Intern-Master

Considered your internship application done :)

Team 17 - CS 30700: Design Document

Ji Ma, Nathaniel Young, Ji Soo Cha, Xinping Zhang

Purpose

People who are actively looking for jobs have undoubtedly suffered the poorly designed repetitive experience for applying to different companies. I genuinely believe that prospective interns and job candidates should not need to go through this complicated and repetitive cycle. There is a need for automated service to apply for internships. Because the internship applications forms are highly repetitive: each company has its unique application system, and the candidates must use that specific system to apply for a position. Most of the things on these forms are the same: first name, last name, school, graduation date, skills, projects, experiences, ethnicity, disabilities, gender, etc. Although sometimes the auto-completion in Google Chrome or Apple Safari will do some of the magic for you, the result is so minimal and ineffective. Many students find themselves copying and pasting when filling these forms. Our purpose is to create an automated system with a smart 'brain' that would find and apply to intern positions would save students from physical labor, mental stress, and invaluable time.

Functional requirements

1. Smooth sign in experience
    a. As a developer, I would like to create the ability to automatically import a prospective job applicant's basic information and skills by using their LinkedIn account information so that prospective job applicants can have a smoother experience using the Intern Master platform.
    b. As a prospective job applicant, I would like the ability to use "Sign-in with Google" to make the setting up my account with Intern Master a breeze.
2. Job application
    a. As a prospective job applicant, I would like to the ability to login to the website so that I can have all of my important information stored for me which would allow there to be the one click apply feature.
    b. As a prospective job applicant, I would like to automatically apply to a job in one click so that I do not have to go through the same repetitive text and drop down boxes for each job I want to apply to.
    c. As a prospective job applicant, I would like the ability to see what date the job was published so that I can know how recently it was posted, and if I am one of the first applicants.
3. Notifications
    a. As a prospective job applicant, I would like would like the ability to be notified by email or text message when a new relevant job is posted that where the job requirements match my skills.
4. Skills
    a. As a prospective job applicant, I would like to add my skills so that I can have jobs automatically matched for me so that job opportunities can come find me instead of me trying to find job opportunities.

b. As a prospective job applicant, I would like to polish my skills with the best recommendation so that I could have a best match to the target company.

c. As a prospective job applicant, I would like the ability to share my job profile which will feature a downloadable resume, skills tab, work experience tab, and links to external websites (like LinkedIn, GitHub, etc).

d. As a prospective job applicant, I would like the ability to manually edit my public facing profile so that I can change any errors in the automatically generated profile.

5. Suggestions for Jobs/education/open source work
   a. As a prospective job applicant, I would like to browse other similar jobs so that I can apply to more relevant jobs quicker.
   b. As a prospective job applicant, I would like to ability to find online education sources to help me polish my skills so that my application looks better to interviewers.
   c. As a prospective job applicant, I would like to be suggested which online education sources I should take in order to meet the requirements of job X.
   d. As a prospective job applicant, I would like to see awesome relevant open source project suggestions that I should consider working on so that I can boost my job application and resume.
   e. As a developer, I would like Intern Master to automatically suggest awesome open source projects to work on (in the same way that it automatically suggests relevant jobs to apply to).
6. Blog
   a. As a prospective job applicant, I would like to read success stories of other people who used Intern Master on the Intern Master blog so that I can be encouraged to strive for my goal of getting that job.
   b. As a prospective job applicant, I would like the ability to comment on an Intern Master blog to express my feelings or congratulate somebody else on their job success story.
   c. As a prospective job applicant, I would like the ability to see what date the blog on the Intern Master blog was published so that I can know how recently that particular person had a successful job acceptance.
   d. As an organization, Intern Master, would love to host guests who work in industry on the Intern Master blog to write an article so that our prospective job applicants can have awesome content to read.
   e. As a developer, I would like to put Amazon affiliate links on our website to the best books, computer gadgets, and things to help prospective job applicants study for their next interview, and so that Intern Master can gain a little bit of profit for their services.
7. Forum
   a. As a developer, I would love to create an Intern Master forum (using the same login account information) can ask and answer questions for each other.

    b. As an organization, Intern Master, would love to create a community of passionate people to interact on the Intern Master forums.
    c. As a organization, Intern Master, would love to have a group of volunteering professionals in industry to monitor the Intern Master forums.

