EasyPick

A webapp to facilitate choosing classes at MIT

Famien Koko
Jared Hanson
Lara Araujo
Subby Olubeko

Overview

Problem
There are over 2000 different courses to choose from at MIT. Before every semester, students are faced with the same challenge of narrowing down all these options to a set class schedule. Currently, there are a very limited number of ways to decide what classes to take, and most of the existing solutions are inefficient and ineffective.

Existing solutions
Browse the MIT course catalog and read through course descriptions.
- time consuming searching through all the classes, sorted only by course number
- the descriptions don't provide enough insight to what the class is really like

Find statistics about specific classes using MIT subject evaluations.
- no way to filter classes based on data (i.e. find the class with the smallest time commitment)
- no way to discover classes; users have to search the course number to see its evals

Ask friends for suggestions and advice.
- limits discovery to the scope of your friends' class knowledge
- a student's friends might not share the same interests as the student

Our solution
EasyPick helps students find courses in two ways. First, EasyPick will have the ability to recommend classes to students. These recommendations are created using class reviews. Every user of EasyPick will review and rate the classes that he or she has taken at MIT. Then if User A and User B have similar ratings for many classes, User A will be recommended classes that User B thinks highly of, and vice versa. This system of recommendations is based on the idea that you will like the same classes as students with similar interests to yourself.

EasyPick also allows students to search, sort and filter classes based on data from class reviews. For example, with EasyPick it becomes easy to find all HASS-A classes, listed from least time commitment to greatest. This eliminates the need for students to mindlessly browse through hundreds of listings in the course catalogs before finally finding a class that even fits his or her specific requirements.

Concepts

While designing EasyPick, we identified five main concepts. Below, we discuss each concept individually and elaborate on their purpose, operational principle and anticipated misfits.

1. *Review*
   Purpose: To provide a source of data for the app which will ultimately be used to provide search results and recommendations to users
   Operational Principle: Upon registering and at the end of each term, users fill out forms giving feedback for each course they have taken
   Anticipated Misfits:
      - Users never fill out evaluations.
      - Evaluation data is not always helpful

2. *Student Profile*
   Purpose: To maintain individual data for each user which is used to determine their background and interests
   Operational Principle: The app prompts registered students to fill out a personal profile that details what classes they have taken and when they took them. Students are also prompted for their academic details, e.g. their major(s), minor(s), fields of interest, HASS concentration, and topics they've conducted research in.
   Anticipated Misfits:
      - Interests are too broad or too specific.
      - User has interests that are not listed.

3. *Recommendation*
   Purpose: To simplify the task of deciding which courses to enroll in for MIT students by suggesting classes for each student based on data from users with similar qualities. This concept is the main source of value provided by our app.
   Operational Principle: Once a user has completed their profile and submitted evaluations for their courses, the app identifies a list of students whose backgrounds and interests are most similar to that of the user. These students' reviews are then considered and the app suggests that the user register for classes that the similar users enjoyed.
   Anticipated Misfits:
      - Unique students may receive recommendations that are overly broad and impertinent since their preferences won't be closely related to those of many other students

4. *Course Stats*
   Purpose: To provide users with helpful statistics about the students and faculty that participate in specific courses and the overall ratings of these courses over time
   Operational Principle: After a sufficient number of students have submitted information about a given course, the app calculates various metrics representing the

characteristics of students who took said course and the general feedback for the course at each offering
Anticipated Misfits:

