

MadCourseEvaluator

Requirements and Specification Document

2022-12-12, version 4.0

Project Abstract

Our project will be a web application that allows a user to search and view unique pages for each course at UW-Madison. Each page will contain aggregated information from DARS course information, Mad Grades, RateMyProfessor, and Reddit. This information will consist of course requirements, grade distributions for courses, professor ratings (including a base rating out of 5 and instructor specific grade distribution for the course), and general mentions that are found on the r/UWMadison subreddit. The user can also search up professors and look up the list of courses the professor teaches/has taught.

Document Revision History

Rev 4.0 2022-12-13: Third Revision

Rev. 3.0 2022-11-17: Second Revision

Rev. 2.0 2022-10-27: First Revision

Rev. 1.0 2022-10-09: Initial version

Customer

The customer for this software is any UW-Madison student who is trying to plan what courses they would like to take. In a typical semester, students usually do their own research into different courses and the professors teach them before their enrollment date. As students, we have all had our own experiences doing this, and we see a lot of utility in building a system that will serve as a one-stop shop for all the information we would typically take into account when making our course schedule selections for any given semester.

Competitive Landscape

Since this application looks to aggregate data from existing websites, the largest competition is in swaying a user to use our services over referring to these sources directly. Our service holds the obvious competitive advantage that a user only needs to come to our website, complete one simple search query, and be directed to a relevant dashboard instead of finding information from three different websites.

Additionally, contingent on the implementation of our schedule recommendations interface, we believe this to be a novel service as it doesn't appear to be offered to UW-Madison students anywhere else.

User Requirements

- A user will be able to **look up course information** for every course at the university by name.
- A user will be able to **see Reddit comments for their course of interest**.
- A user will be able to **see professor ratings for professors who have taught courses in a given semester**.
- A user will be able to **review the grade distribution of the courses**.

- A user will be able to **see the list of courses the professor has taught/is teaching and their review**

Use Cases

We need to create use cases that follow the format in the example files. They will need to cover all of the general actions a user will make in the application. After we compile our list of use cases each team member should select two to put into tables.

Use cases:

1. Search for a course page.
2. Search for an instructor page.
3. Look through instructor information on the course page.
4. Look through Reddit comments on the course page.

name	Searching for an instructor
actor	Any user of the application
trigger	When the user begins to type in the search form on the home page.
event	<ul style="list-style-type: none"> - The user begins to type in the search form, where there will be auto-complete suggestions of instructors whose name contains the substring provided in the search query. - The user either finishes typing in the full instructor's name and clicks search, or they select the full instructor's name from the auto-complete suggestions. - Once an instructor has been selected, the user will be redirected to that instructor's Instructor Page, where they will see a list of courses that the instructor has taught

	<p>before.</p> <ul style="list-style-type: none"> - The user will click on the course for which they want more information. - The user will be redirected to that course's Course Page.
exit	The user will be on the course page of the course that they selected from the instructor's taught courses.
post	n/a
acceptance	n/a

name	Viewing Course Details
actor	Any user of the application
trigger	<ul style="list-style-type: none"> ● When the user begins to type in the search form on the home page. ● When the user clicks on a specific course hyperlink on the professor page
event	<ul style="list-style-type: none"> - The user begins to type in the search form, where there will be auto-complete suggestions of courses whose name contains the substring provided in the search query <ul style="list-style-type: none"> - The user either finishes typing in the full instructor's name and clicks search, or they select the full instructor's name from the auto-complete suggestions. - Once a course has been selected, the user will be redirected to that Course Page, where they will see all the information related to that course. - The user clicks on the course hyperlink listed under the professor <ul style="list-style-type: none"> - Once a course has been selected, the user will be redirected to that Course Page, where they will see all the information related to that course.

