

# Project Proposal: Academic Job Market and Application Tracker Website

Cengiz Gunay

September 4, 2012

## Summary

Applying for academic (e.g., tenure-track) positions is different than regular job applications, but there is no general website (e.g. like Monster) to help in this process. The applicant must make a large number of applications because of their inherent low success rate, and each application is time consuming because of the required additional research and communication. The application itself consists of a multi-stage process of steps like: submission, sending recommendation letters, follow-up requests, and interviewing.

The amount of work required by the applicant in this process can be significantly reduced by a website that allows community collaboration and computerized help in several different stages of the process, for instance by helping to research for information, to track submission status, and to send reminders.

Although tenure-track applicants constitute a narrow audience, when diverse types of academic fields are considered, such a website can host a large community and can be successful with the right business model.

**Client:** Cengiz Gunay

## Problem Definition

A successful academic application often requires a customized application package and some research and communication to investigate the specific needs of a department and search committee. Often the communications involve sending official and/or friendly inquiries to acquaintances and colleagues. The research involves reading about the school, the department and faculty members, to identify one's strengths relevant for the position. Information from both of these sources is then used to decide if the applicant is a good fit for the position and to custom tailor the application.

Some of this research is highly repetitive, and burdened by the non-standard format of academic websites and the nature of these types of announcements. To help with this burden, there are;

- several academic job announcement websites that are general (e.g., [Science Careers](#), [Nature Jobs](#), [HigherEdJobs](#)) or domain specific (e.g., [Computing Research Association's Job Service](#), [IEEE Job Site](#), [ACM Career & Job Center](#), [NeuroJobs](#)), which serve announcements in one place and allow filtered searching and email alerts;
- specialized job application websites (e.g., [AcademicJobsOnline](#)), which only allow applying to positions posted by few registered employers;
- few online forum discussion boards (e.g., [Chronicle of Higher Education website forum](#)), which is position independent but nevertheless a very significant source of information; and

- wikis (e.g., [Academic Jobs Wiki](#)), which are intended for manually entering information about positions and feedback from people who are in the application process. However, these are not heavily used by job seekers.

However, none of these provide a standard platform allowing to connect the announcements and the user comments, and to have a customized application process into one place to help find announcements to apply. Having them in one place, an applicant can easily compare different opportunities and track them.

Once an application is selected and initiated, a second burden comes from keeping track of deadlines, recommendation letters, and personal communications. This burden is multiplied by the large number of applications one has to prepare because of the nature of the academic market (see [article "For Science Ph.D.'s, There Is No One True Path"](#)).

What can help this burden is an automated system of tracking items specific to private applications like email communications and send reminders and notifications to the users. However, this system must be more customized than a generic calendar application.

## Proposed Solution

The burden of researching specifics of a position announcement can be reduced by a community-driven website. As positions are announced, registered website users can create, edit, and comment on positions and the school units they are hosted from, like departments, schools, colleges, and universities. Like other community or forum websites, it should protect itself by allowing to establish users' reputation and rank in the community and also allowing to flag abusive use.

The list of positions would become easily searchable and cataloged thanks to a community effort like tagging with keywords. While the list of positions is public, the applications will be tracked in a private section of the website where the applicant can start *following* certain positions and initiate an application. Then the website can help track deadlines with a reminder system, and more importantly a user interface that clearly shows the applicant's most immediate tasks. The user can also use the tags to customize each application (e.g., curriculum vitae, cover letter).

The proposed functionality is not provided by any one of the above available websites. However, the proposed website will only be useful if the burden of filling in information can be traded off with the added value gained from it. For this careful user feedback must be collected during development from potential users (e.g., Ph.D. candidates and post-doctoral fellows). Only with large participation a community website can properly established.

Ideally, with large participation and large number of visitors, the website can potentially profit from advertisement revenue. Furthermore, paid sponsor accounts can be given to job providers for making official announcements.

## Planned Work

In one semester, a proof-of-concept version of this website can be built by focusing on supporting a few academic subdisciplines and by getting feedback from a focus group rather than aiming to create a large community. To make this project more manageable, the project team can use an existing preliminary database structure that contains a data model for position announcements, application tracking, and personal communications. However, this data model is missing website user information, comments, and information about academic units (e.g., departments).

A minimal implementation of this project should include as many as possible of the below features:

1. A main website user interface that allows accessing the following components
2. A component to enter job position announcements and link them to academic units.
3. A component to search and browse existing positions recorded on the website.
4. Ability for other users to improve the details of the position and academic units, and to comment on them.
5. Select positions to be applied for in a user's private section on the website.
6. A component to list saved positions by deadline, and set reminders.
7. Allow adding additional private comments and tags to these positions before starting an application (research phase).
8. Allow annotating private communications related to a position.
9. A component to enter application tracking information connected to a selected position.
10. A component to see status of applications, update them and get reminders about their status and related communications.
11. Optional: Options to share partial information from the private application section to let other users about certain aspects of the application process experienced by a user.
12. Optional: A complete forum section where users can discuss topics and send each other private messages.

To achieve a minimally working website, the project team will need to choose an appropriate web application framework and platform. There are two recommended platforms:

1. **Google App Engine**: A very high level service that uses Python or Java.

Pros:

- It is easy and simple to program and deploy (see [demo](#)).
- Python has a wide library and is known by most team members.
- Hosting this project will be free because of its size.
- There is an [existing forum software, Fofou](#), that can be modified. It is based on the forum software [FruitShow](#), which is written by the software engineering guru and our class reading author, [Joel Spolsky](#).

Cons:

- By default it does not use SQL.

2. **Apache-MySQL-PHP (AMP)**: Free, open source industry standard for developing web applications.

Pros:

- Lots of existing AMP code for web apps.
- Lots of existing online tutorials, examples and information.

- Can use existing partial MySQL database schema developed for this purpose.

Cons:

- Hard to install and maintain runtime environment.
- Hard to learn, develop and modify existing code.
- None of the team members have experience in components of AMP, and especially the programming language, PHP.

Once a software platform is selected, the project team members then each must fulfill one or more roles:

**Software architect:** To define how multiple components of the software interact and how the control flows between them.

**Data modeler:** For display and internal storage, by taking advantage of existing database structures.

**User interface designer:** Easy finding of existing information, and allow flexible editing and user comments, similar to that of social media sites.

**Software developer:** To devise algorithms and integrate other components together.

**Testing lead:** To design and apply a testing strategy

**Documentation lead:** To make sure the project and the code is properly annotated sufficiently to be transferred to another software team at the end of the semester.

The project team must accomplish this using an Agile Computing approach while getting continuous feedback from the client, and less frequently from a focus group of Ph.D. candidates and postdoctoral fellows, who constitute the intended audience for this product.