Design Document

Identification

Project Name: Coach
Team Leader: Samvit Jain ‘17 (samvitj@)
Team Members: Henry Shangguan ‘18 (hys@), Joshua Shin ‘18 (jdshin@)

Section 1: Overview

Coach is a peer-to-peer marketplace and video platform for mock interviews, geared toward college students in the areas of computer science and quantitative finance. Our service gives students seeking highly-competitive internships access to high-quality mock interview sessions conducted by more experienced students who have recently made the exact same journey.

Our site allows experienced upperclassmen to list their skills and industry experience, and connect with students seeking interview practice. Interviewees search for interviewers based on industry, experience, and/or university, and schedule mock interview sessions through the site. The mock interviews are conducted through Coach's video-chat platform, with a virtual whiteboard or coding window provided as needed. Rates are set by the interviewer, and all scheduling and billing is handled by the site.

Section 2: Requirements and Target Audiences

Our target audience consists of college students at top-tier schools in the United States and Canada looking for internships in software engineering and quantitative finance. Job interviews in these fields require students to think quickly under pressure, develop creative solutions, and implement their ideas cleanly and accurately on the spot. These conditions are not easily replicated by a textbook or online guide. Our site provides interviewees with the opportunity for 1-on-1 practice in a realistic environment, which can make the difference between landing a "dream job" early on, and months of scrambling for an opportunity. As of now, available resources exist mostly in the form of books, online forums, and impersonal self-prep tools (e.g. for software, Cracking The Coding Interview, Stack Overflow, HackerRank exercises). An effective service that bridges the gap between self-guided learning and real industry experience could thus fill a valuable niche.

For thousands of college students, the desire to land a selective summer internship is incredibly high. Jobs at the top tech companies, trading firms, and hedge funds (e.g. Google, Facebook, Dropbox, Palantir, D.E. Shaw, Jane Street, Two Sigma) carry immense brand value, immerse students in meaningful projects in highly productive environments, and, of course, offer salaries that can easily exceed $20,000 for summer interns, and $100,000 for full-time employees. The skills needed to secure these opportunities, however, can be learned; consequently, for prospective candidates, there is a high perceived and real return to interview preparation.

For interviewers, our service enable strong students to earn money on a part-time, low-commitment basis from the comfort of their dorm rooms. Mock interviews are likely to command hourly rates that compete with tutoring, without requiring students to travel off campus or join an organization. Finally, coaching underclassmen in an area of personal passion (e.g. probability theory, programming languages) is likely to be personally and intellectually rewarding for interviewers.

There are a few existing services that facilitate mock interviews between strangers; however, we've identified missing components or significant flaws in each of these tools, which we believe explain their lack of traction.

- *Pramp* (https://www.pramp.com/) matches people with similar backgrounds to take turns interviewing each other, but as a free service that pairs participants with overlapping objectives (i.e. improving their own interview performance), it is unlikely to attract the quality of interviewers that separately sourcing and compensating strong, experienced students would.
- *CareerCup Evisors* (https://careercup.evisors.com/search?functions=9) is closer to our conception of a mock interview marketplace. It allows users to search for interviewers by industry, most of whom are current employees at top companies, and choose a suitable match. However, the mock interview itself is arranged to occur elsewhere. Moreover, the typical price for an interview on the site is hundreds of dollars, and strangely, there are only 6 interviewers listed in the software engineering space. We believe this is due to a fundamental supply problem in the market this service targets: full-time employees at Google, Facebook, and the like can only interview at off-hours, will likely demand much higher rates than college students, assuming they are even interested in the extra income, and could possibly see a conflict of interest in helping candidates get into the companies they work for, as a side job.

Section 3: Functionality

There are two general use cases for Coach, that of the interviewer and that of the interviewee, as illustrated below:

User A (Interviewee): Henry has his first interview of the season with Google in a week. He's gone through "Cracking the Coding Interview" and refreshed himself on basic algorithms and data structures, but is nervous, and feels he could benefit from solving a real problem in a timed setting, where he can get guidance but also experience what it's like to be judged on his work. Henry logs in to Coach, and sees that several upperclassmen acquaintances are offering mock interviews on the site, including one he knows from his Computer Networks class, who answers all the questions in lecture. Henry sends him an interview request for Tuesday 3pm for 1 hour, and adds an optional message stating that he'd like to focus on binary trees and recursion...

User B (Interviewer): Vlad is a senior in the Computer Science department and has previously interned at Cloudera, Google, and Databricks. Vlad feels that he has "cracked" the coding interview and is confident that he can coach others to do well. Vlad creates an account on our site as an Interviewer, and makes a profile listing his qualifications and the hourly rate he'd like to charge. He also fills out a calendar with his availability for the next few weeks.

User Interaction: Vlad accepts Henry's interview request, and both are sent a confirmation notice. At the specified time, both users log into the site and are connected on our built-in video platform. After they have introduced themselves, they officially begin the timed session, using the built-in virtual whiteboard and CoderPad as needed. At the end of the interview, Vlad goes over his assessment, and gives Henry some last-minute tips for his upcoming phone screen. Henry is automatically billed for the time, and Vlad has the option to pass on his written feedback to Henry, which will be sent along with the code written during the session. Henry must leave a review for Vlad, which will affect Vlad's aggregated rating on the site and will be viewable by other users.

