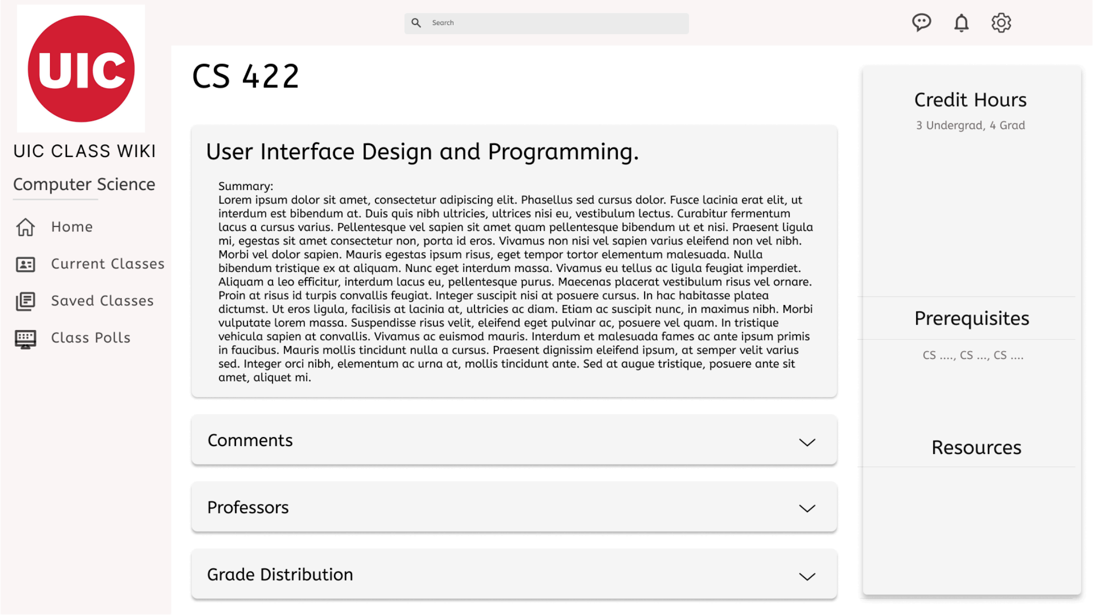
Oct 9, 2025
Tam , Rohan, Ayush, Trenton

Start with Figma Designs



CS 422

User Interface Design and Programming.

Summary:
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus sed cursus dolor. Fusce lacinia erat elit, ut interdum est bibendum at. Duis quis nibh ultricies, ultrices nisi eu, vestibulum lectus. Curabitur fermentum lacus a cursus varius. Pellentesque vel sapien sit amet quam pellentesque bibendum ut et nisi. Praesent ligula mi, egestas sit amet consectetur non, porta id eros. Vivamus non nisi vel sapien varius eleifend non vel nibh. Morbi vel dolor sapien. Mauris egestas ipsum risus, eget tempor tortor elementum malesuada. Nulla bibendum tristique ex at aliquam. Nunc eget interdum massa. Vivamus eu tellus ac ligula feugiat imperdiet. Aliquam a leo efficitur, interdum lacus eu, pellentesque purus. Maecenas placerat vestibulum risus vel ornare. Proin at risus id turpis convallis feugiat. Integer suscipit nisi at posuere cursus. In hac habitasse platea dictumst. Ut eros ligula, facilisis at lacinia at, ultricies ac diam. Etiam ac suscipit nunc, in maximus nibh. Morbi vulputate lorem massa. Suspendisse risus velit, eleifend eget pulvinar ac, posuere vel quam. In tristique vehicula sapien at convallis. Vivamus ac euismod mauris. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris mollis tincidunt nulla a cursus. Praesent dignissim eleifend ipsum, at semper velit varius sed. Integer orci nibh, elementum ac urna at, mollis tincidunt ante. Sed at augue tristique, posuere ante sit amet, aliquet mi.

Credit Hours
3 Undergrad, 4 Grad

Prerequisites
CS ...., CS ..., CS ....

