the Yale Projects Board
CPSC 490: Senior Project

Author:

Bobby Dresser
B.S. Computer Science, Expected May 2015
bobby.dresser@yale.edu

In collaboration with:

Rafi Khan
B.S. Computer Science, Expected May 2015
rafi.khan@yale.edu

Advisor:

Daniel Abadi
Associate Professor of Computer Science, Yale University
dna@cs.yale.edu

http://yale.projectsboard.io
1. Abstract

Yale is home to a diverse set of energetic entrepreneurs, interesting academic projects, and talented programmers. Even though the community is large, individuals with meaningful ideas can have trouble finding the talent necessary to build their project into reality.

The Yale Projects Board was created to facilitate collaboration between students with programming skills and individuals with ideas. By organizing a central database of projects on campus with an elegant and easy-to-use search function, the Projects Board can become an essential tool to get any project from ideation to reality. Those with specific projects in mind—whether they’re students looking to start a company, scientists conducting an experiment, or professors building something for research—can post their project on the site and outline specific roles they’re looking to fill. Students with programming talent can browse the available openings using a robust set of search features to find a job that suits their skill set and interests.

The Projects Board takes advantage of the trust built-in to the Yale community. CAS login is required to use the site, which provides a baseline of privacy and quality control. The concept is not Yale-specific, though: the platform could help collaboration in any university or neighborhood, and we hope to branch out to new communities using the insights we gather from the Yale pilot.

We hope the Projects Board will let Yalies work together to build amazing things across a number of disciplines. Please try it out yourself at yale.projectsboard.io. Any feedback to improve the product is greatly appreciated!
2. System Design

The Projects Board is designed to be very robust over a specifically defined set of features. The site was conceptualized from the bottom up with a sharp focus on the core functionality: connecting people. The application’s features are all a result of two main goals: (1) make information as clear and easy to access as possible, and (2) encourage contact between two users with minimal friction.

Anyone in the Yale community (with valid CAS credentials) can:
- See all openings posted on the site
- Filter, sort, and specify searches to find what they’re looking for
- Create a personal profile with a short bio, list of relevant programming skills, and contact information
- Create a new project
- Create specific job openings for projects they’ve created
- Keep a list of starred openings for organization or later reference
- Contact other users of the site via email

For site maintenance and project approval, the site allows for administrators with extra privileges. Quality control is an important part of this project and will be discussed later. Users with administrator status can:
- Edit or delete projects
- Edit or delete specific job openings
- Approve projects so they may appear in search

A component of the site is available to the public for demonstration purposes. Even without CAS login, anyone on the site can:
- See and sort through job openings
- View contact information for site creators
- View “about” page for information on the project

2.1 Profile Page

A profile page is created for each user who logs in to the site. The profile page provides space for customization: the user may fill out a short bio, add programming languages to their list of skills, and include links to their Github, Linkedin or personal website. These profile pages are also available in search.

2.2 Creating a Project

Anyone logged in with CAS can create a project, which will later become the parent to specific job openings. A project simply consists of a title and a description of the project’s purpose and goals. We decided including a high-level description of the overall project is essential to any specific job description, and have found many job sites do a poor job of establishing this relationship.
2.3 Creating an Opening

Once a user has created a project s/he may add specific openings to the project. These openings are the specific roles the leader is looking to fill – anything from “Frontend developer” to “Graphic Designer” to “Full Stack Engineer”. The opening allows for a title and description so the project leader may include as much information as s/he chooses. The opening also allows for the leader to list skills the opening calls for, chosen from a provided list of common technical skills. The opening also has a field to describe compensation and timeframe, allowing for term-time, summer and full-time jobs at whatever pay the user specifies.

3. Interaction Design

Users of this platform can generally be bracketed into two categories: those posting information to attract job applicants, and those who are searching for jobs. The site’s success is dependent on user content, so getting new users to post openings is crucial to its success. Job openings and applications are typically taken quite seriously, so establishing a sense of reliability and trust is also essential. With these goals in mind, we divided the user experience into four workflows, each designated by a tab on the navigation bar.

3.1 Explore

We want to make job openings as easy to see as possible. Rapid discovery of new and relevant content is important to draw users back to the site. So upon login, the first page to load is a complete list of all available openings sorted by date. The newest openings are visible first, and the rest load when necessary as the user scrolls. Without any interaction, and without having to search or click, the user is able to see all jobs available. Openings are presented in a three-column layout that is consistent across the entire application.

Although immediately engaging, this does not present the user with content relevant to his/her own interests or skills. That’s where search comes in: the user can input text that matches any skill, opening name or description, or project name or description—anything from “econ” to “javascript” to “iOS”—and without a new page load (thanks to Angular) a list of relevant results is returned. For further control over search results, the user can select from a list of “more options” to display users or openings; to sort by pay or newness; or to show or hide jobs specifically for the term, the summer or full-time.

With this model, we can allow for incredibly specific queries while also encouraging discovery of new projects or ideas. There is no hidden information: openings are all available, and the search function behaves like a filter to narrow down the list of openings based on the user’s needs.