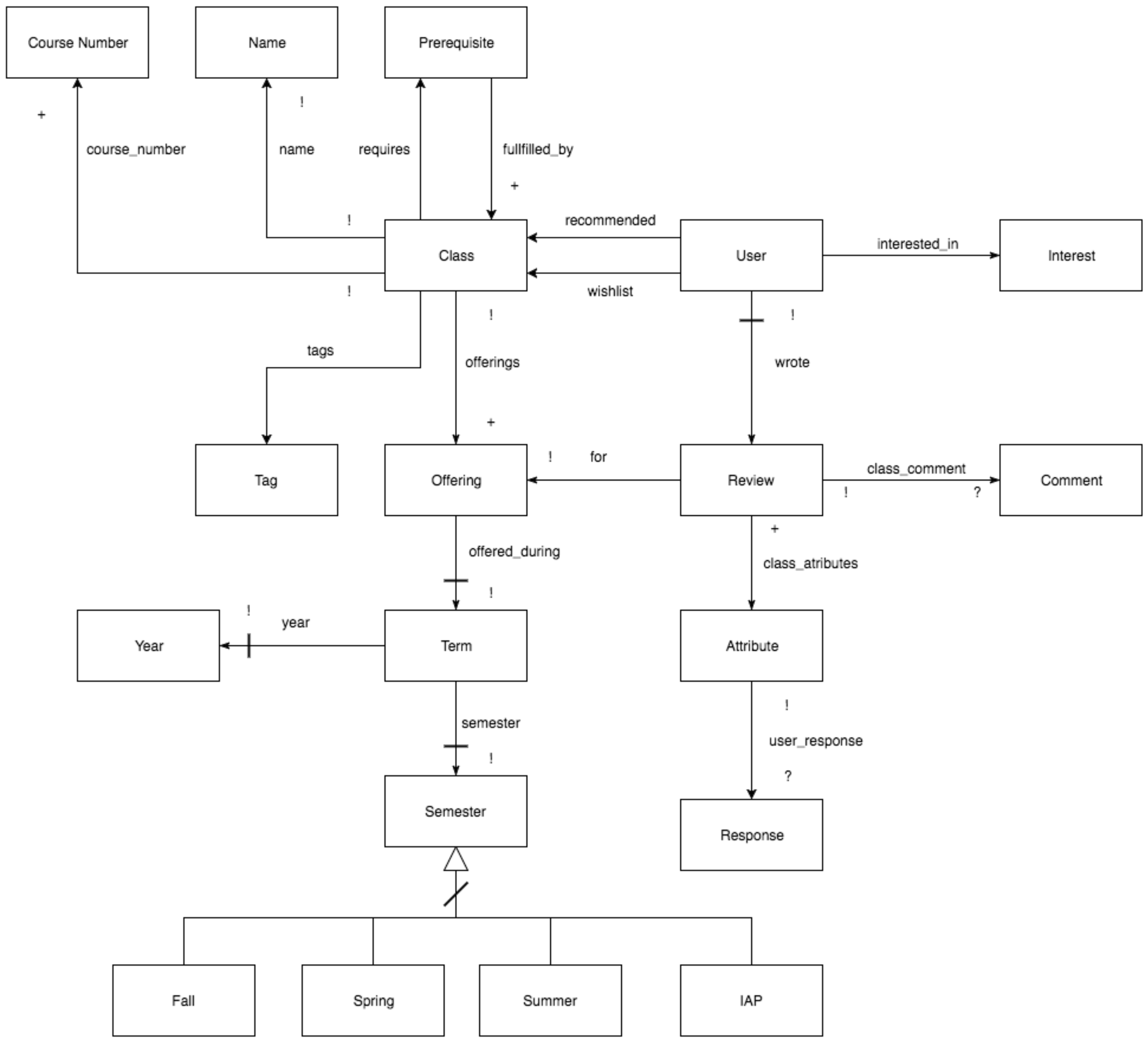
- Small classes or classes that are offered sporadically might have statistically insignificant data.

5. *Comment*

Purpose: To supply users with richer information by allowing students to publicly give more detailed feedback and justification for their evaluations of courses
Operational Principle: Students who have taken a class voluntarily and anonymously post comments recounting their experiences and opinions on the class. These comments can be upvoted or downvoted by other students to indicate their usefulness and accuracy.
Anticipated Misfits:

- Even with anonymity, the ability to upvote and downvote can lead to the creation of a toxic environment.
- Students aren’t being incentivized to write meaningful comments.

Data Model



Textual Constraint:

1. Users can only have 1 review per class
2. A class cannot be required by itself or any of its prerequisites (must be acyclic)

Explanations:

1. An interest is an area of work or study, such as machine learning, web development, etc
2. An attribute is a quality of a class such as rating and average number of hour spent a week.
3. Tags are special characteristics of a class like HASS, CI-H, etc.

Insights

1. A user can have no reviews, which in turn can have no attributes and no comment, but the more data is filled on these, the better recommendations a user has

Security

As with any application, there are certain security vulnerabilities and risks that we need to be aware of when building EasyPick. First of all it is important to understand that our user base will only be MIT students or affiliates, since only @mit.edu emails will be given access to registering an account. We see this as an advantage because we are limiting who can join, which inherently limits some potential attackers from joining our site. We are still aware of a few different ways that attackers could make an impact on EasyPick though.

General Attacks:
We are mainly concerned with ways potential attackers could corrupt our user's and class' data. We mitigate these potential risks in a few ways. First and foremost, we will only allow access to any of our pages, besides the register and login page, to properly logged in users. Additionally, as mentioned earlier, only allowing students with MIT emails to register reduces the possibility of an attacker registering multiple different accounts in order to skew the data for one or multiple classes. We eventually plan on rolling out the registration and login process using MIT certificates, because we have still identified the possibility that potential attackers could register mailing lists as emails to create multiple accounts, but that would come after the MVP. We will also send registration emails to prevent attackers from using a fake email, or someone else's.

To prevent a single malicious user from skewing data for one or multiple classes, we plan on only allowing one review per class per person, and we will only allow one comment per class per user. We will also have a limit on the number of classes that you can review because there is definitely an upper bound to the number of classes that a single user can possibly take at MIT. By identifying users who are at or above that upper bound, we may be able to find cases in which a user is making false claims about taking one or more classes.