Resources

Low to mid fidelity

- We need a straightforward and programmable design for the website frontend page
- Could be sectioned into three, sort of like Wikipedia or Fandom style sites
- Using CS Links and Resources Discord as a starting point for student-reviewed content on classes
- Feature: Students can mark them as “Taken”
- CS_Flowchart 2023 - Advanced idea (implementing an interactive CS flowchart, can mark classes for completion, “Taken”)
    - You should be able to hover over classes and transfer to a class page (as shown above)
-

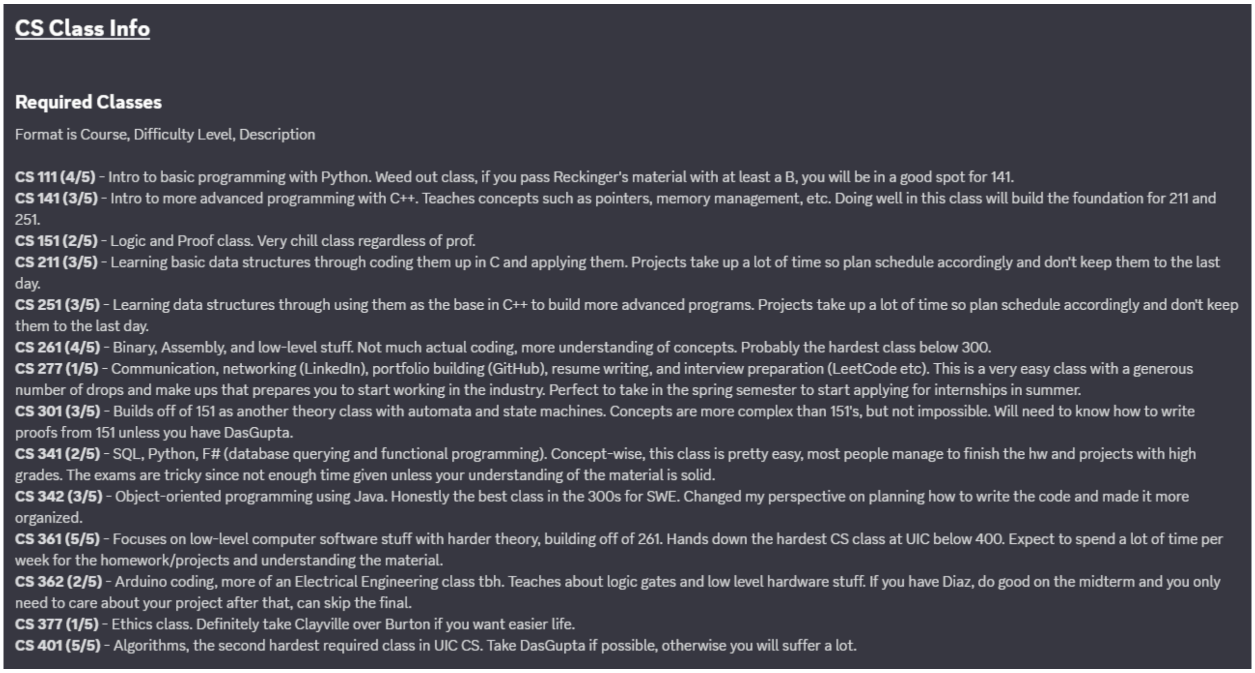
Key Designs (Making Low and Mid Fidelity Designs)

- Design Home page
- Design flowchart
    -
    -
- Design individual pages

- Figma Pair 1: Trenton & Rohan (working between Friday [10/10] and Sunday [10/11])
    - Making basic stuff to start Rohan off

        - Flowchart
- Figma Pair 2: Tam & Ayush (working between Sunday [10/12] and Tuesday [10/14])
    - Making basic stuff to start Ayush off

