CS 316 Milestone 2 Progress Report

Matthew O'Boyle, Christina Li, Feroze Mohideen,
Suchir Bhatt, and Salil Mitra

November 2018

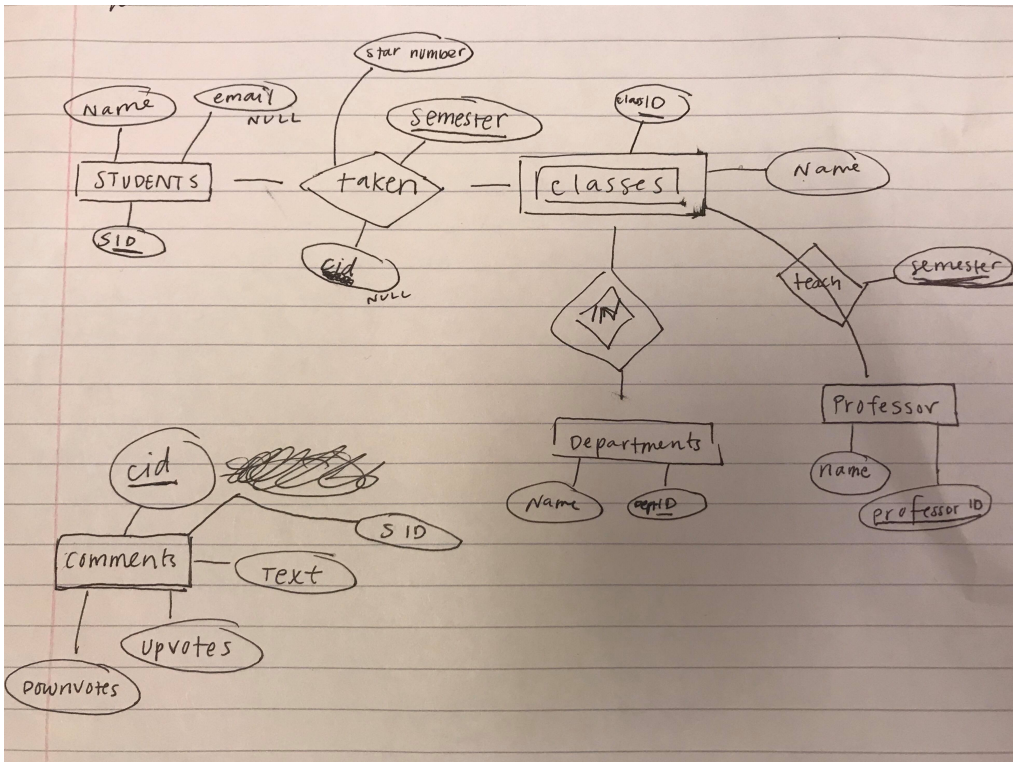
1 New Assumptions / Reworking of Idea

1.1 Problem Statement

Following our somewhat negative feedback from Milestone 1, we decided to pivot our idea to something more manageable and something we believe would be as useful. Previously, we wished to build a tool that advised students on what jobs/internships they could expect to receive given details about their schedule; however, we realized that this approach might have been somewhat short-sighted, since classes that are taken aren't necessarily the only contributing factor to the job search. Instead of reworking our design to incorporate external factors like extracurriculars, we decided to move on to a smaller problem to tackle - class scheduling.

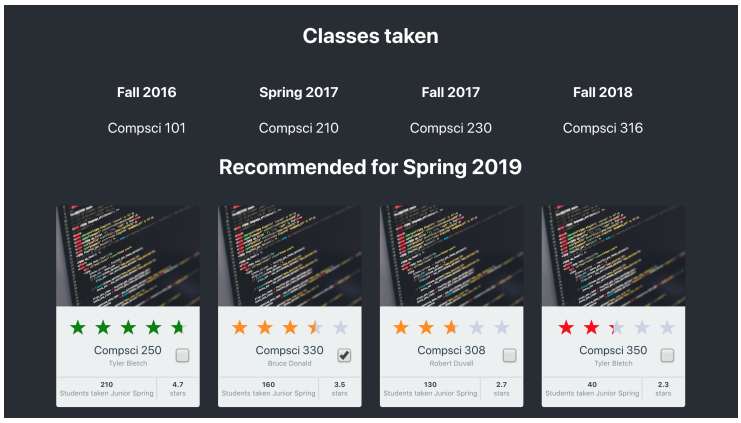
Book-bagging and scheduling during the last few weeks of each semester can be a very difficult task, especially for underclassmen who are unsure of what directions they may want to jump from their introductory level classes. We wish to address this problem by providing students with a recommendation service for class scheduling using aggregated previous schedules and course reviews from other students, who must opt-in with their data if they want to use the service. We would employ a database holding information on Students, Classes, Professors, Departments, and User Reviews, the details of which are found below. We hope that this reworked application will be more manageable and applicable to students.

