6.170 Software Studio:
Fall 2014
AmazeJobs: Problem Analysis

Catherine Liu, Elliott Marquez, Olga Shestopalova

<u>Description:</u>

AmazeJobs keeps track of users' job applications: their progress, deadlines, and tasks associated with each application. It strives to alleviate the users' burden of remembering all the jobs they applied to, which of them responded, how long ago, and tasks with deadlines that must not be forgotten. When the user inputs the data from the different applications, AmazeJobs will aggregate it, and display it in an organized and user friendly interface.

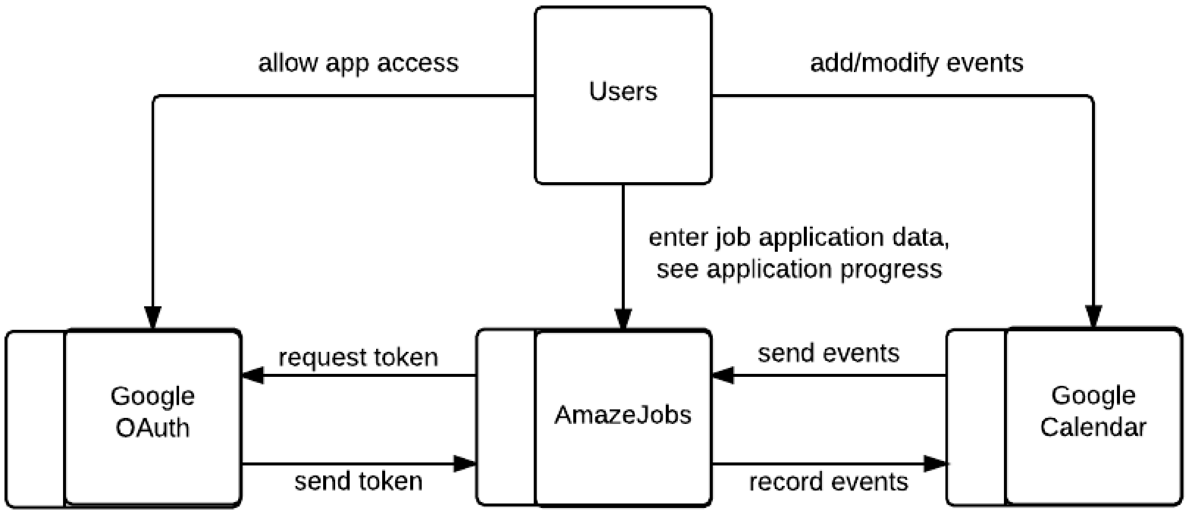
A key user group we are targeting are students around career fair season, where each student will speak to many companies, but frequently will forget to apply on time, or follow-up. We interface with the most popular calendar app - Google Calendar, to propagate events and deadlines throughout all devices, and to help schedule interview times around other activities.

AmazeJobs has phases for each job application: Applying, Interviewing, Offered, Terminated; only one can be active at a time, and the completion of one will give rise to another (unless it is a terminal phase such as Offered or Terminated). Terminated can be entered after any of the other phases. Tasks are tied to a specific phase.

<u>Purpose:</u>

- **Organize the process of searching for a job, especially if you apply to a large number of places.** In particular, AmazeJobs will gather the status of each job application in a single, easy-to-use interface to make it easier to determine which applications need further work, and what stage of applying you are in with each application. If an application needs further work, a user can assign a task to it.
- **Keep track of and remind the user of important events and deadlines.** For example, we would like to help the user schedule interview times/dates and remember when applications have due dates. By integrating with Google Calendar, AmazeJobs will make it easier for people to schedule times for interviews between other activities.
- **Have an aggregate task list of things to do.** AmazeJobs will gather all the assigned tasks from multiple job applications and organize them so that a user can view them all together and re-order the tasks by priority.