Programming Aspect

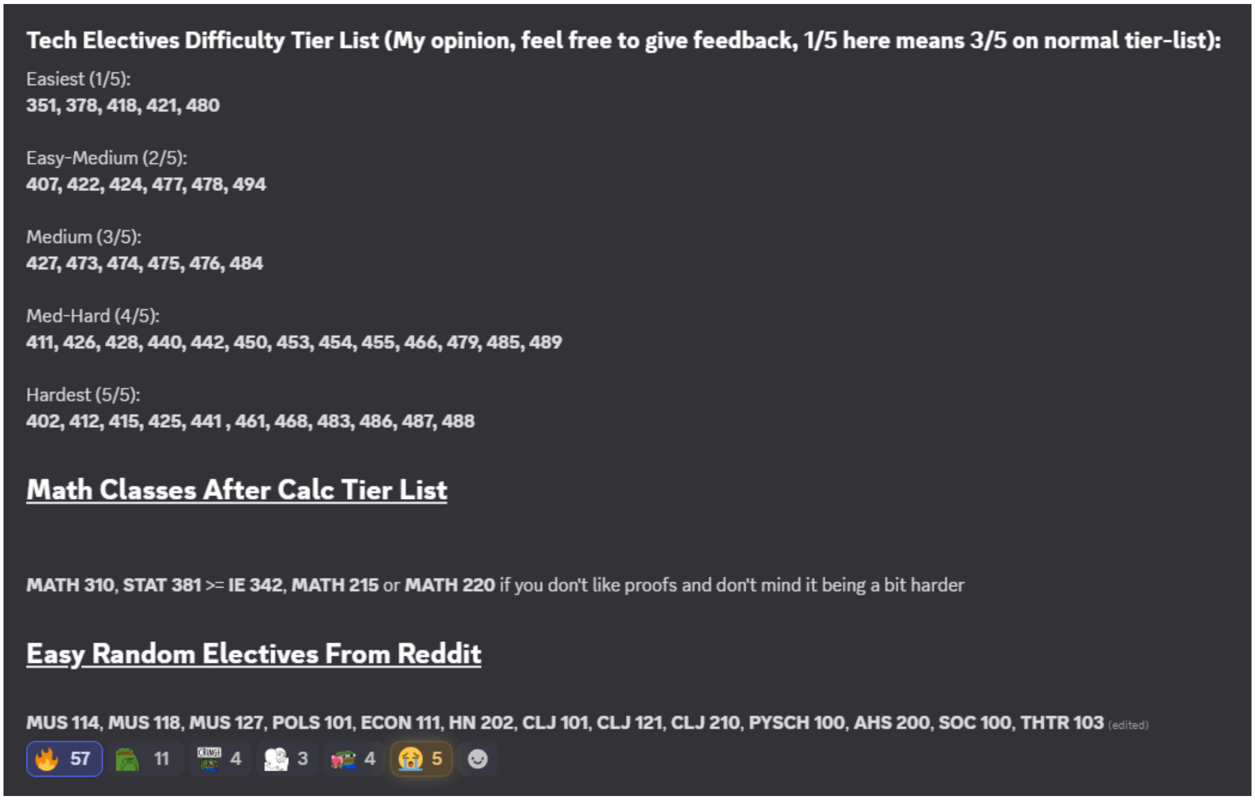
- The flowchart can possibly be implemented through a Trie
    - A child node, multiple children
    - How are copies handled (Have a customized Trie to store c
    - Home Page & Individual Pages
- 