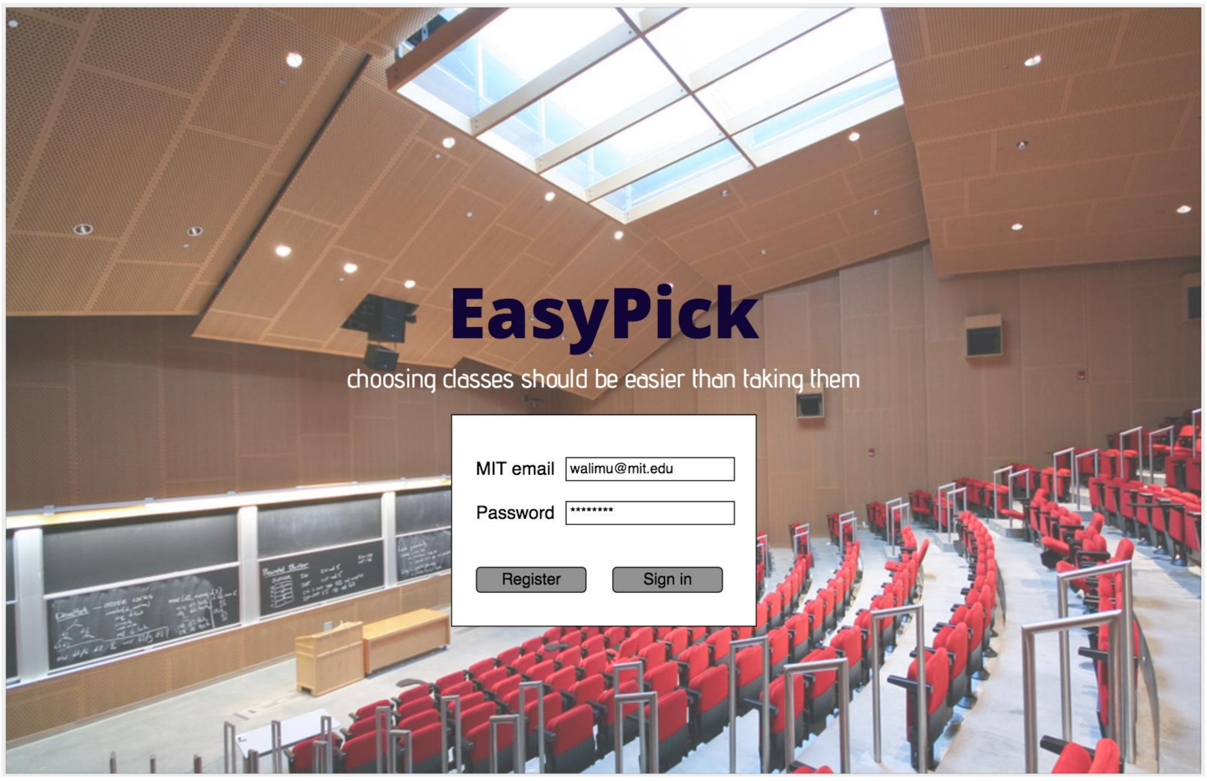
To keep users information safe, only User X can view User X's personal information and profile. Additionally, since we want comments on class posts to be anonymous, but we still want to only allow one comment per class per person, we will store hashes of the user who commented, as opposed to identifying information about that user. This makes it so even if an attacker had access to our comments database, they would not be able to uniquely identify who commented what, while we still are able to hold the restraint of one comment per class per user. Finally we will make sure only to store hashed passwords in our database.

XSS & CSRF Attacks:
In order to protect our web app from XSS and CSRF attacks, we will do a few things. First, all form data will need to be sanitized (via whitelisting) before being input into the database, or displayed to users, so that scripts can never be run when users are unsuspecting, and injections to our servers can be avoided. Next, we will require all requests have CSRF tokens associated with them to mitigate the risks associated with those types of attacks. Finally, we will periodically log out all inactive users.