exit	The user will be on the course page of the course with all its details.
post	n/a
acceptance	n/a

name	Look through instructor information
actor	Any user of the application
trigger	When the user scrolls in the instructor component
event	<ul style="list-style-type: none"> 1) Information about further down instructors shows 2) The user can click an instructor to link to their instructor page
exit	Only one section of the page, n/a
post	n/a
acceptance	n/a

name	Look through the reddit comments
actor	Any user of the application
trigger	<ul style="list-style-type: none"> • Users enter the course page and check out the comments • Expand to see more comments referenced by reddit
event	<ul style="list-style-type: none"> 1) The users enter the course page and see the comments section on the bottom left from reddit. <ul style="list-style-type: none"> a. The comments are limited by the space and only a few recent comments are shown b. Scroll down for more comments 2) The users try to see more comments and

	scrolls down a. The comment section will scroll down to less relevant comments
exit	only in one section of page, n/a
post	n/a
acceptance	n/a

User Interface Requirements



Figure 1: Sample design for ~/course/CS577 endpoint.

Madger Courses 🔍

CS 577

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occ...

+

+

-

+

+

-

A
B
C
D
F

reddit

+

+

-

+

+

-

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occ...

4/10 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occ...

4/10 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occ...

Figure 2: Sample design for ~/course/CS577 endpoint using Figma.

The screenshot shows a course page for 'Introduction to Algorithms' (COMP SCI 577) on a platform called Madger Courses. The page includes a navigation bar with a search icon, a sidebar with 'Reddit Insights' from reddit, and a main content area with sections for SUBJECT, CREDITS, DESCRIPTION, REQUISITES, INSTRUCTORS, and a bar chart showing student grades.

SUBJECT: Computer Sciences
CREDITS: 4
DESCRIPTION: Basic paradigms for the design and analysis of efficient algorithms: greed, divide-and-conquer, dynamic programming, reductions, and the use of randomness. Computational intractability including typical NP-complete problems and ways to deal with them.
REQUISITES: (MATH/COMPSCI240 or STAT/COMPSCI/MATH475) and (COMP SCI 367 or 400), or graduate/professional standing, or declared in the Capstone Certificate in Computer Sciences for Professionals

Instructors:

- Professor A 3.5/10**
Department: Computer Science
Rating: Poor
Average Class GPA: 2.33
- Professor B 5/10**
Department: Computer Science
Rating: Good
Average Class GPA: 3.05
- Professor C 4.5/10**
Department: Computer Science
Rating: Good
Average Class GPA: 3.05
- Professor D 2/10**
Department: Computer Science
Rating: Poor
Average Class GPA: N/A

Reddit Insights: reddit

Student Grade Distribution:

Grade	Count
A	2
AB	1
B	3
BC	1
C	1
D	1
F	1

Figure 3: Updated sprint 2 design for ~/course/CS577 endpoint using Figma.