3.2 Post

We also want to make posting a new project completely frictionless. One of the four navigation tabs is specifically dedicated to creating a project: it opens the project page with a blank title and description and places the cursor inside the title bar. As long as the project has a title and description, the user can save it and await approval (described below). Openings behave similarly – with two clicks and no load time, the project’s leader can add job openings available to the entire Yale community.
3.3 Contact

The contact feature is really what the application is built for, so each opening has a large amount of screen space dedicated to said button. If the user is interested in a specific opening, s/he is invited to make contact with the opening’s poster by using this feature. The “contact” button sends an email to both parties with a brief introduction and statement of interest. The project leader can check out the applicant’s profile on the Projects Board, which may include a resume, etc. and it’s up to both parties to make something happen!

3.4 Profile

Every user gets a profile when s/he logs in with CAS. The profile has room to include a short bio, select skills from a pre-determined list of common programming languages, and add links to github, Linkedin, and a personal website. The user can also upload a resume. We strongly encourage users to fill out at least a basic amount of information, and we have implemented persistent flash messages to help remind them. Each project also includes a link to the user who created it, so the project leader’s profile can also help give some background on the project itself.

3.5 Starred

Eventually we hope the Projects Board will have a high volume of openings that appeal to a wide range of students. With this high volume, it may be difficult to keep track of specific openings a user has seen in the past. We created a “starred” feature to help users keep tabs on openings they like or have applied for. It’s simply an angular directive that allows the user to toggle a boolean for each opening. If the opening is starred, it is added to the “Starred” tab on the navigation bar for later reference. We hope this will keep the users’ experience orderly and convenient.

3.6 Project Approval

To ensure the projects posted on the site remain free of spam, we have created an approval process for all projects. In this instance, we had to make a trade-off between simple project flow and quality control. After a project is created, the user is notified that they may have to wait up to 24 hours for their project to be approved. An administrator of the site (currently Rafi and I) gets an email, and we read the project over to make sure it is legitimate. The user is then notified when his/her project has been approved, and it immediately appears in search results.

3.7 Summary

In general, we hope to deliver on our stated goal of making the site as frictionless and obvious as possible. CAS login provides a huge advantage to getting users in and set-up on the site. The implicit trust of the Yale community creates a fantastic platform to advertise projects. We hope the interface we designed mirrors the simplicity of the idea and platform.
4. Technology

The Yale Projects Board was built using a Ruby on Rails framework for the back-end and the Angular JS Javascript framework on the front end with SASS for CSS. The project relies on several Ruby gems (open-sources plugins for Ruby) to help with CAS login, search, and resume upload. The site also takes advantage of Angular’s two-way data binding to allow for instantaneous saves with little to no noticeable delay to the user, immediate reactions to user input, search results without a page load, and greatly simplifies the structure of UI elements.

4.1 Data Model

The data model has five distinct components, as dictated by the architecture of the application on the whole. They are: user, project, opening, favorite, and skill. Most of the back-end to this project was implemented by my partner, Rafi Khan, and further details of its design are outlined in his project report.

5. Analytics & Analysis

This user interface was designed with specific goals in mind, but we were only able to gather qualitative feedback from peers and friends. Fortunately, once the product is used by a portion of the Yale community, we can gather quantitative feedback from their usage patterns to understand what works and what doesn’t. This will allow us to iterate on the product to make it even easier to use. It will also provide some insight into UI/UX fundamentals.

Besides feedback from how users interact with the Projects Board, we can also learn from the actual content posted on the site. Ideally, the Board will be an index of all extra-curricular tech projects taking place on campus, and aggregation of that data can tell us some interesting things about what Yale students and faculty are interested in. We can learn what subject areas (commerce, psychology, social platforms, etc.) are the most popular, and we can find out what skills are most in demand. If the Board continues to exist over time, we could also gain some insight into changing technological trends.

The site uses Google Analytics spread to track page loads, specific events (explained below) and general goal flow.

5.1 Conversions

The point of the Projects Board is to initiate contact between two parties. With this as the stated goal, we can easily track conversions as the number of users who eventually contact a project poster regarding an opening. (Whether the student who makes contact is hired is obviously up to the project leader.) Ideally, we would see a high conversion rate, indicating that users find the site useful and trustworthy. Obviously, this is a combination of several factors not directly relevant to the site’s UI, including quality of projects and number of openings.
5.2 Discovery

Search is one of the Board’s most prominent features. To understand how users interact with this feature, we can track the number of search queries, popular searches, and the number of users who use the additional search options. The search is built with few instructions; it is almost implied that you can search by whatever keywords you want. However, if users don’t search by content and instead only search by skill (or vice versa) we can easily understand this feature needs more explanation.

5.3 Patterns

In general, we can observe users (or groups of users) behavior through the different modules of the site. Hopefully users will linger on the search page, fill out their profile, visit several project pages, and eventually make contact with one or more projects. If one of these steps is noticeably absent from the flow of the user’s experience (or noticeably too-present) than we can look at the factors that may have caused such behavior.

6. Acknowledgments

The author of this report would like to thank his partner, Rafi Khan, for his humor, good planning, and flexibility. He would also like to thank his friends and Professor Abadi for providing helpful feedback throughout the process.