CS Class Info

Required Classes

Format is Course, Difficulty Level, Description

CS 111 (4/5) - Intro to basic programming with Python. Weed out class, if you pass Reckinger's material with at least a B, you will be in a good spot for 141.
CS 141 (3/5) - Intro to more advanced programming with C++. Teaches concepts such as pointers, memory management, etc. Doing well in this class will build the foundation for 211 and 251.
CS 151 (2/5) - Logic and Proof class. Very chill class regardless of prof.
CS 211 (3/5) - Learning basic data structures through coding them up in C and applying them. Projects take up a lot of time so plan schedule accordingly and don't keep them to the last day.
CS 251 (3/5) - Learning data structures through using them as the base in C++ to build more advanced programs. Projects take up a lot of time so plan schedule accordingly and don't keep them to the last day.
CS 261 (4/5) - Binary, Assembly, and low-level stuff. Not much actual coding, more understanding of concepts. Probably the hardest class below 300.
CS 277 (1/5) - Communication, networking (LinkedIn), portfolio building (GitHub), resume writing, and interview preparation (LeetCode etc). This is a very easy class with a generous number of drops and make ups that prepares you to start working in the industry. Perfect to take in the spring semester to start applying for internships in summer.
CS 301 (3/5) - Builds off of 151 as another theory class with automata and state machines. Concepts are more complex than 151's, but not impossible. Will need to know how to write proofs from 151 unless you have DasGupta.
CS 341 (2/5) - SQL, Python, F# (database querying and functional programming). Concept-wise, this class is pretty easy, most people manage to finish the hw and projects with high grades. The exams are tricky since not enough time given unless your understanding of the material is solid.
CS 342 (3/5) - Object-oriented programming using Java. Honestly the best class in the 300s for SWE. Changed my perspective on planning how to write the code and made it more organized.
CS 361 (5/5) - Focuses on low-level computer software stuff with harder theory, building off of 261. Hands down the hardest CS class at UIC below 400. Expect to spend a lot of time per week for the homework/projects and understanding the material.
CS 362 (2/5) - Arduino coding, more of an Electrical Engineering class tbh. Teaches about logic gates and low level hardware stuff. If you have Diaz, do good on the midterm and you only need to care about your project after that, can skip the final.
CS 377 (1/5) - Ethics class. Definitely take Clayville over Burton if you want easier life.
CS 401 (5/5) - Algorithms, the second hardest required class in UIC CS. Take DasGupta if possible, otherwise you will suffer a lot.

- 

Tech Electives Difficulty Tier List (My opinion, feel free to give feedback, 1/5 here means 3/5 on normal tier-list):

Easiest (1/5):
351, 378, 418, 421, 480

Easy-Medium (2/5):
407, 422, 424, 477, 478, 494

Medium (3/5):
427, 473, 474, 475, 476, 484

Med-Hard (4/5):
411, 426, 428, 440, 442, 450, 453, 454, 455, 466, 479, 485, 489

Hardest (5/5):
402, 412, 415, 425, 441 , 461, 468, 483, 486, 487, 488

Math Classes After Calc Tier List

MATH 310, **STAT 381** >= **IE 342**, **MATH 215** or **MATH 220** if you don't like proofs and don't mind it being a bit harder

Easy Random Electives From Reddit

MUS 114, MUS 118, MUS 127, POLS 101, ECON 111, HN 202, CLJ 101, CLJ 121, CLJ 210, PYSCH 100, AHS 200, SOC 100, THTR 103 (edited)