8. Website
    a. As a prospective job applicant, I would like to the ability to easily navigate the job suggestion website so that applying to more jobs will be a breeze with good UI.
    b. As a developer, I would like to create an about page about Intern Master was created and the amazing developers behind it.
    c. As a developer, I would like to show our personal profiles so that we could be found easier and can acquire our next interviews and jobs easier.
    d. As a prospective job applicant, I would like the ability to have a toggle button on the website to a night mode (because we all know that you probably are burning the midnight oil, and a darker color screen will be better for your long term health).
    e. As a prospective job applicant, I would like the ability to share awesome jobs that I applied to on Facebook, Twitter, etc. so that I can use my energy on social media while “Intern-Master” does all the hard work of job applying in the background.
9. Other (if we have time features)
    a. As a developer, I would like to add a subscription based service called “Intern Master Pro” for only $19.99 per month which allows prospective job applicants to apply to all jobs that are relevant to them without ever having to do anything.
    b. As a developer, I would like to give “Intern Master Pro” subscribers access to the beta software (the next big things).
    c. As a prospective job applicant, I would like the ability to gain “App-Points” by referring other prospective job applicants to apply to jobs through “Intern-Master.”
    d. As a prospective job applicant, I would like the ability to get early access to special jobs by redeeming “App-Points”.

Non-functional requirements

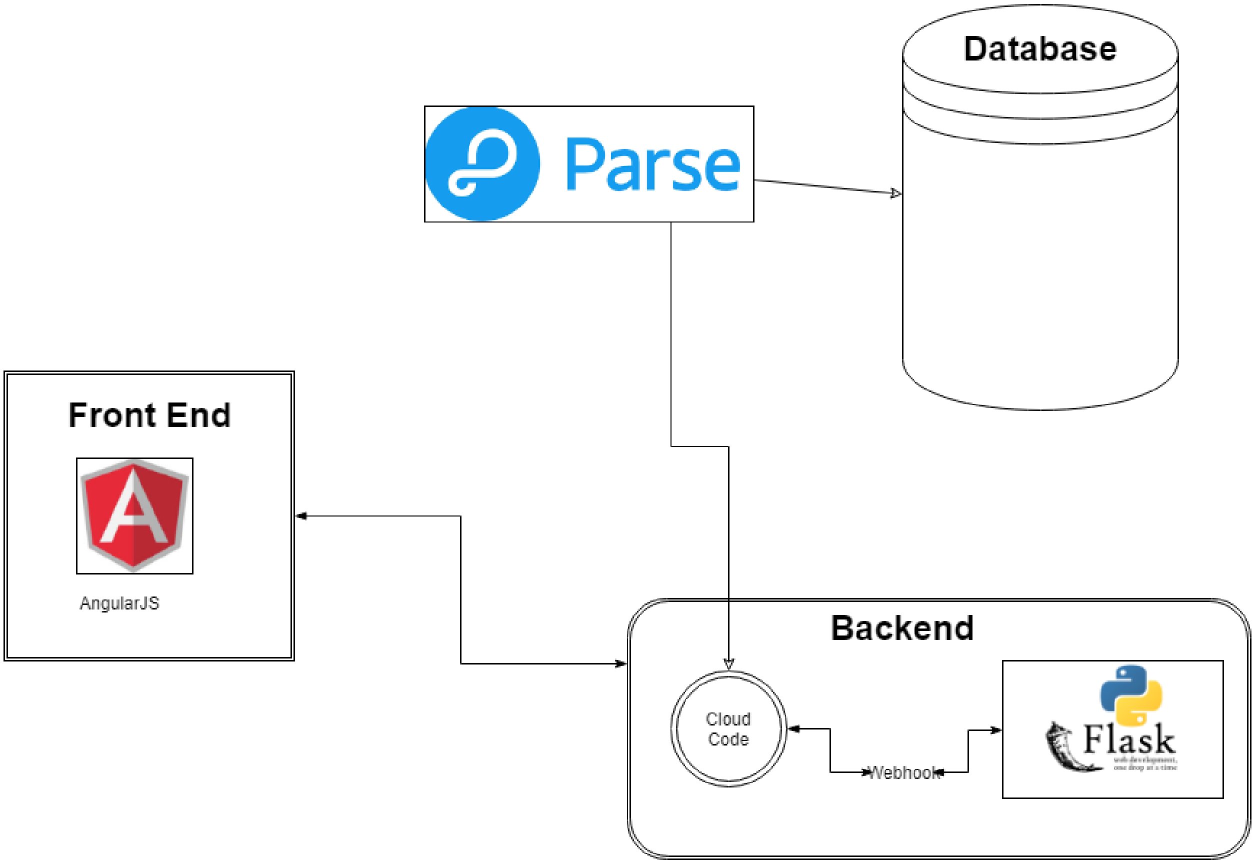
1. Architecture & Infrastructure
    a. Backends will be Python based using Django and Flask framework to design a restful API.
    b. React/AngularJs based front end web app allowing users to filling their information, manage detailed information, skills development suggestions and search for matching positions.
    c. BaaS such as Firebase/Parse to store user information.
    d. Chrome extension to help frontend web app to apply the dom manipulation.
    e. We will use docker container heavily to ease the deployment process.
    f. We will use travis.ci to automate the build and for our source code.
2. Security