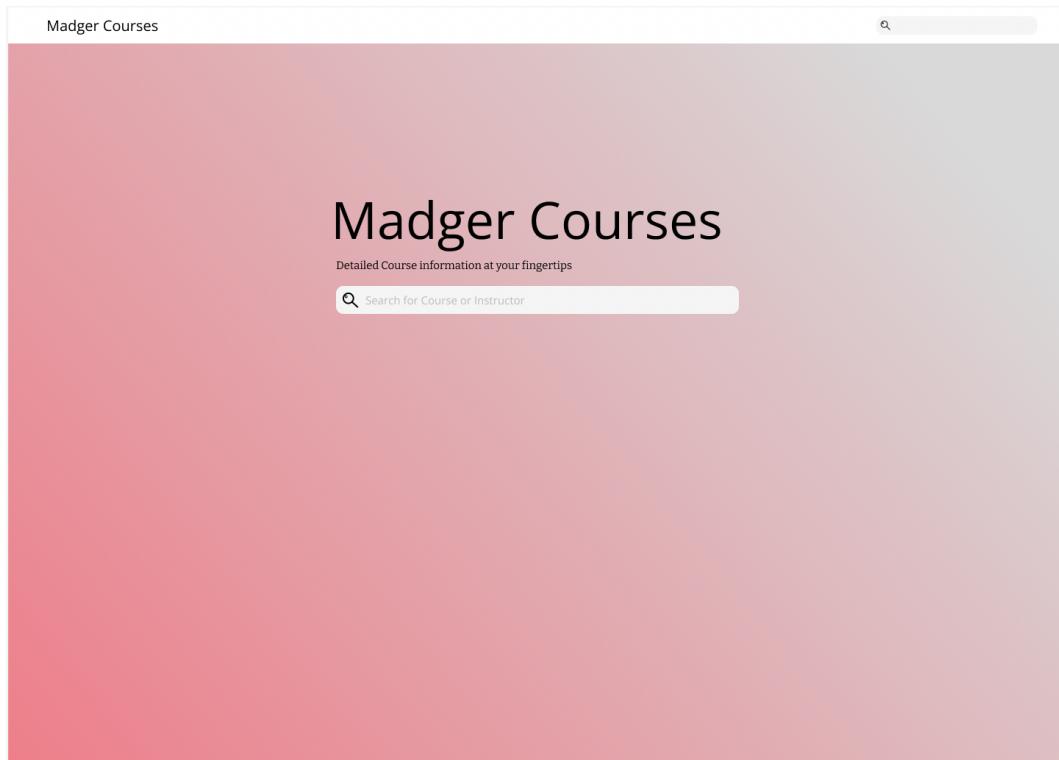


Figure 4: Course Homepage updated sprint 2 design using Figma



Professor A

Department
Computer Science
Rating
3.5/10 - Poor

Courses Taught

- COMP SCI 577 - Introduction to Algorithms
- COMP SCI 540 - Introduction to Artificial Intelligence
- MATH 222 - Calculus II
- MATH 341 - Linear Algebra