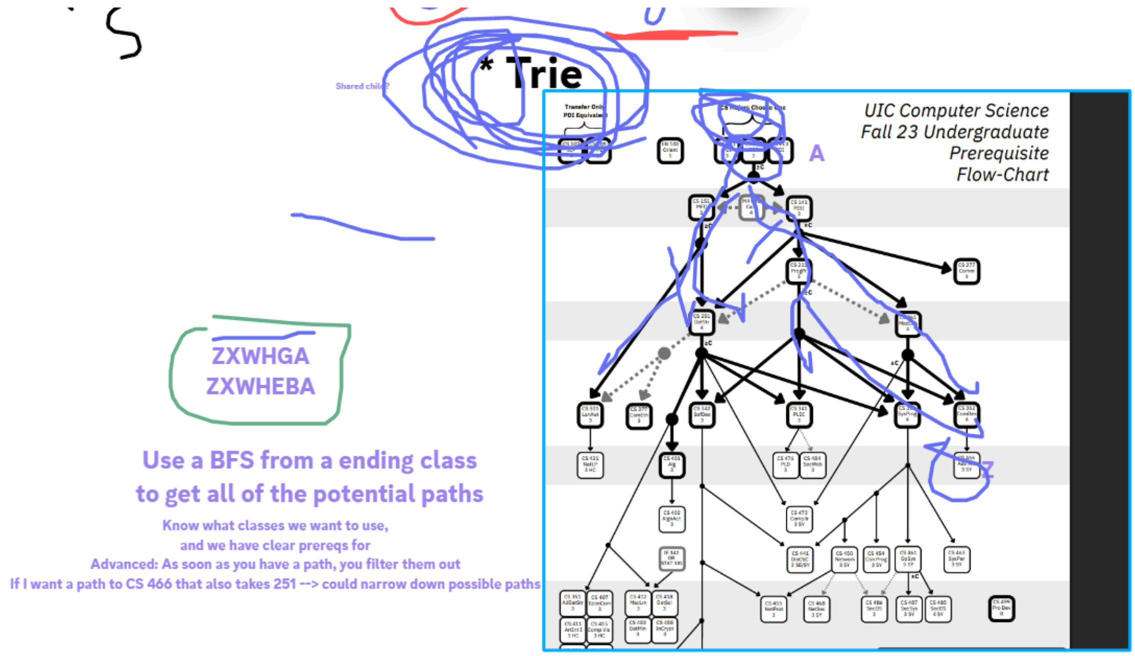
🔥 57 11 4 3 4 😭 5

- Aim to get this done before we meet again next Thursday

Backend

We will use a Trie to store UIC classes and use BFS to get all the prerequisites for a class

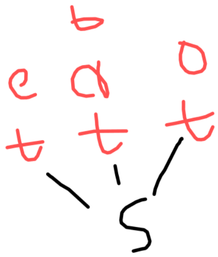
If Trie does not work, we switch to a priority queue

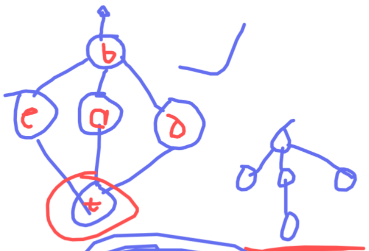


ZXWHGA
ZXWHEBA

Use a BFS from a ending class
to get all of the potential paths
Know what classes we want to use,
and we have clear prereqs for
Advanced: As soon as you have a path, you filter them out
If I want a path to CS 466 that also takes 251 --> could narrow down possible paths

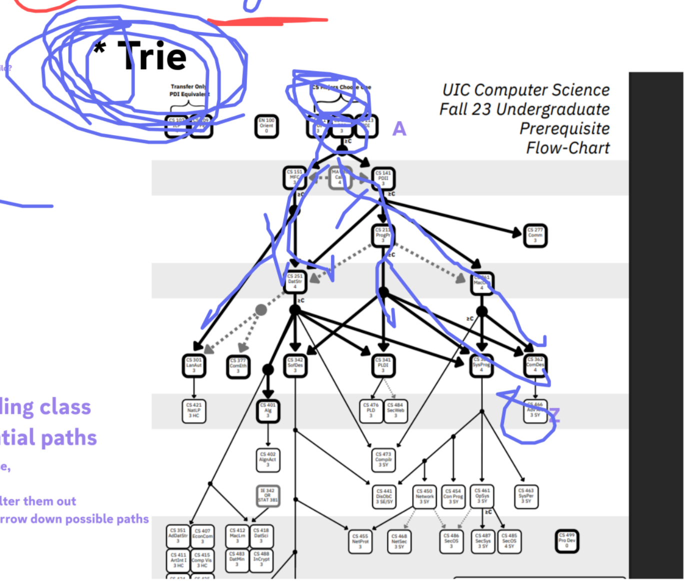
-
- Use a BFS from a ending class
- to get all of the potential paths
- Know what classes we want to use,
- and we have clear prereqs for
- Advanced: As soon as you have a path, you filter them out
- If I want a path to CS 466 that also takes 251 --> could narrow down possible paths
- Classes have higher priority than others (CS 141 may have more than one kind of priority)





* Trie

Shared child?


UIC Computer Science
Fall 23 Undergraduate
Prerequisite
Flow-Chart

ZXWHGA
ZXWHEBA

Use a BFS from a ending class
to get all of the potential paths

Know what classes we want to use,
and we have clear prereqs for
Advanced: As soon as you have a path, you filter them out
If I want a path to CS 466 that also takes 251 --> could narrow down possible paths