Context Diagram:



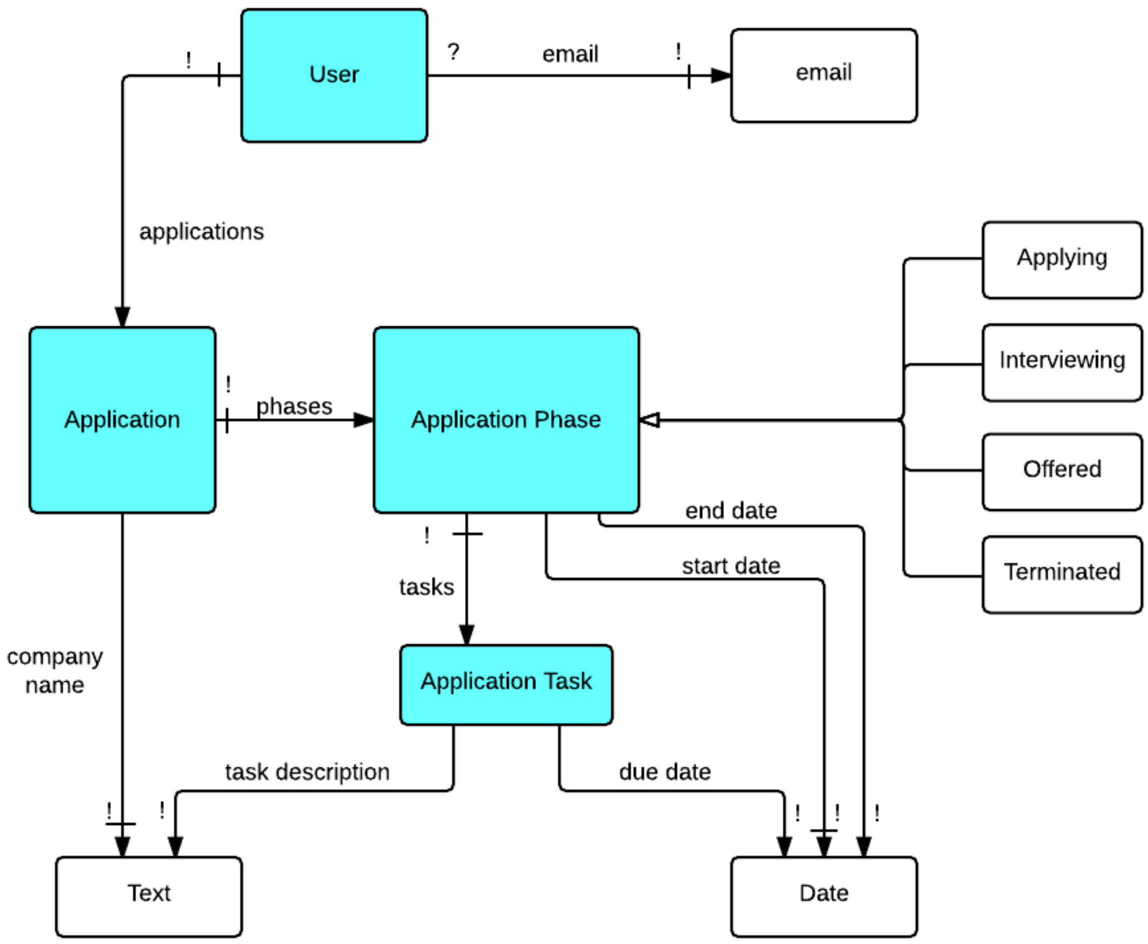
- Sign In/Authentication with Google
  - User approves app login through Google, login completes
  - Potential problem: user doesn't have Google account
- Google calendar integration for tasks/deadlines
  - User adds tasks with deadlines, app inserts this as event in calendar
  - Potential problem: user can externally modify dates

Concept:

- Application: Motivated by first purpose. An application to a job - must have a company name associated with it. Can contain a link to online applications that must be submitted, specific job position and/or job description. Each application is in one of four phases (Applying, Interviewing, Offered, Terminated)
  - Application Phase – must have a start date, can have an end date, and tasks associated with it. It doesn't necessarily need an end date if it is the current phase. Only needs an end date when you finish a phase.
    - Applying – When you first begin an application to a company, either via an online form or resume drop or any other method.
    - Interviewing – When a company offers you an interview, your phase of Applying ends and Interviewing begins
    - Offered – When the company give you a job offer
    - Terminated – Can occur after any stage, either via voluntary withdrawal or via company rejection. Has start date = end date.
- Calendar: Motivated by second purpose
  - A way to view and track dates/times of events/deadlines
  - Reminds user of events/deadlines
  - Aid in planning: user can see their other activities

    - See things in perspective: how close together or far apart deadlines are
- Task list: Motivated by third purpose. Has priority order, drag and drop to reorganize
    - Collects all job application data in one place
    - Task list of things to do

Data model:



- **User**: represents a user of AmazeJobs
    - email: email used to authenticate users
    - applications: a list of current and old Applications
- **Application**: represents a single job application made by a user
    - company name: a non-empty string
    - phases: a list of current and past (not future) phases of the job application
- **Application Phase**: represents a phase in the application (currently there are four: Applying, Interviewing, Offered, Terminated), only one can be active at a time, and the completion of one will give rise to another (unless it is a terminal phase such as Offered or Terminated. Terminated can be started after any of the other phases.
    - start date: date user started the phase, immutable

    - end date: date user ended the phase, used for history purposes
    - tasks: list of Application Tasks associated with the phase
- **Application Task**: represents a task that must be completed
    - task description: non-empty text description of the task
    - due date: date the task should be completed by

<u>Design Challenges:</u>

- We were unsure of how to represent progress in each job application.
    - Options:
        - We could have some string indicator of the current phase
        - We could have predefined phases.
    - Decision:
        - We opted for the latter because each job application went through roughly the same phases and this was more elegant than a string encoding. We also made it easy to add another phase, creating a generic phase set that contained specific phases. Also, having predefined phases makes it easier to display progress consistently across different job applications.
- We were unsure of how to visually represent tasks and events.
    - Options:
        - We could have a list of dates
        - We could have a calendar.
    - Decision:
        - We decided on both: we will have a task list that can be reordered by the user (so that the user can order things by priority or how long the task would take) as well as a calendar, to aid in planning (user can see their other activities) and seeing things in perspective (how close together or far apart their deadlines are).
- We were unsure of how to keep track of users.
    - Options:
        - We could implement some username/password design
        - We can have users login with an outside service (Google, Facebook, LinkedIn, etc).
        - We can have a certificate-based authentication system
    - Decision:
        - We opted for the using an outside service (Google) primarily because we planned to integrate with Google Calendar, and for that we would need the user to allow us access to their Google account anyhow. Also, this kind of login is easier for the user and doesn't require remembering usernames and passwords. This also allows us the possibility to integrate with other Google Apps if we deem it necessary. Additionally, we did not want a certificate-based authentication system because certificates work differently across operating systems.
- We were unsure where tasks belonged in our design.

    - Options:
        - We could have associated them with the job application, have them be their own entity tied to each user.
        - We can have them be associated with a particular phase.
    - Decision:
        - We chose the last option because tasks should be tied to a specific phase, and would not stretch across phases, and certainly not across applications, so for reduction of scope, we placed tasks under their respective phase.
- How to integrate a calendar with an application
    - Options:
        - Use Google accounts and Google Calendar
        - Create our own authorization and calendar system
    - Decision:
        - If we used Google accounts and somebody didn't have a Google account, they would be unable to use our service. But creating our own authorization and calendar system just to provide service to the smaller fraction of potential users without Google accounts seemed excessive, so we will be using Google's APIs and services to implement calendars.
- Whether to associate applications with a company or companies with applications
    - Options:
        - Associate applications with companies
        - Associate companies with applications
    - Decision:
        - Associate companies for applications because we do not think that it will be so common to apply to a company for multiple positions, and if a user would want to look up old applications for a company, this data model will not cause too much of a backend mess to look up.