Figure 5: Professor Page sprint 2 design using Figma

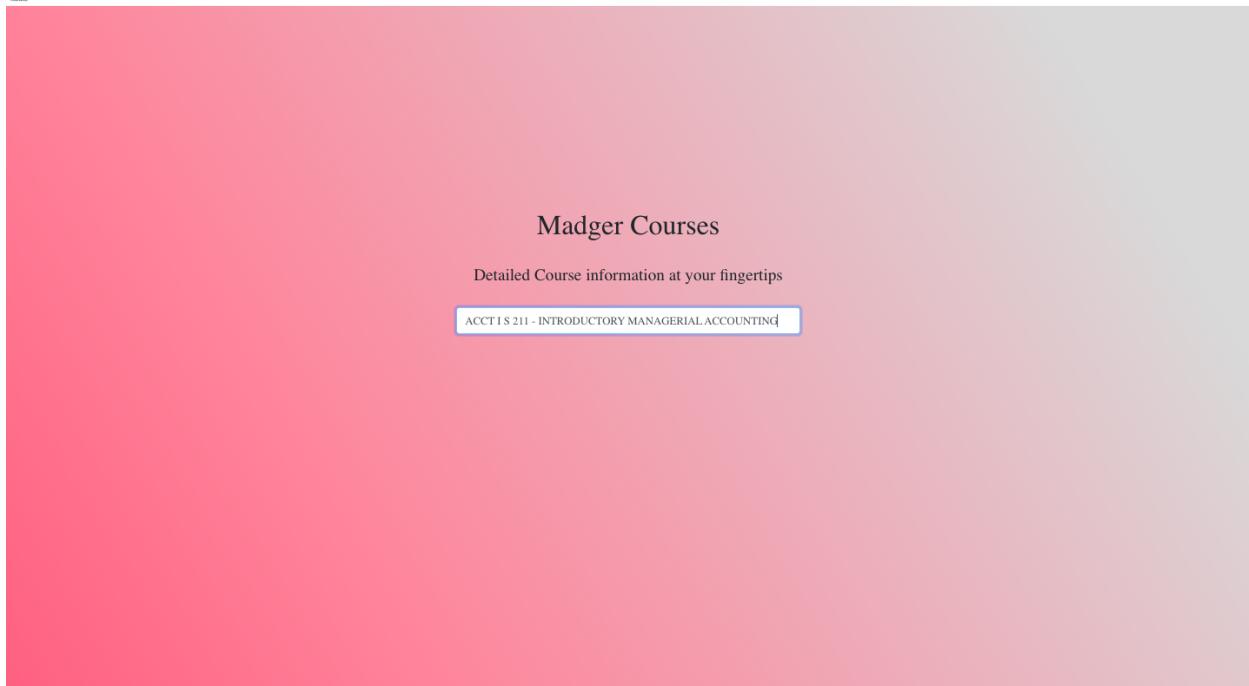


Figure 6: Hosted website's Home page on big screen sprint 3

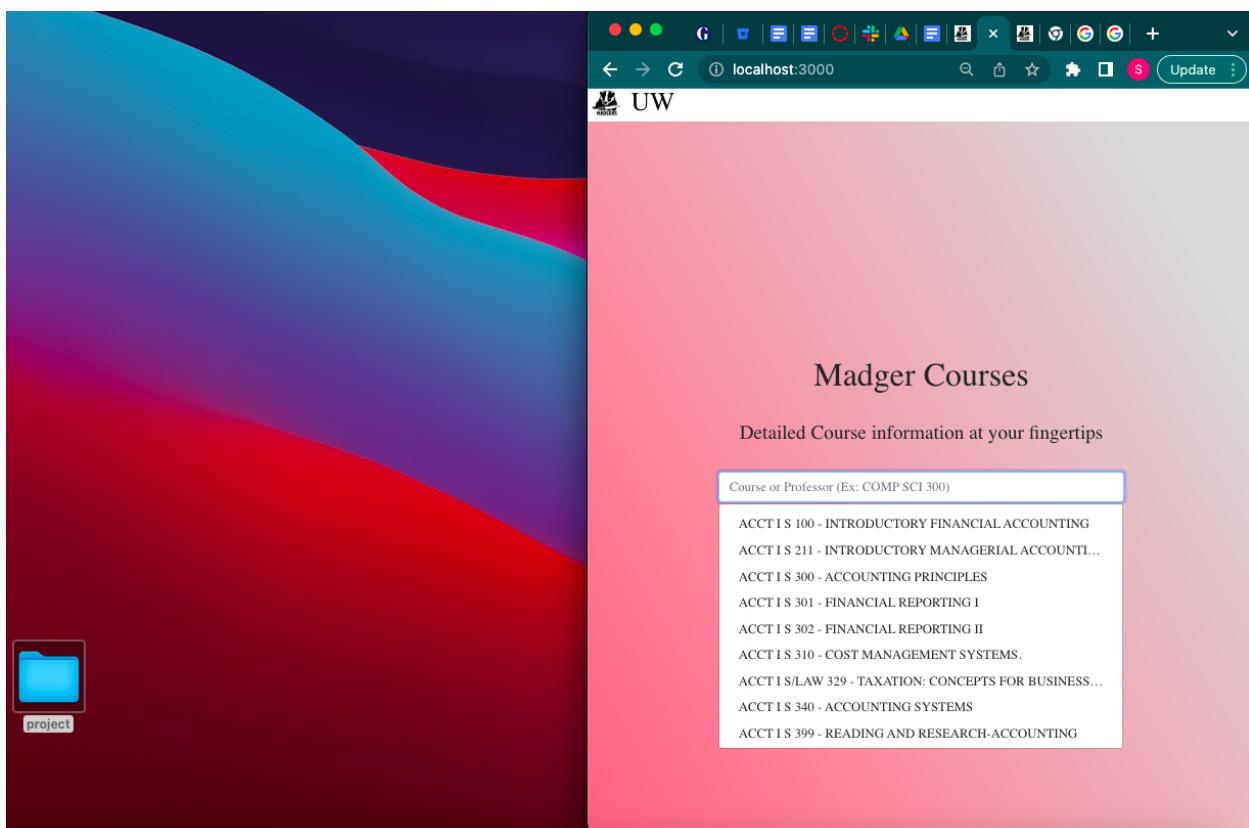


Figure 7: Hosted website's Home page on small screen sprint 3 [responsive]