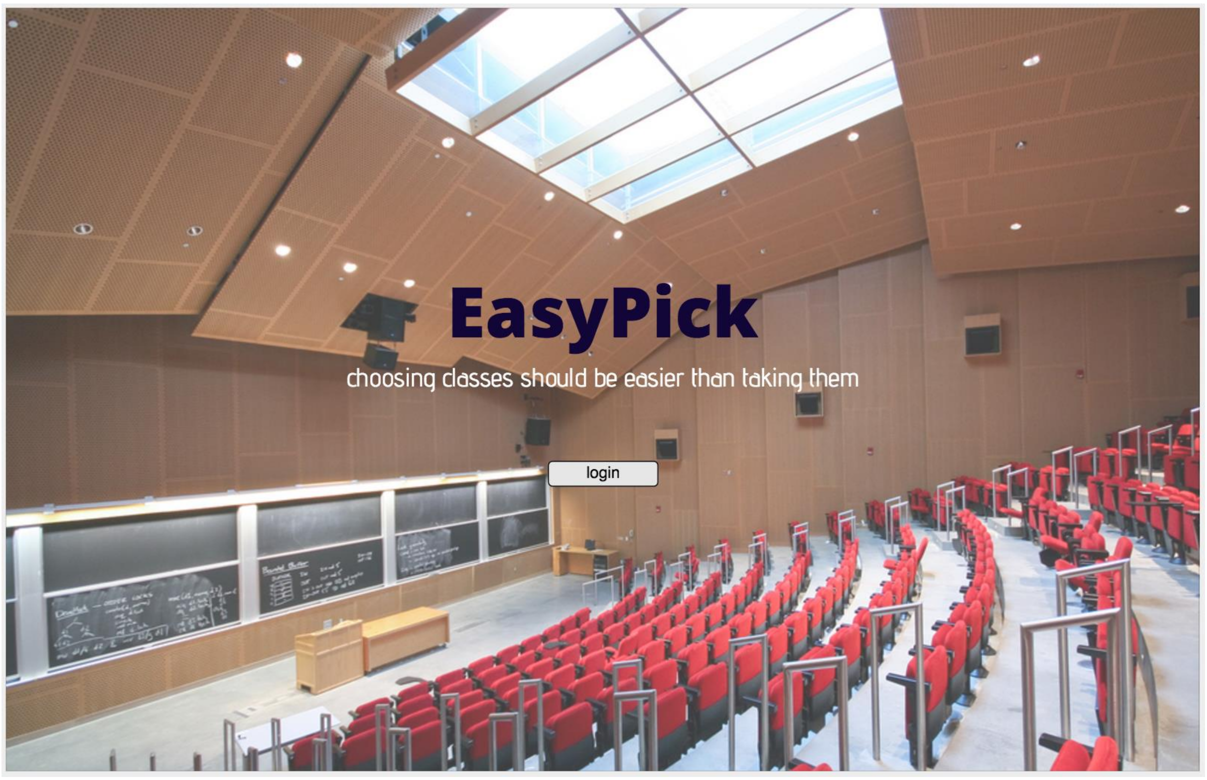
EasyPick needs to be well protected against all types of attacks because we rely on responses and user input that many may not want in the public domain. Although we don't see a huge risk from potential attackers, because this product is limited to the MIT community, our assumption could be proven wrong and we need to be well protected anyway.