    a. Security is essential for Intern-Master, the Intern-Master’s database will contain users’ crucial personal information such as their first name, last name, school, graduation date, skills, projects, experiences, ethnicity, disabilities, gender, etc.
    b. Although the third party BaaS will take care of the security issues for us by using token authentication and other methods, we still looking forward to implementing our services with https to ensure the best-secured service.
3. Usability
    a. The interface should be easy to navigate and simple enough for the average user to understand.
    b. The Intern-Master works on any types of internship application formats and accessible on all the platforms such as Mac, Windows, Android, or iOS.
4. Hosting & Deploying
    a. One single server is enough for the starter phase. We will use Google Cloud/AWS to host frontend + backend at the same time. We will heavility use container services like docker to ease the hosting and deploying headache.

Design Outline

High level overview

Our project will use the client-server model. Here is a diagram of how the high-level overview of



the system works:

Front End
AngularJS for user interface and interactions. AngularJS is a javascript framework that will boost the development process and implemented high level MVC architecture with ease for developer. The front end will be a thin client, all the logical operations and functions will be sent to the server by restful API. Also, with the powerful sdk provided by Backend as a Service platform Parse which wrapped up the raw http communications, we could handle a lot complicated functionalities without even worrying about http requests.

Back End
Cloud code in Parse will work as general backend for handling APIs and database. The runtime environment will be based on Node.js, and the framework inside of the cloud code is Express. Cloud code would be very useful for dealing with functions that is heavily related to data manipulation and simple logical functions that don't need heavy computing power.
For those tasks like crawling and training that can't be handled by the Parse, we are going to "redirect" the request for that particular task to our full fleshed server driven by python Flask framework. Python flask will handle the operations that can't be performed by parse, but the Parse API will be the entry/wrapper to communicate with the individual server by webhook.
Flask <-> Webhook <-> Parse Cloud Code

Database
The database will be mainly use parse, a backend as a service platform that enhanced the utility of NoSQL that supports a lot of convenience features like user management, push notification, and of course object storage. The data object will be stored as json object, so we could store more complicated class instance inside of our database.

