

CS 312 Project Proposals

Fall 2019

4-bars

Target users: lonely musicians who would like to loop different instruments and just play some music by themselves.

Main functionality: Will contain a few synthesizers that can be played using keyboard keys, and a drum machine. The app will have the ability to layer the tracks and loop them. The idea is that the tracks loop over themselves, so that more complex compositions can be made.

Backend: the back end will handle the Virtual instruments, and the looping music tracks and recording.

API: I have been looking at some API's for this purpose but could not get too much information in time. For the time being, I have found WEB AUDIO API from chrome. This would allow me to make a synthesizer in my web application. It would be very interesting to have a synthesizer that uses touch or click functionality in the browser to make music, instead of keys.

<http://tibersynth.cc/play> for an example of what I mean.

Caroline's Closet

The target users or other stakeholders: People who enjoy fashion and would like more variety with their fashion choices. Also, people who have lots of clothes but find themselves wearing the same clothes every day. It would be particularly helpful for people who travel frequently or live in multiple places (say home and college or children with divorced parents, who live in both houses at different times in the week)

The main functionality of the application. Specifically, you should describe the major "epics", i.e. the big coarse-grained user stories.

For each article of clothing in your closet, you upload information (picture of the piece of clothing, top/bottom, temperature you like to wear it at, maybe the store you bought it at.) The webapp will allow you to plan your outfits and also will generate suggestions. A potential additional functionality could be allowing users to make multiple "closets" or "bags" where you could move clothes between different closets and plan what you want to pack for a trip.

At least as an MVP, this product would be a an app that's designed to appear on your phone screen. I am imagining that the most basic way to upload a picture would be to take a picture of your shirt from your phone. We would need an API to incorporate this functionality. We would probably use AVCam if we're working with apple or Android has a Camera API called "Camera."

Intramural Sports App

Target users: Intramural commissioners and players, although I have not talked to any commissioners on whether this would be helpful

Functionality: The primary functionality of this app is to provide an easy interface for IM commissioners to schedule games and store data about those games, as well as for players to sign up teams and to view the schedule. The app would have two main views that would be accessible through a login, a commissioner and a player. The commissioner would have the ability to open a new season for players to register, activate a random schedule, manage playoffs based on results, and easily communicate with teams. For them, the primary interface would likely be very input based. The players would be able to sign up for a season with their team, view their schedule, view other players and their teams, and would receive email updates for their games. For the players the primary interface would likely primarily be a calendar. Not really sure if having a login for players is necessary/ if having a login for the commissioners defeats the SPA approach.

Back end: The back end would primarily be data storage of people's user information for login, the teams and their players, and emails to be used to send reminders about the games.

APIs: Maybe a calendar based one or an automatic email sending one, not really sure of the availability of these or scope that these would require to just recreate.

Interactive Informational Parking Application

I'm proposing the development of an informational application for parking on campus. Given set user inputs like date, time, student status, and intended parking length, the application will interactively display a map detailing where you can park. The targeted users would be largely students, and potentially also campus visitors. The purpose of the app is not for finding an open parking space, but instead for knowing where one could legally park.

Let's say I return to campus on a late Friday evening on May 2nd. I have a senior parking pass. I normally park in Atwater but all the student spaces in Atwater, Coffrin, and E Lot are full – a classic problem. I'm looking for alternatives close to home. I know the winter parking ban is no longer in effect, but don't know to what extent that affects my options. I know you can park in some faculty/staff spots on weekends, but I also know that some are restricted and will land you a \$50 fine. I'll be moving my car tomorrow morning. I can open the application, enter the relevant details, and see that I can park in a certain row of E Lot spaces. Having a car on campus is a nice convenience but can also be a severe liability. This app would exist to mitigate parking infractions while also giving students peace of mind that they won't wake up to a fine or worse, a boot or towing.

Based on the filtering inputs, the application could use a premade map or the Google Maps API to display a map of campus. Different colored overlays could indicate different possibilities for parking. The user could click these overlays to get a more detailed birds-eye view of the lot to show which particular spaces are open to them.

The backend of the application would be based upon the standard parking rules but could also be manipulated by the administration in case of special occasions/sporting events/emergencies. The project then would require some research into the parking regulations and perhaps cooperation on behalf of the administration/public safety. They might be disappointed that they'd be getting less revenue from tickets, but perhaps it will allow them to focus their attention to more pertinent issues.

Midd Book Market

Target Users: Students looking to buy or sell textbooks

I propose that we create an application that provides Middlebury students with a place to buy and sell used textbooks. Students will be able to post textbooks that they want to sell and input the price of the book, the condition, and any other comments regarding the book. Students can also post inquiries about books that they want. The application pairs students that are interested in buying and selling the same book. Users will also have a profile to enter the courses they have taken and may have the textbooks for. The application can also recommend a selling point for a student, based off of prices that have been scraped from the internet from other book sellers. This application can include more features that will help students get the textbooks they need for cheap.

One outstanding question is how the payments will be made. This may require pulling in an external service or API or finding out how to tie in a service such as Venmo where the application should be able to confirm whether the transaction has been made.

The major stories for this project will be:

1. Creating a student's profile where they can enter the classes they have taken and have the textbooks for.
2. Create a request to buy or sell a textbook.
3. Pairing students together to buy and/or sell books with one another. This will be an anonymous process, until transaction confirmation where both parties can choose to disclose their info to trade the book.
4. Create a page where there are listings for books that are up for sale. These books can be searched by ISBN, class name, or course code. There will also be a page with student requests for books.
5. Scraping price data from external sources.

The back end of this project will hold all of the book data. This will include ISBN numbers, the names of the textbooks, whether or not the books are required, and the course code the book belongs to. The back end can also hold another table that holds price points for the textbooks. This data can be scraped from bookstore sites like Amazon, Barnes and Noble, etc. using the book's ISBN numbers. These prices can be used to provide a book seller with a recommended selling price that will encourage other students to buy from them instead of online. A lot of the requisite data has already been published in a clean format on the Middlebury site at the following link: <http://sites.middlebury.edu/middbooks/>

Online HMMM Assembly Simulator

Target users: CS 101 students and professors

Project: In CS 101, the students learn a little bit of assembly by interacting with a toy virtual machine called the Harvey Mud Miniature Machine (HMMM for short). A reference for the machine can be found here (<https://www.cs.hmc.edu/~cs5grad/cs5/hmmm/documentation/documentation.html>).

Interaction with the machine is done through a command line program which makes it difficult to see what is going on.

An online version of the tool would allow users to upload code or type it in directly. The user could then run the program or step through the code line by line. While stepping through the code, the interface should show the line being executed, as well as the state of the registers and memory.

In addition, the professor should be able to add “problems” with associated test cases. In a format somewhat like CodingBat (<https://codingbat.com/python>), students can try their code and get feedback about if their code passes the tests. The professor should be able to control which tests are visible so that a student will see how many test they fail, but not necessarily be able to see what the test was.

Most of the functionality will be placed on the front end, with the back end used for storing the problems and tests.