INTRODUCTION TO ALGORITHMS

COMP SCI 577

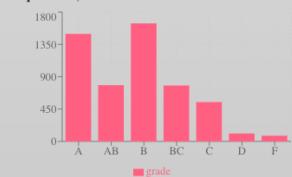
Subject

Computer Sciences

Description

Basic paradigms for the design and analysis of efficient algorithms: greed, divide-and-conquer, dynamic programming, reductions, and the use of randomness. Computational intractability including typical NP-complete problems and ways to deal with them.

Requisites: (MATH/COMPSCI240 or STAT/COMPSCI/MATH475) and (COMP SCI 367 or 400), graduate/professional standing, or declared in the Capstone Certificate in Computer Sciences for Professionals



Credits

4 credits

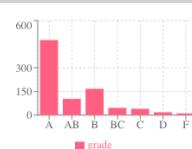
Instructors

Deborah Joseph

Dept
Computer Science

Rating

2.8/5, average

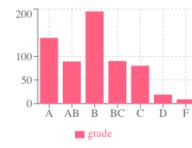


Eric Bach

Dept
Computer Science

Rating

1.9/5, poor

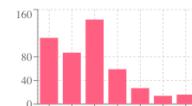


Marc Renault

Dept
Computer Science

Rating

4.2/5, good



This seems like a normal schedule, I doubt taking a lighter one would help with either cs course

1. I don't think 407 is offered next semester, I could be wrong, and there's no info about when it will be offered again. 2. 579 is pretty doable, you just need some progress every week and as long as you define a not too chill either. Some people can do it but if you do this combo, then all of your other courses should be chill enough and you shouldn't be taking any other CS courses other than these.

Not particularly, I wouldn't say it's an easy class either. I found it to be an average difficulty CS class, just stay on top of the work and you'll be fine.

Figure 8: Hosted website's Course page on big screen sprint 3

INTRODUCTION TO ALGORITHMS
COMP SCI 577

Subject
Computer Sciences

Credits
4 credits

Description
Basic paradigms for the design and analysis of efficient algorithms: greed, divide-and-conquer, dynamic programming, reductions, and the use of randomness. Computational intractability including typical NP-complete problems and ways to deal with them.

Requisites: (MATH/COMPSCI240 or STAT/COMPSCI/MATH475) and (COMP SCI 367 or 400), graduate/professional standing, or declared in the Capstone Certificate in Computer Sciences for Professionals

Instructors

Deborah Joseph

Dept
Computer Science

Rating
2.8/5, average

Eric Bach

Dept
Computer Science

Rating
1.9/5, poor

Marc Renault

Dept
Computer Science

Rating
4.2/5, good

Reddit Post

This seems like a normal schedule, I doubt taking a lighter one would help with either cs course

1. I don't think 407 is offered next semester, I could be wrong, and there's no info about when it will be offered again. 2. 579 is pretty doable, you just need some progress every week and as long as you define a not too