1.2 E/R Diagram

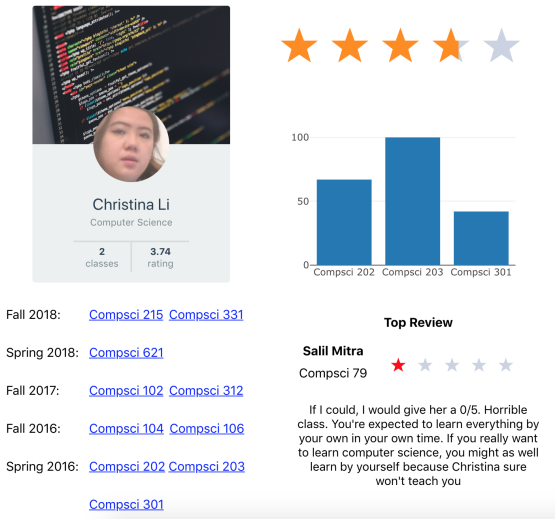


1.3 List of Tables

- Students(SID (Student ID), name, email)
- Taken(Semester, SID, ClassID, CID (Comment ID), Star Number)
- Classes(ClassID, Semester, Name)
- Teach(Semester, CID, ProfessorID)
- Professor(ProfessorID, Name)
- Departments(DeptID, Name)
- Comments(CID, SID, Text, Upvotes, Downvotes)



Timeline



Prof

2 Platform Choice

We opted for a front-end built in React. For this milestone, we mainly focused on two of the main screens: Timeline and Professor.

The first screen allows users to input classes that they have taken each semester, and recommends classes that they should take next semester. Our algorithm is currently very simple – it combines the user ratings of particular classes with this professor, along with the number of students in similar situations (i.e. have taken similar classes up to this point) who have taken the class to give one score from 1-5. It then presents the user with the top four classes in the major that it thinks the user should take next semester. We are currently working on a smarter algorithm that combines multiple classes in one semester, and classes across majors, to give a more in-depth recommendation to the user. The second view we have created is for viewing a certain professor. Here, we

give a general overview of the classes that they have previously taught, their current overall rating, their rating for each specific class, and the most relevant comment/review that they have (the review with the most up-votes). The next scene that we are hoping to implement is Class, where you can view reviews on a certain class, possibly comparing different professors that are teaching the class, along with all reviews for this particular class. We believe that this will be much more useful than RateMyProfessor, as it allows you to view by class and more easily compare to see if you should wait to take a class with a better professor, rather than looking into previous professors that have taken the class, and individually researching each one.

3 Changes Made to Database

Initially, we had to change our E/R diagram and database because we changed our project goals. We took out tables for company, location, and position because our project goals changed where internship experience shouldn't correlate more to bookbagging and when to take classes. We also struggled with figuring out whether there were redundancies in our tables or not; we had to use BCNF to test our database redundancies, but in the end, we did not have any redundancies by having comments be separate and using a foreign key to attach the comments ID to the taken diamond. During performance testing, we have not had to create additional indexes yet, but we will keep this in mind as we continue developing. Once we have a larger database, we will continue testing performance times for our database and make/note necessary changes. We also brainstormed ideas on incentivizing users to submit the necessary data. In the end, we agreed that when users went to select classes that they have taken, they would simply have to give a 1-5 star rating of the class (very simple for the user to do). We were also thinking of introducing a rewards system to gain a large amount of initial data. Essentially, we are going to give users a chance to win a gift-card, so that they would receive one extra entry to the lottery for each review they submit. We believe that this will provide us with the large amount of data necessary to get this project up-and-running.