Design Issues

Functional Issues:

1. How do users sign up or sign in to our server?
    - **Allow to login using Google account.**
    - Create a username and password that is unique to our server.

   Decision: We plan to implement the ability to use "Sign-in with Google" to make account for Intern Master. We want our Intern Master to be accessible so that there is no extra work to create new user account just for Inter Master.

2. What should users write to their personal information for their accounts?
    - Fill in the forms that are provided by Intern Master.
    - **Import existing account informations from LinkedIn account.**

   Decision: We plan to create ability to automatically import a users' personal informations and users' skills by using their LinkedIn account informations. Therefore, the users can have a smoother experience as they start using the Intern Master platform.

3. How can users find job opportunities from Intern Master?
    - Search it from the job list provided by Intern Master.
    - **Intern Master automatically match jobs for users.**

   Decision: Intern Master automatically match job opportunities for users based on information provided in users' "my skills" information section instead of users searching to find their job opportunities. Additionally, Intern Master automatically suggest online education sources that are needed to meet the requirements of suggested job opportunities if needed.

4. How should users use Intern Master program to complete their job applications?
    - Fill out every job applications for that users are applying.
    - **Intern Master will automatically fill out the job application for the user.**

    Decision: Intern Master will automatically fill out the job application for the user.

Non-Functional Issues:

1. What front end technology to use?
    - **Angular**
    - React
    - Vue

    Decision: AngularJs. AngularJS is a community driven web frameworks developed by Google that eased a lot frontend problems. The reason we choose this is because we have the experience with it before, so the learning curve would be minimal.

2. What back end technology to use?
    - Java Spring
    - **Python Flask**
    - Python Django
    - Nodejs

    Decision: Python Flask. Although we will use Backend as a Service to handle the simple logic services like login and etc, some heavy lifting job can't be done by BaaS. Plus, our project related to scrawling and machine learning, which is typically a good fit to use Python. Flask will wrap up the logic as RESTful API to communicate with our client.

3. What database system to use?
    - SQL
    - Redis
    - MongoDB
    - **Parse / Firebase**

    Decision: Firebase. Initially we considered to go with Parse because of the familiarity of the previous work experience. However, we found that Parse is not providing any public hosted service, which requires some amount of complicated works to get it running. So, in terms of avoiding those weird issues and potential bugs along the way, we chose to go with more matured platform Firebase by Google.

4. Where do we collect company data?
    - Intern.supply

- LinkedIn
- Google search
- Monster
- Indeed
- Startwire

Decision: All of them have numerous job postings that would be beneficial to get data from. Our first target should be intern.supply because it is a nice easy list of links to new job postings.

5. How do we collect user information and operations?
    - Python and Beautiful Soup for webscraping.
    - Scrapy

    Decision: We will choose to use Python and the library Beautiful Soup for webscrapping because we are familiar with it and have used it before. We may incorporate Scrapy since it provides a nice framework to scrap given a root URL, so it can actually look at the application form.

6. What data structure and organization for the user operation and information will be fed to machine learning model and will be used for communication?
    - **Json object**
    - XML

    Decision: Json Object. Come on, this is 21st Century. Json is a standard right now. Plus, it is literally very easy to deal with. Simple but power, that's why we love json.

