure on what to do and how to do it
 - If help is needed regarding a tech stack you're unfamiliar with, it relies on the response of the people who do know it, creating a potential reliability problem
 - Having a new system takes time to learn, so you need to set time aside to understand how it works
- **Creating a GUI is hard**
 - Our client initially told us in 2023 that creating a GUI was going to be very difficult
 - While we still think our choice to create a GUI was better than using a Greedy algorithm, the combination of creating a GUI and using a tech stack new to most created a recipe for disaster
- **Milestones are important**
 - Throughout most of the semester, work rarely got done unless it needed to be done. The same idea that if you give someone a deadline to code something, the code will not be done until near the deadline
 - In order to counteract this, we decided to schedule demos. However, by this point we had mostly already gotten super frustrated with the GUI and new tech stack

Appendix 4 - Code

Most of our code currently resides on our dev branch on our git repository, found here:

https://git.ece.iastate.edu/sd/sdmay24-42/-/tree/dev?ref_type=heads