

CMPE 275 Section 2 Spring 2017

Term Project

*Last updated: 04/15/2017 (status: **published**)*

In this project, you will build a job board for companies to post jobs and for job seekers to search for jobs. This must be a web app hosted in the cloud. The primary language you use for server implementation *must* be Java. You do not have to use Spring, but you need to exercise the principles, patterns, and methodologies that you have learned in the class, such as DI, AOP, MVC, ORM, and transactions. You *must* use either a relational database, unless you choose Google App Engine or Compute Engine, in which case you can use Google Cloud Datastore if you prefer.

If any feature described below is unclear or ambiguous, and you fail to get a clear answer from the instructor or TA, you can use your best judgment to interpret it and add the missing details, provided that you clearly document and explain your reasoning in your project report.

Functional Requirements

The job board helps companies find the right employees and job seekers get the position they want. The interface must be web based, and the server needs to be hosted in the cloud, and accessible from anywhere with Internet connection.

Users and Authentication

There are two types of users, company and job seeker.

1. Company can post jobs.
2. Job seeker can search for jobs and submit applications.
3. Both job seekers and companies must use a valid email to sign up.
 - a. The system does not allow the same email address to be used to register for a job seeker account if it is already registered as a company account -- and vice versa.
 - b. Your app must send an email to users [for both job seeker and company] with a verification code. The user needs to use that verification code to complete his account registration. A registered user cannot really use features in the system until his account is verified. A confirmation email must be sent to the user after completion of account verification.

Job Seekers

Job seekers must fill in their profile information before applying for a position.

4. The profile contains at least the following information
 - a. First name
 - b. Last name
 - c. Picture **[optional]**
 - d. Self-introduction [optional]

- e. Work experience
- f. Education
- g. Skills

User can edit and update his profile at any time he likes.

5. Job seekers can search for **open** job positions

- a. Search by text the user types in, which can be a job title, a company name, a skill, a mix of them, or basically, any arbitrary piece of text. A word in the text can be matched against any part of a job posting.
- b. Search by filters
 - i. Company name (allow multiple)
 - ii. Location (city names, allow multiple)
 - iii. Salary range (can be a single value, an open range, or a close range)

You can add more filters if you prefer.

- c. Within each filter type, the relationship between the multiple values is OR; e.g., if you specify both Cisco and IBM as the company filter, the it matches a job from Cisco *or* IBM . The different filters and free text work together with AND; i.e., all need to be matched. For example, If you specify both the company filter and the location filter, a matching job must match both the two filters.
- d. The layout of search result is up to you, but you need to support paging in case the number of matching jobs goes beyond one page, which contains up to 10 jobs.

6. Job seeker can mark/unmark any open position as interested in the search result page.

- a. If a position gets filled or cancelled by the company, it would be removed from applicant's interesting list automatically. The app must send an email notification to applicant too.
- b. A user can also taggle an interested job as not interested in the search result page.
- c. There is no limit to the number of positions applicant can add to his interested list.
- d. Your app needs to show a user's interested list, which should be presented similar to the search result page, except that there is no need to support paging in the view for interested jobs. A user needs to be able unmark/mark jobs in the interested job view too.

7. Apply for a position

- a. A user can apply for a selected position in search result page or the interested jobs page.
- b. A user can choose to apply with his profile, or attach a resume. In both cases, the user name and email from the user's profile becomes part of the application.
- c. For any apply operation, the user would receive an email notification with the information of position(s).
- d. You need to provide a view for a user to browse or cancel his applications.

- i. Each application has a status, Pending, Offered, Rejected, OfferAccepted, OfferRejected, or Cancelled. The latter four states are also called terminal state.
 - ii. A user can cancel a pending application and reject an offered application.
 - iii. The user can cancel (or reject) one or more applications (or offers) in one transaction.
 - iv. The company can cancel any application that is not in a terminal state.
 - v. Jobs of all states need to show up in the application view.
 - e. A user cannot have more than 5 pending applications.
 - f. A user cannot apply for the same position again if the previous application is not in a terminal state.
 - g. For any status change of an application, the job seeker receives an email update as well.
8. Interview scheduling [**Bonus Feature**].
- a. If the company has set up the interview time for a position with an applicant, the applicant will receive an email, ideally with a calendar invitation.
 - b. The app also provides an interview list page for the job seeker, which shows all the interviews scheduled.
 - c. A job seeker can decide to accept or refuse any interview invitation. He should be able to do this through his interview list page, or possibly by accepting or rejecting the calendar invitation as well. The company will also receive an email notification with user's decision.