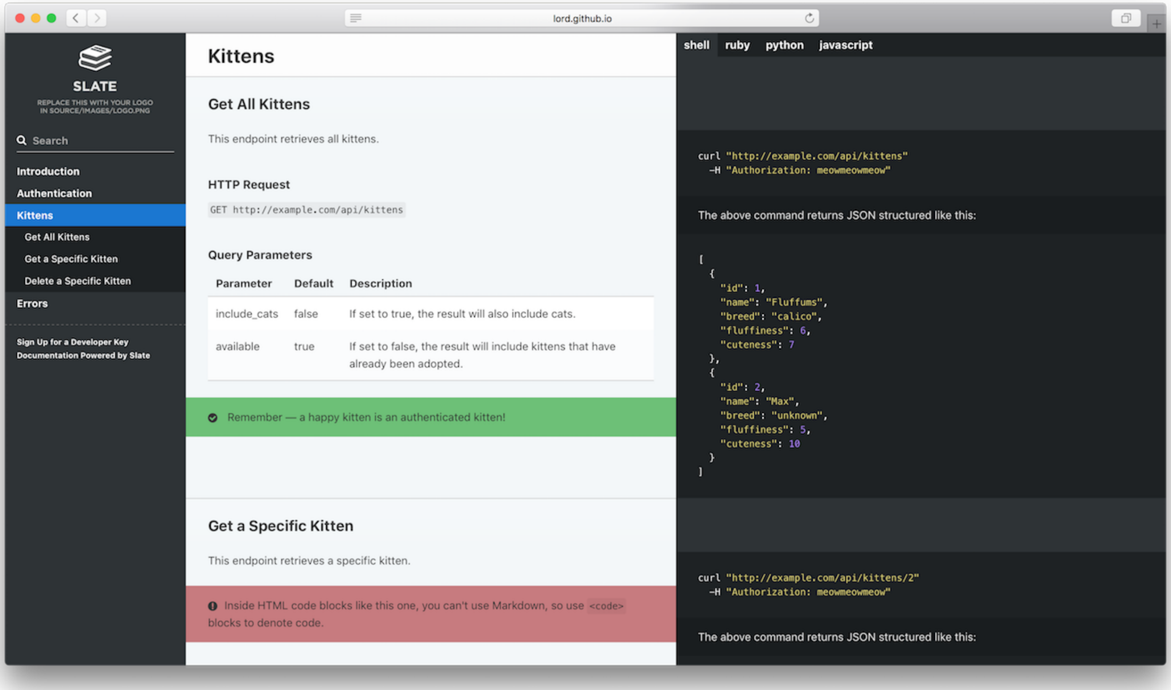
7. What if we can't accomplish machine learning feature, what is the workaround for auto completion?

    Decision: We will use hard coded automation script by using Chromless. As long as we collect feedbacks and get enough amount of data from users, we will start developing more error prone machine learning algorithms.

8. What framework to use for DOM operation and automation?
    - **Chromeless**
    - Nightmare
    - Puppeteer

    Decision: Although we never use those chrome automation library before, we are sure about Chromeless will do the task for our auto filling function because it has the most amount of stars in the github community. Yes, we do trust the quality of the community driven project.

9. Do we need to focus on other internships opportunities besides CS industry?
    - Not for the first sprint as we are trying to get the main concept to work, but after it runs smoothly, we should have any problems adding other types of internship opportunities from other industries.



Kittens

Get All Kittens

This endpoint retrieves all kittens.

HTTP Request

```
GET http://example.com/api/kittens
```

Query Parameters

Parameter	Default	Description
include_cats	false	If set to true, the result will also include cats.
available	true	If set to false, the result will include kittens that have already been adopted.

Remember — a happy kitten is an authenticated kitten!

Get a Specific Kitten

This endpoint retrieves a specific kitten.

Inside HTML code blocks like this one, you can't use Markdown, so use `<code>` blocks to denote code.

```
curl "http://example.com/api/kittens"
  -H "Authorization: meowmeowmeow"
```

The above command returns JSON structured like this:

```
[
  {
    "id": 1,
    "name": "Fluffums",
    "breed": "calico",
    "fluffiness": 6,
    "cuteness": 7
  },
  {
    "id": 2,
    "name": "Max",
    "breed": "unknown",
    "fluffiness": 5,
    "cuteness": 10
  }
]
```

```
curl "http://example.com/api/kittens/2"
  -H "Authorization: meowmeowmeow"
```

The above command returns JSON structured like this:

10. How do we documenting the backend api properly in order to let our developer to work with our service with confidence?