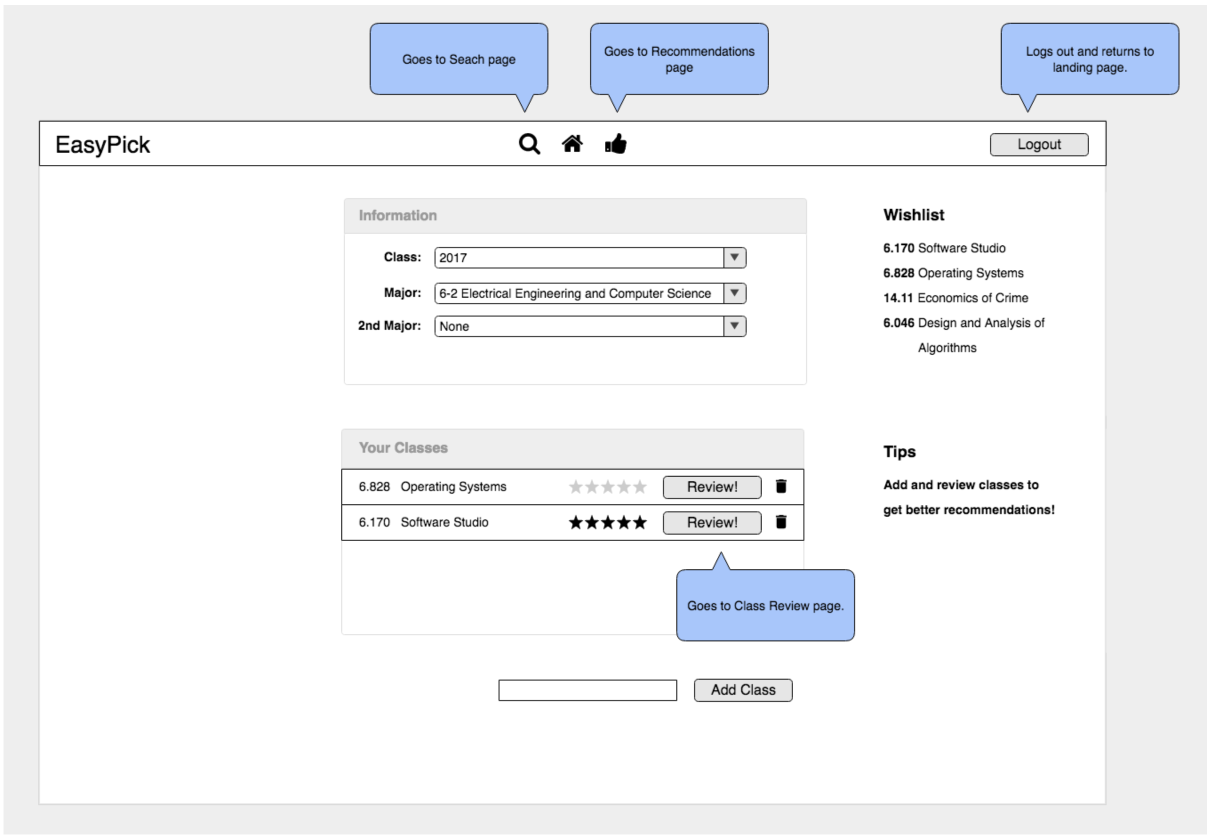
Wireframes

Moqup project: https://app.moqups.com/laratimbo7@gmail.com/npovpKLXMg/view



Login page using MIT email and password (none of us have ever developed authentication using certificates, so for the MVP we plan to use standard email/password login)





Goes to Seach page

Goes to Recommendations page

Logs out and returns to landing page.

EasyPick

Logout

Information

Class:	2017
Major:	6-2 Electrical Engineering and Computer Science
2nd Major:	None

Wishlist

6.170 Software Studio
6.828 Operating Systems
14.11 Economics of Crime
6.046 Design and Analysis of Algorithms

Your Classes

6.828	Operating Systems	☆☆☆☆☆	Review!	🗑
6.170	Software Studio	★★★★★	Review!	🗑

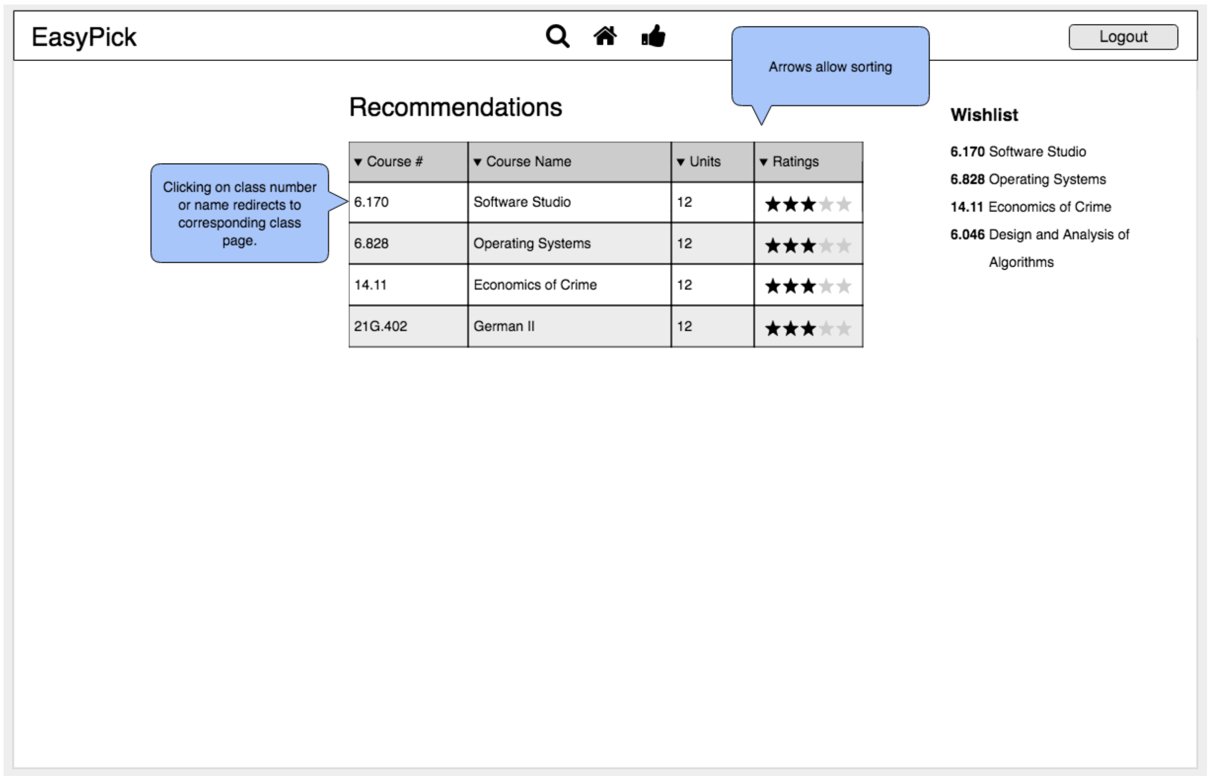
Goes to Class Review page.

Tips

Add and review classes to get better recommendations!

Add Class

User profile page



EasyPick

Logout

Arrows allow sorting

Recommendations

Clicking on class number or name redirects to corresponding class page.