Figure 9: Hosted website's scrollable Course page on small screen sprint 3

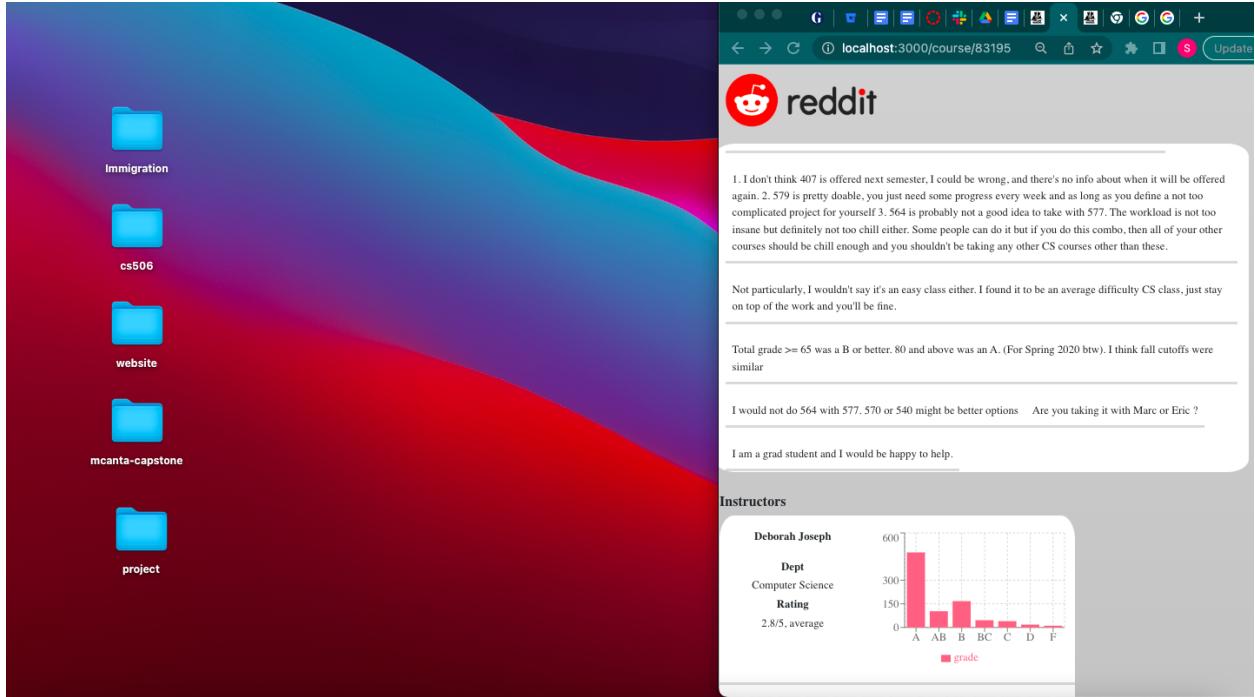


Figure 10: Hosted website's scrollable Course page on small screen sprint 3

Previously Taught Courses (10 courses found):

- COMP SCI/MATH 240 : INTRODUCTION TO DISCRETE MATHEMATICS
- COMP SCI 298 : DIRECTED STUDY IN COMPUTER SCIENCE
- COMP SCI 520 : INTRODUCTION TO THEORY OF COMPUTING
- COMP SCI 577 : INTRODUCTION TO ALGORITHMS
- COMP SCI 638 : UNDERGRADUATE TOPICS IN COMPUTING
- COMP SCI 681 : SENIOR HONORS THESIS
- COMP SCI 698 : DIRECTED STUDY
- COMP SCI 799 : MASTER'S RESEARCH
- COMP SCI 899 : PRE-DISSENTATOR RESEARCH
- COMP SCI 999 : DISSERTATOR RESEARCH

Figure 11: Hosted website's Professor page on big screen sprint 3

- Home page
 - Search for prof by form
 - Search for course by form
- Professor page
 - Display and link to set of courses that professor teaches
- Course page
 - Course name, code, description, requirements
 - Overview of grade distribution of the course (retrieved from madgrades.com)
 - List of instructors and their RMP info and professor-specific grade distribution
 - List of reddit comments with permalinks