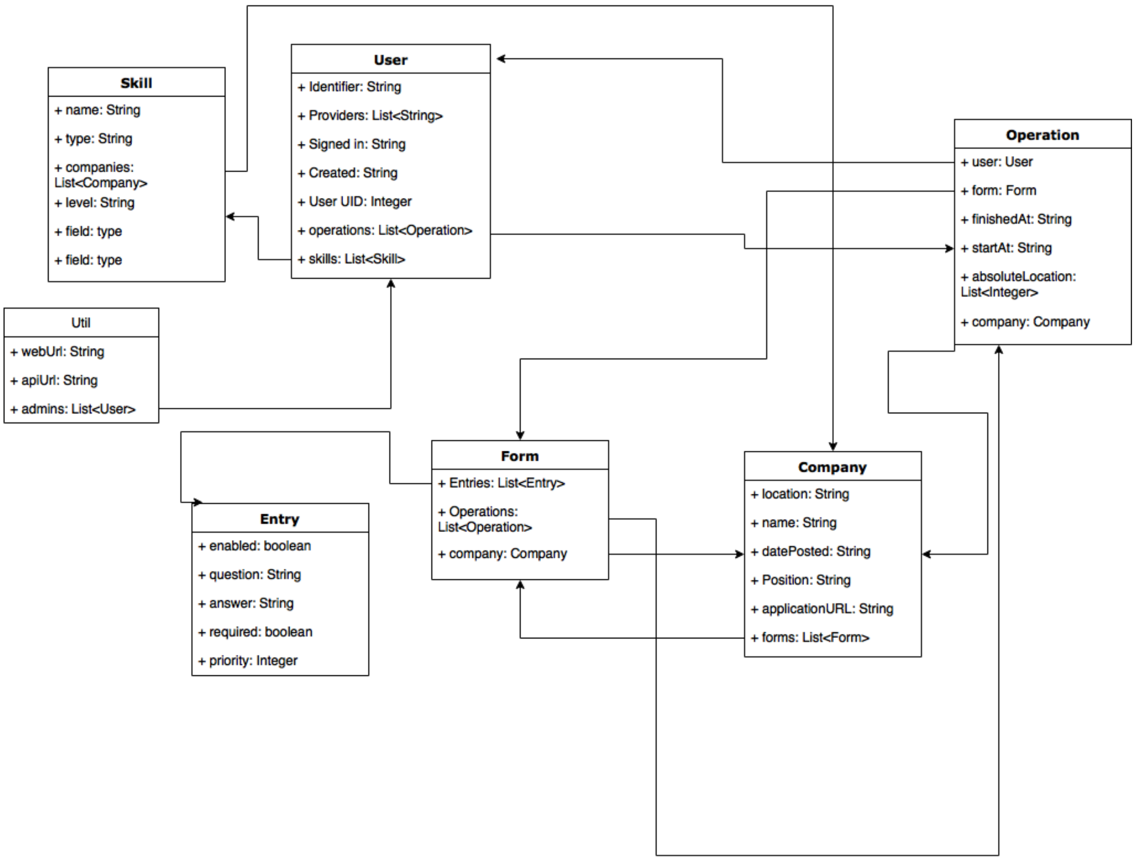
    Decision: We will use slate, a open source API documentation generator for the greater good. It is not fun to write API document, so that’s why we use tools to set us free from boring jobs.

11. How do we automate our test, build, and test our product?

    - **Travis CI**
    - **Jenkins**

    Decision: Both. Jenkins is the leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project automation. Travis CI is a tool for testing and deploy with Confidence. Easily sync your GitHub projects with Travis CI and you'll be testing your code in minutes!