Mock-up of Profiles page:

[Coach] [Search Parameters]	[Search for Interviewer] [Sign In] Available Interviewers: ● Vladimir Feinberg [Request] ○ Interned at Cloudera, Google, Databricks ● Bradley Snider [Request] ○ Princeton '17; Trader at Jane Street ● Blair Wang [Request] ○ Intern, Incoming Analyst at Citadel LLC
[Help | Contact]	

Section 4: Design

We plan to build our frontend with Angular.js, a client-side Javascript web framework, and our backend with Express.js, a server-side Node.js framework, and MongoDB, a NoSQL document-store database. We will deploy our service on Heroku, a hosting platform that allows apps to be managed and scaled without manual configuration of the underlying infrastructure.

We will follow the Model-View-Controller (MVC) paradigm in structuring our application. We list the views, models, and controllers we intend on using, but may modify this list:

Our core views will include:

- Landing page with service summary/features and login/signup options
- Interviewer profile creation, with calendar integration
- Interviewer listings table

- Interview screen, with embedded video chat and code editor views
- End-of-session page for leaving feedback/reviews

Our models will include:

- Read/write logic for a collection (table) with interviewer data; fields will include username, password, name, school, headline, bio, aggregated rating, reviews, availability, hourly rate, Skype username, bank account info, and interview history
- Read/write logic for a collection with interviewee data; fields will include username, password, name, school, Skype username, credit card information, and interview history
- Read/write logic for a collection with interview data; fields will include interviewer, interviewee, start date/time, planned duration, hourly rate, is confirmed, actual duration, content, rating, review

Our controllers will include:

- A listings controller that displays a list of interviewers ranked by weighted sum of net rating, proximity to interviewee (e.g. same school), and other factors
- A profile creation controller that creates a new interviewer (or interviewee) object, populates its fields, and saves it to the interviewer table
- A review composition controller that updates the corresponding interviewer object (aggregated rating and reviews fields)
- A login/signup controller that accesses (i.e. for authentication) and stores (i.e. for account creation) interviewer/interviewee account data
- A timing/billing controller that times the current interview session, and at the end of the interviews, charges the interviewee/credits the interviewer by the appropriate amount

We will use the following integrations:

- Code editing: CoderPad or Collabedit API
- Video conferencing: Skype or Talky API
- Payment: Stripe or Braintree API (if time permits)

We will use git for version control, and a variant of gitflow for our branching model. We will maintain separate dev, staging, and production environments (through a scriptable infrastructure), and also hope to incorporate automated testing.

Section 5: Timeline

Minimal plumbing:

- Node.js web server, backed by a MongoDB database, running on Heroku
- Webpage with embedded video-chat and embedded collaborative editor

Minimum feature set

- Prototype
    - Webpage with embedded video chat tool and code editor

- Alpha
    - Profile listings
    - Google Calendar availability indication/selection
    - Interviewer profile creation
    - Interviewee account creation
    - Screen-size/device support
- Beta
    - End-of-session reviews
    - Session summary notes transmission/download
    - Landing page

Week 0 (Spring Break, ending 03/19):
- Test integrations
    - Build static webpage with embedded video-chat (Skype, Talky, etc.) and embedded collaborative editor (CoderPad, Collabedit)
- Setup frameworks and hosting
    - Install Angular.js
    - Configure Express.js with MongoDB to run on Heroku
- Basic storyboarding
- Create static **Project Website** ***(deadline 3/21)***

Week 1 (ending 03/26):
- More detailed mockups
- Define database schema
- Basic webpage with embedded video-chat and code editor
- Test: two people access domain and communicate through video and code editor
    - This will be our ***Prototype (deadline 3/28)***

Week 2 (ending 04/02):
- Basic, interactive profile listings with information pulled from database
    - Setup REST interface between Angular.js and Express.js frameworks
- Setup Google Calendar integration
- Continue developing mock interview interface

Week 3 (ending 04/09):
- Allow for Interviewee to select a time slot in Interviewer's Google Calendar
- Build Interviewer profile creation and Interviewee account creation interface
- Continue developing mock interview interface

Week 4 (ending 04/16):
- Work on backlog of features not implemented yet
- Test on different screen sizes (resizing, mobile, etc.)
- Test with 5 real users and get feedback

- Test end-to-end functionality for **Alpha** ***(deadline 4/18)***

Week 5 (ending 04/23):

- Allow Interviewer to review Interviewee at end of session, and vice versa
- Allow for Interviewer to send Interviewee notes/summary/code from the interview
- Develop landing page
- Test with 10 real users and get feedback
- Test end-to-end functionality for ***Beta (deadline 4/25)***

Week 6 (ending 04/30):

- Work on backlog
- Trim partially or poorly implemented features
- Test with 15-20 real users and get feedback
- Test end-to-end functionality for ***Demo (5/2-4)***

Week 7:

- Prepare for ***Demo (5/2-4)***
- Write Product Guide and Final Report

Section 6: Risks and Outcomes

Our largest risks arise from our integrations. We plan to test that all of our integrations will work as intended during Week 1. If our intended choices (Skype, CoderPad) don't work, we'll try our backups (Talky, Collabedit). If these don't work either we will have to quickly find other alternatives or modify our site's feature set.

While our group members have some experience in Javascript (e.g. in Meteor.js) and with some of the technologies we'll be using (MongoDB, Heroku), we'll be using a stack which is relatively new to us (MEAN, with its JS/Node.js backend), so it might take us longer than expected to get it set up and running. We are planning to implement our core features first, so if we fall behind schedule we should still have a complete and functional product, just minus a few auxiliary features.

We don't have any dependencies on proprietary data or closed-source tools.