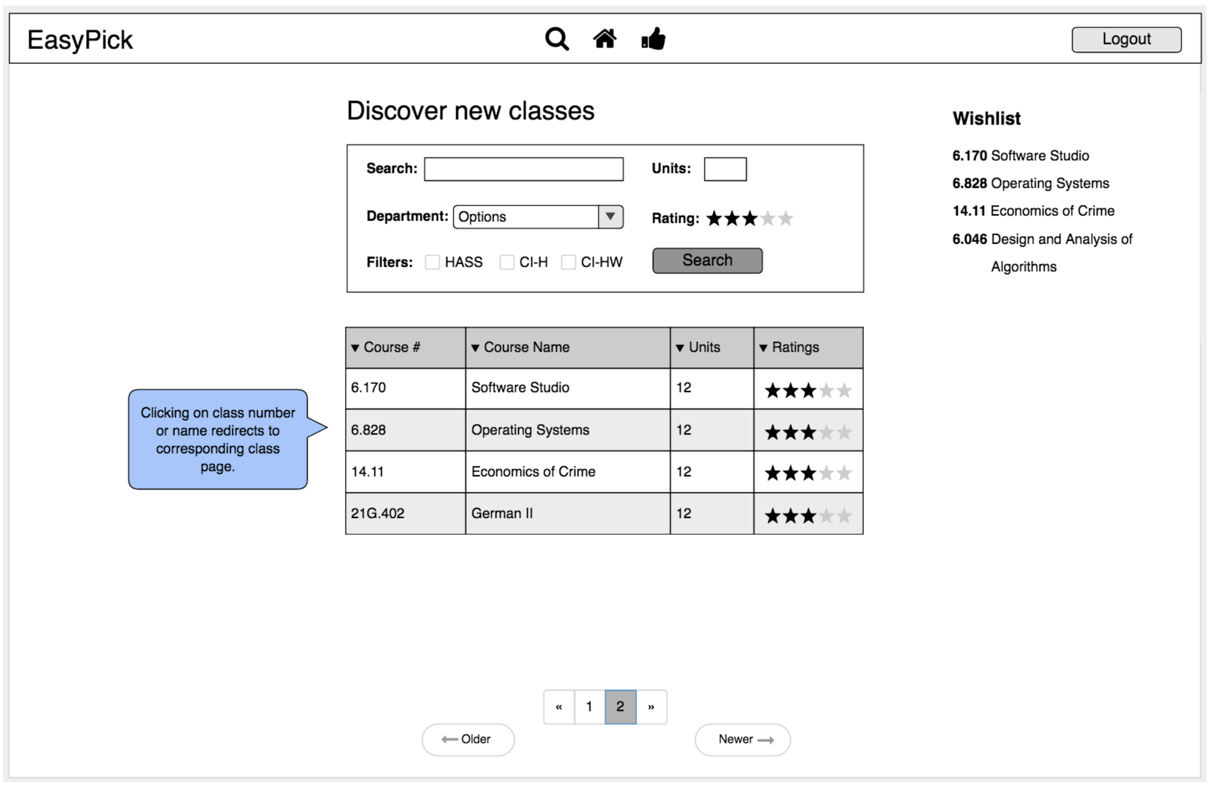
▾ Course #	▾ Course Name	▾ Units	▾ Ratings
6.170	Software Studio	12	★★★☆☆
6.828	Operating Systems	12	★★★☆☆
14.11	Economics of Crime	12	★★★☆☆
21G.402	German II	12	★★★☆☆

Wishlist

6.170 Software Studio
6.828 Operating Systems
14.11 Economics of Crime
6.046 Design and Analysis of Algorithms

User recommendations page

Search page



Discover new classes

Search:

Units:

Department: Options

Rating: ★★★☆☆

Filters: ☐ HASS ☐ CI-H ☐ CI-HW

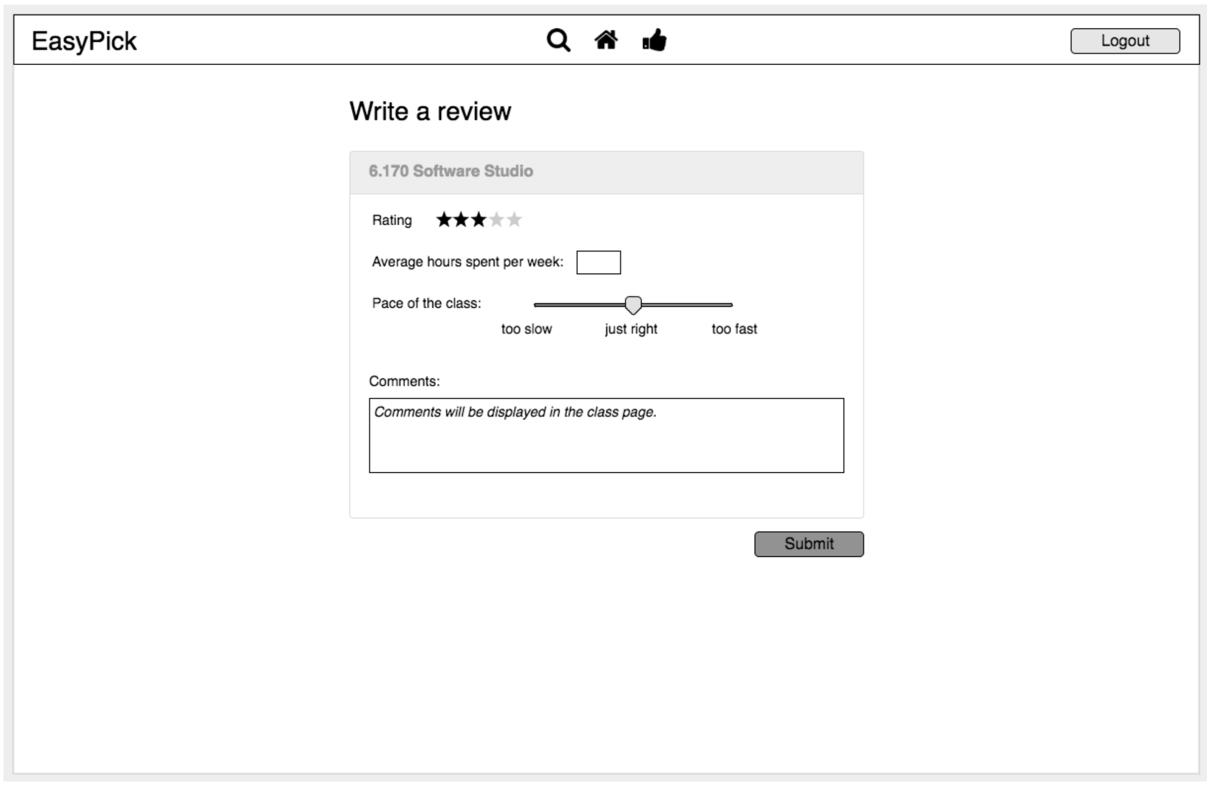
▾ Course #	▾ Course Name	▾ Units	▾ Ratings
6.170	Software Studio	12	★★★☆☆
6.828	Operating Systems	12	★★★☆☆
14.11	Economics of Crime	12	★★★☆☆
21G.402	German II	12	★★★☆☆

Clicking on class number or name redirects to corresponding class page.

Wishlist

6.170 Software Studio
6.828 Operating Systems
14.11 Economics of Crime
6.046 Design and Analysis of Algorithms

Search page



Write a review

6.170 Software Studio

Rating ★★★☆☆

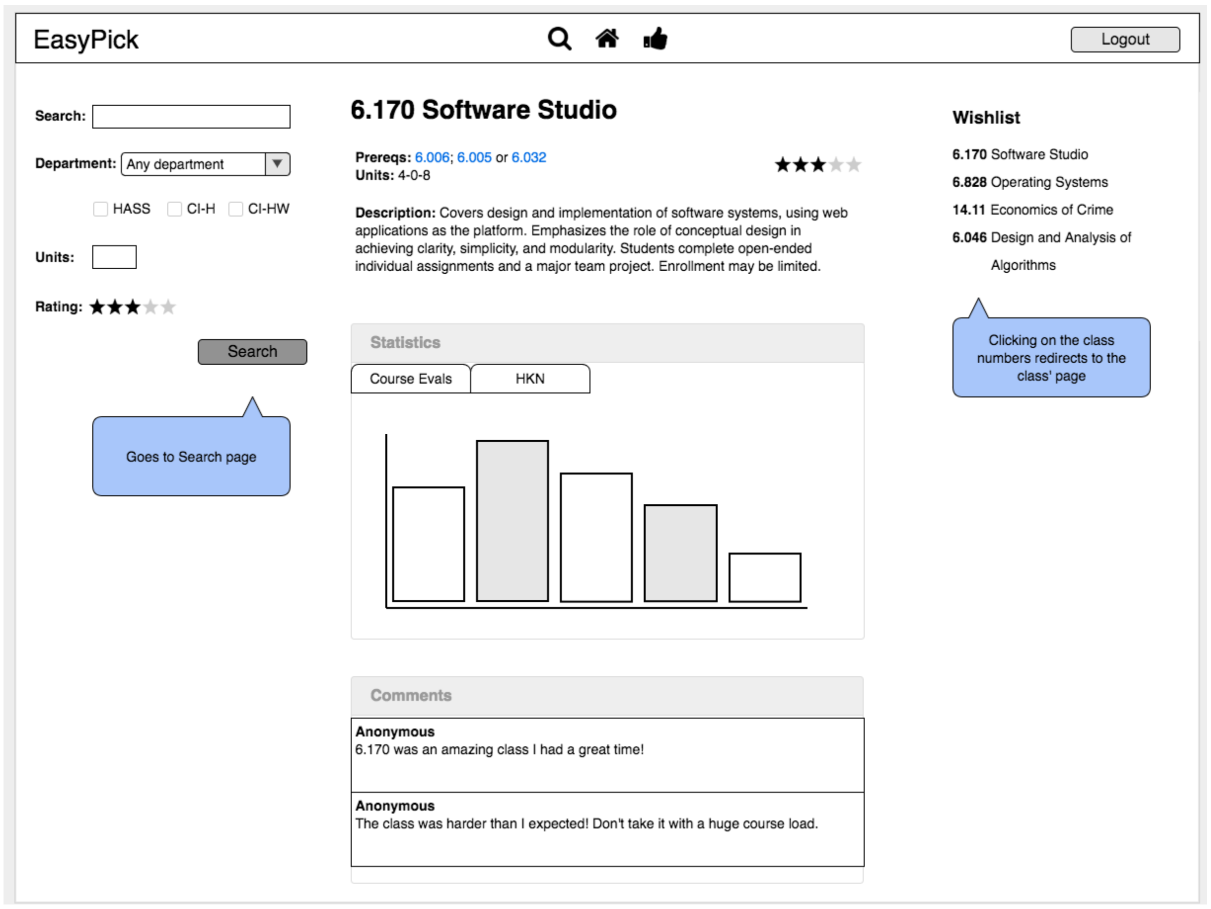
Average hours spent per week:

Pace of the class: too slow · just right · too fast

Comments:

Comments will be displayed in the class page.

Class review page



EasyPick

Logout

Search:

Department: Any department

HASS CI-H CI-HW

Units:

Rating: ★★★☆☆

Search

Goes to Search page

6.170 Software Studio

Prereqs: 6.006; 6.005 or 6.032
Units: 4-0-8

★★★☆☆

Description: Covers design and implementation of software systems, using web applications as the platform. Emphasizes the role of conceptual design in achieving clarity, simplicity, and modularity. Students complete open-ended individual assignments and a major team project. Enrollment may be limited.

Statistics

Course Evals | HKN

Comments

Anonymous
6.170 was an amazing class I had a great time!

Anonymous
The class was harder than I expected! Don't take it with a huge course load.

Wishlist

6.170 Software Studio
6.828 Operating Systems
14.11 Economics of Crime
6.046 Design and Analysis of Algorithms

Clicking on the class numbers redirects to the class' page

Class page

Design Challenges

Design Risks

The three main risks we identified with our design are about user engagement, quality of recommendations and data privacy.

- **User engagement**
  Our app relies on users providing us with reviews about the classes they took while at MIT. If users are not encouraged to write reviews, our app will perform poorly.
  To mitigate this problem, we decided to make the reviewing process as small and simple as possible. Moreover, we stimulate a positive feedback loop by actively telling the users that the more they contribute the better recommendations they will get. Finally, we plan to use course evaluation data to create initial engagement.

- **Quality of recommendations**
  A core feature of our application is being able to give good class recommendations to the users. In order to do that, we plan to use a collaborative filtering technique to make recommendations based on users with similar profiles.
  For our similarity measure, we thought about considering how similar did two users ranked a given class. However, this comparison might be misleading. For example, after sophomore year most course 6 students have taken the same foundation classes and are bound to have somewhat similar reviews.
  To mitigate this problem we allow users to express broader interests such as 'web development' or 'music theory' in their profile. To avoid dealing with unstructured data, the set of interests a user can choose from will be predefined in the application.

- **Anonymous Comments**
  We decided that users comment on classes anonymously. We believe that this allows students to give more honest feedback about the class. However, this also creates the risk of possibly generating a toxic environment in the comments section.
  To mitigate this problem, we will try asto prevent comments with possible toxic language from ever being posted.

Design Choices

Below we discuss some choices we made while designing EasyPick. We omitted design choices like using MIT certificates for login and allowing anonymous comments since these have already been discussed in previous sections.

1. **Offering set**
   While designing our data model, we had a discussion about including an Offering set or not. At first, we were unsure whether this set was needed since we have no functionality that relies on it and it creates an additional level of indirection between a Class and its Reviews. However, we ultimately decided to include the Offering set because it allows us to have more fine grained data about each class. Moreover, we believe that the performance overhead caused by the additional level of indirection is minimal.

2. **Class reviews as a way to keep track of classes taken**

Our data model does not have a relation 'classes_taken' between Users and Classes. Instead, we use the reviews of each user to determine which classes they took. Again, this adds a level of indirection between users and the classes they took, but this design choice makes it easier to reason about reviews and classes simultaneously without worrying about whether an user's set of reviews is consistent with their set of classes taken.

3. **How to make recommendations**
   We spent a lot of time deciding how our algorithm for making recommendations should work, since that would determine what type of data we would store. Ultimately, we decide on a collaborative filtering technique where we first find users with similar interests and then recommended classes that these users enjoyed. We decided on this algorithm mostly because of its simplicity. The algorithm is relatively simple to reason about and it can implemented using well-structured data.