Easy-Hiring

Team Member: Yongrui Lin, Xiaojing Ji

Due Date: September 22, 2017

Motivation

During the recruiting season, job hunters are busy on talking to companies and handing over their resumes to recruiters. On the career fair, they wait in line for hours with their resume hardcopies in order to reach out companies, but they may even have no clue whether themselves are a good fit for the positions. At the same time, recruiters are exhausted by reading over thousands of resumes to find best candidates to move forward. There's a joke saying that both applicants and recruiters need to mentally prepare for handling the massive information flood during the career fair. We realize the job fair services may need a more dynamic solution so that applicants and recruiters can quickly filter out irrelevant options. Here we attempt to develop a mobile application which assists job hunters to search for matched positions, as well as helping companies select qualified candidates based on resume contents and position required skills.

Related Work and Tools

- Many job searching platforms have successfully integrated company information and user profiles. For example, as an employment-related search engine, Indeed[1] lists jobs that meet searching conditions. The professional networks like LinkedIn can compare profile contents to some posted positions and acknowledge users with recommendations if there exists a match. Glassdoor collects anonymous company reviews from employees and offers access to online job applications[2]. Job-related services provided by those websites significantly improve the efficiency of job hunting experience as users can utilize integrated data to filter choices and find best matches.

- **React Native**: we decide to use React Native as the platform for this mobile application. React Native was originally released in 2015 and has started gaining popularity ever since then[3]. It shares the same design as React and supports UI building blocks for iOS and Android deployment. JavaScript package manager npm keeps many public API packages to accommodate React Native, which simplifies implementation of certain features on this application.

- **Firebase Real-time Database**: a NoSQL cloud database sponsored by Google[4]. It synchronizes data in realtime and stores data in JSON format. The data synchronization feature ensures instantaneous update in case clients have made any changes.

Features of Proposed Work

The major goal of this mobile application is creating a platform for career fair events so that participants can store and access the electronic copy of resumes, as well as filtering out unfitting positions or candidates. Instead of bringing printed resumes, people simply grant access to their resumes through the application. The system analyzes contents of uploaded documents and extracts keywords from resumes. With the job requirements posted by recruiters and knowledge of applicant's' skills, it is capable of determining whether a candidate is qualified. Such refinement benefits both job hunters and recruiters. For instance, after receiving tons of application submissions, instead of manually going through each one of them, recruiters can save some time by entering keywords to filter out unqualified items and only keep desired results on their devices for future consideration. Similarly, applicants can check if any companies look for certain technical skills(e.g., C++ programming experience) which perfectly match their backgrounds. To meet those requirements, we will implement the following features.

1. Companies that attend career fairs can create an account and post job descriptions.
2. Each job applicant can upload resume in PDF to the personal account.
3. The application automatically generates QR code for the corresponding PDF file, then uploads the resume to NoSQL database. The purpose of QR code is replacing the traditional hardcopy resume transmission. When talking to a company, people can share resumes by letting the recruiter scan QR code.
4. After the uploading succeeds, it reads contents to extract keywords and updates the database accordingly.
5. Recruiters give keywords and select matched candidates. They are able to download selected resumes to the local file system.

System Architecture

The system is composed of a mobile frontend and a centered backend. The backend connects to a cloud-hosted database storing user profiles, PDF documents, and other relevant data. We select React Native as the platform which will integrate with Firebase database. The npm dependency management initializes the connection between React Native and the database. At this moment, we expect this application to be feasible for iOS and Android deployment. Here is the system architecture overview:

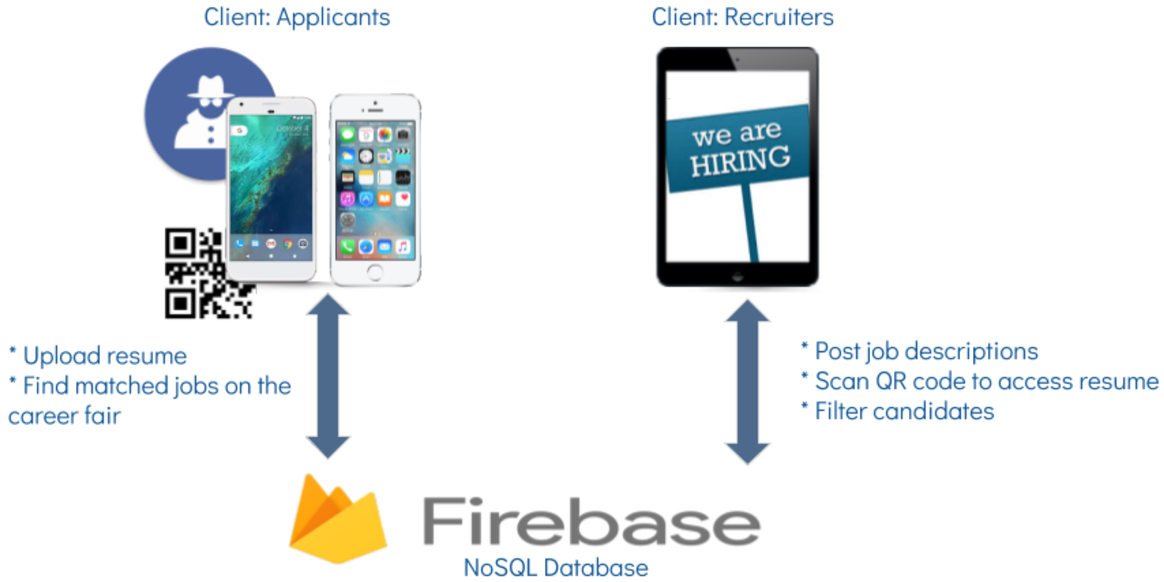

Image Sources: "Apple IPhone PNG Images Transparent Free Download." *PNG Mart*, www.pngmart.com/image/tag/apple-iphone.
Savov, Vlad. "Pixel 'Phone by Google' Announced." *The Verge*, The Verge, 4 Oct. 2016,
"Download Tablet Free PNG Photo Images and Clipart." *FreePNGImg*, 15 June 2015, www.freepngimg.com/electronics/tablet/page/4.
"Seasonal Administrative Assistant Position Available." *Ryan Jorgenson & Limoli, P.S.*, 10 Nov. 2015, rjl-cpa.com/seasonal-administrative-assistant-position-available/.

- Mobile User(Client): career fair participants can register the account and identify themselves as applicants or recruiters. We consider using React Native Elements UI Toolkit to enrich user experience on UI interactions.
  - On applicant's view:
    - Load PDF file from local file system to the profile. API provided by npm package libraries gives React Native simple file system access on iOS and Android.
    - Generate QR code for the profile. Send resume to the backend for storage and content processing.
    - Keywords search to find available positions posted on the career fair. This searching request sends a query to the backend.
  - On recruiter's view:
    - Create posts with detailed information.
    - Scan QR code to download resume from the database. There are npm packages supporting camera access and QR code scan.
    - Filter submitted applications by entering keywords.

- Backend: The backend handles requests received from mobile users and supplies returned results to the frontend. It attempts to read PDF document via an npm module which converts PDF from binary to JSON. For keyword extraction, one trivial solution is letting the system query JSON data to pick up items containing matched entries. A supplementary pre-trained machine learning model may be inserted into the module to improve the content analysis performance. One of the team members has implemented a language comprehension model using LSTM(Long-Short Term Memory) neural network. We will decide whether to embed this additional machine reading model based on the work schedule.

Schedule and Work Plan

Date	Milestone
September 24th - September 30th	Explore React Native and Firebase
October 1st - October 7th	Connect Firebase with React Native. Implement user registration and user login
October 8th - October 21st	*Enable loading PDF files to the system. *Create QR code *Upload QR code with its corresponding PDF file to firebase *Scan QR code
October 22nd - October 28th	*Implement job posting feature *Design schemas for data organization *Allow keywords search for posted jobs

October 29th - November 4th	Implement content analysis on PDF documents and job descriptions, update database accordingly
November 5th - November 11th	Create search filter on the client and display matched results
November 12th - November 25th	Allow document downloading. Refine the project, may involve machine reading model
November 26th - December 2nd	Prepare for the final presentation

Deliverables

- Source code of the project.
- Deployable mobile application that meets the proposed major requirements.
- Project presentation and demonstration videos.
- Final project report.

References

1. Job Search | Indeed." Jobs, www.indeed.com/.
2. "Glassdoor Job Search | Find the Job That Fits Your Life." Glassdoor, www.glassdoor.com/index.htm.
3. "A Framework for Building Native Apps Using React." React Native, Facebook, facebook.github.io/react-native/.
4. "Firebase Realtime Database | Firebase." Google, Google, firebase.google.com/docs/database/.