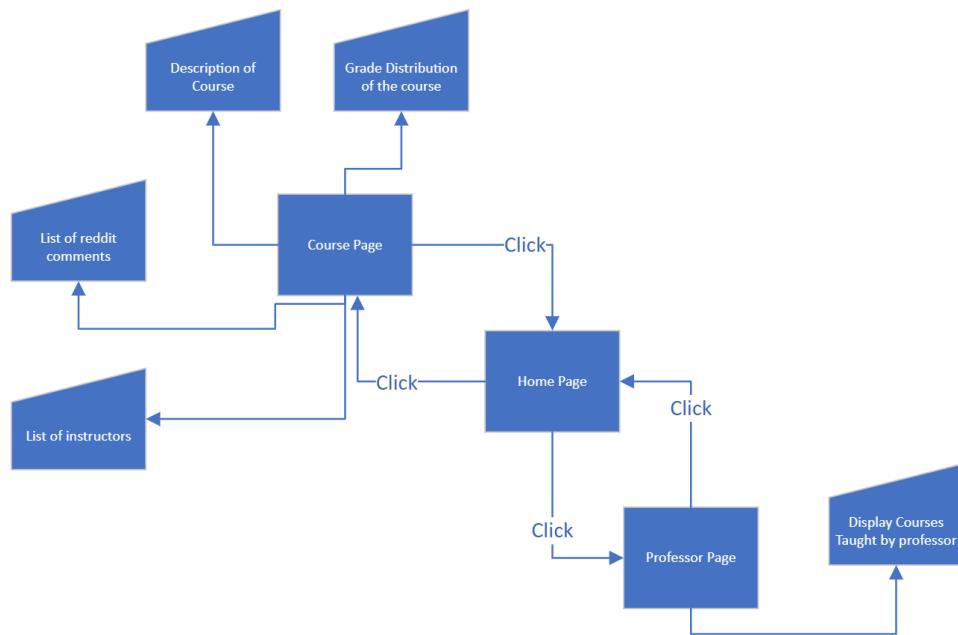
Security Requirements

At the moment, we have no plans to add user accounts. Still, if we did, we would have to encrypt user data stored in our database and take preventative measures to dissuade SQL injections or other attempts to access user information.

System Requirements

1. React (Front-End)
2. Python (Flask) (Back-End)
3. Database (MySQL)
 - a. Relational Database
 - i. Metadata
 1. Information for each course.
 2. Information for each professor.
 3. Reddit comments for each course
 4. Rate my professor ratings for each professor.
 5. MadGrades grade distributions for each course in each semester.

Specification



Product Backlog

Product Backlog				
User Story	Notes	Priority	Sprint	Estimated Story Points
Configure Web APIs for Reddit (PRAW), Mad Grades (custom API), and Rate My Professor (either use open source, or develop from scratch).	We have been able to get data from all WebAPIs, but we are still in the process of populating the database fully.	1	1	6
Component Testing		3	1,2	2

Mock Data Source	The backend team created .json files with the data points they will be passing to the frontend in the final product	1	1,2	2
Figma layout and Color/Theme	We used Figma to plan page layout and test various themes. Multiple iterations were made over sprints 1 and 2—see above user interface requirements	3	1,2	1
Frontend environment		1	1,2,3	8
Relational Database		1	1	2
Flask Starter + endpoints		1	1,2	3
Populate Database	ran into issues getting RMP to populate consistently	1	2	7
Finish frontend components	Components were mostly completed, fine tuning to come in sprint 3	1	2	3
Search Bar	course search complete, instructor search to come in sprint 3	1	2	6
Design updates	Many changes made, may fine tune more in sprint 3	3	2	1
Hosting and integrating the application with Heroku	We tried hosting the backend on heroku but because it's free tier could only handle limited requests, we switched to AWS for backend API and database We hosted the frontend on netlify since that was simpler and easier	1	3	8
Finish backend unit tests		1	3	5

Instructor page	Instructor page is made more responsive and prettier	1	3	2
Instructor List	Instructor list on the course page has graph integrated into it now	1	3	3
Pretty the frontend	The entire application is responsive.	2	3	4
User Story	Notes	Priority	Sprint	Estimated Story Points