Companies

10. A company needs to provide basic information upon registration
- a. Name
 - b. Website
 - c. Logo image URL
 - d. Address of headquarters
 - e. Description
 - ...
- Company can edit and update its own information at any time.
11. Company can post new positions into the system. A position contains at least the following information
- a. Title
 - b. Description
 - c. Responsibilities
 - d. Office location
 - e. Salary

A company can edit and update its information mentioned above. When a job is updated, all the current applicants (applications in terminal states are not considered) are notified about the change.

12. Your app needs to provide a view for the company to browse all its positions, with a filter to restrict by the status of a position.
 - a. A position can be either Open, Filled, or Cancelled.
 - b. A company must be able to select a position to view its detail page, where the company can further make changes to the position, as well take other actions:
 - i. Change the content of the position.
 - ii. Cancel the position unless it has an application that is in OfferAccepted state, in which case the company cannot cancel this position.
 - iii. Mark a position can be marked as Filled if it gets an offer accepted.
 - iv. When a job is filled or cancelled, all related applications in non terminal state get cancelled
 - v. From the job detail page, the company needs to be able to enter a view to see all applications for this job. For each application, the company can further
 - A. view the profile, or download the resume.
 - B. Reject an application
 - C. Accept an application by giving an offer
13. The app allows a company schedule and manage interviews for an application **[Bonus Feature]**
 - a. The company can set interview time and location. Once an interview is scheduled, the applicant should receive an email notification as mentioned earlier.
 - b. Ideally, interview invitations should be integrated with a Calendar service, e.g., the calendar service by Google or Microsoft.
 - c. The company should be able to track all interviews.
 - i. interview scheduling status [invitation sent, scheduled, invitation rejected] (OK to consider candidates only, not interviewers)
 - d. You may consider interview feedback management, but not required at all.
 - e. When is position is filled or cancelled, all unfinished interviews should be cancelled.

Grouping

This project is group based, with group size up to four people. Once the project plan is submitted, group membership cannot be changed.

Source Code Management

You are recommended to use a Source Control Management (SCM) system to manage your team's source code. This can be a private Bitbucket repository or your local git. During the

grading of the term project, you may be asked to provide commit history or any other document to help evaluate each team member's contribution.

Cheating Policy

Your app must be built by yourself, and cannot be based on the code base of any existing app. If you used any code not written by yourself, it must be clearly documented in your README.TXT file, unless it is part of publicly available libraries. *If your app is already used to serve the requirements of any other class, it will not be accepted by this class.* In the case any form of cheating is confirmed, you will get an F grade for this class.

Deliverables and Grading

The project is worth 25 points in total. The actual *due dates* of the deliverables will be specified in Canvas.

Project Presentation and Demo (5 points)

To be presented in class.

- The presentation should cover introduction, high-level design, and major features with screenshots. Time limit: 3 minutes.
- You must also do a live demo. The guideline for how to do demos is to be added. Time limit: 5 minutes.
- Grading will be based on successfulness of the demo, the content and clarity of the slides, and the delivery of the presentation

The presentation slides must be submitted through Canvas as a PDF file.

Project Report (5 points)

The report needs to cover the following topics.

1. Motivation and introduction of your app
2. High level and component level design
3. Technology choices
4. Description of features with final screenshots
5. Testing plan executed and results
6. Lessons learned and possible future work

You are recommended not to exceed **20** pages, but you will not be penalized just because the report is too long or too short, as long as the level of coverage for the required topics is reasonable and clear. The report must be submitted through Canvas as a PDF file.

Project App (15 regular points + 2 bonus points)

Note: the instruction for submission is still *subject to change*.

1. You must submit all your source code / resource files through Canvas
2. Features correctness, stability, performance, choice of technology and implementation are worth 12 points
3. User interface and user experience are worth 3 points
4. The bonus features are worth 2 points
5. You need to keep your app live for at least a week before we finish grading

6. README.TXT, including
 - a. The names, email IDs, and students IDs of the members
 - b. The URL to access your app
 - c. Any other instruction necessary for the TA to grade the app
 - d. Build instructions