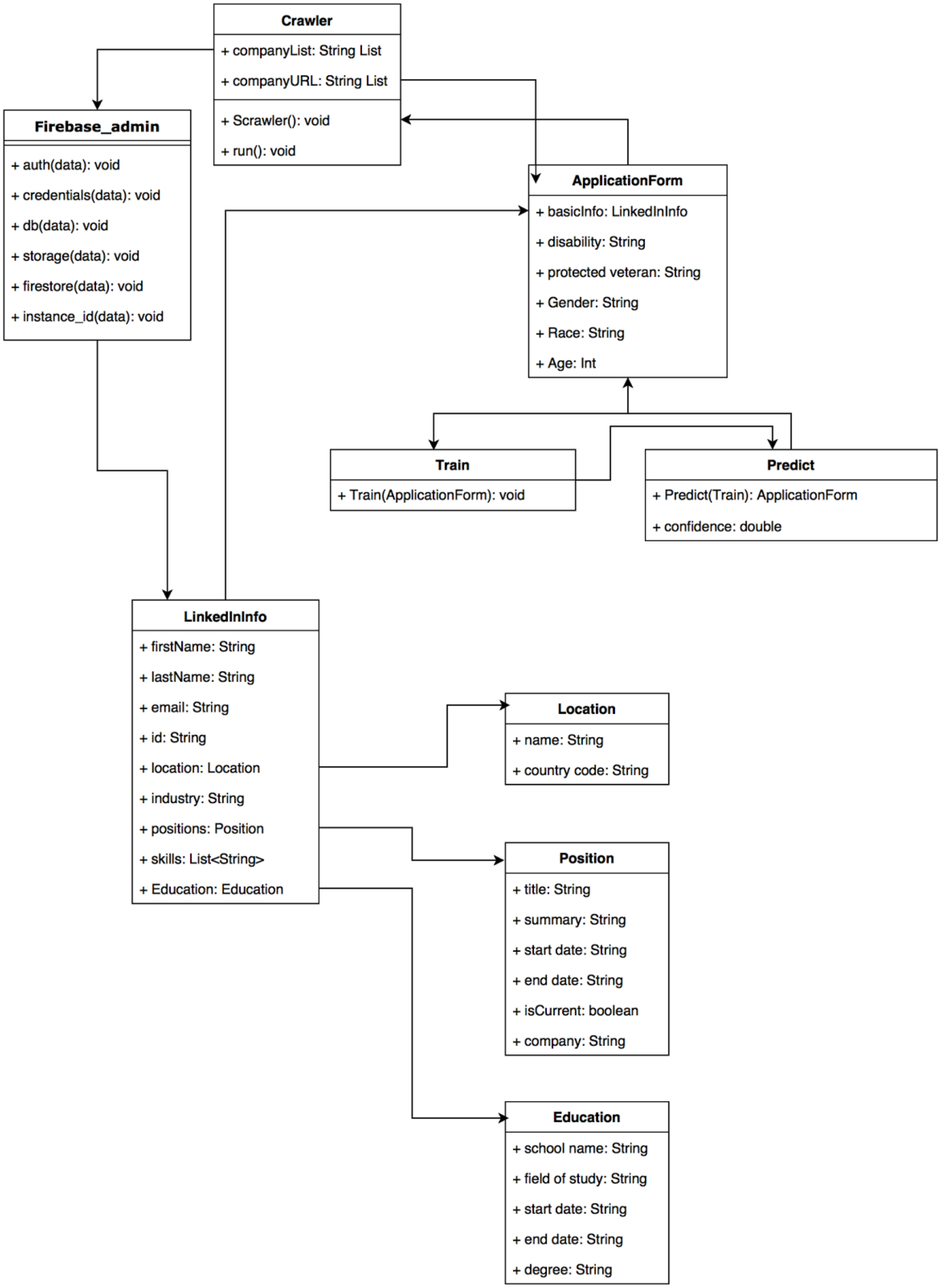
Design Details



Database Design

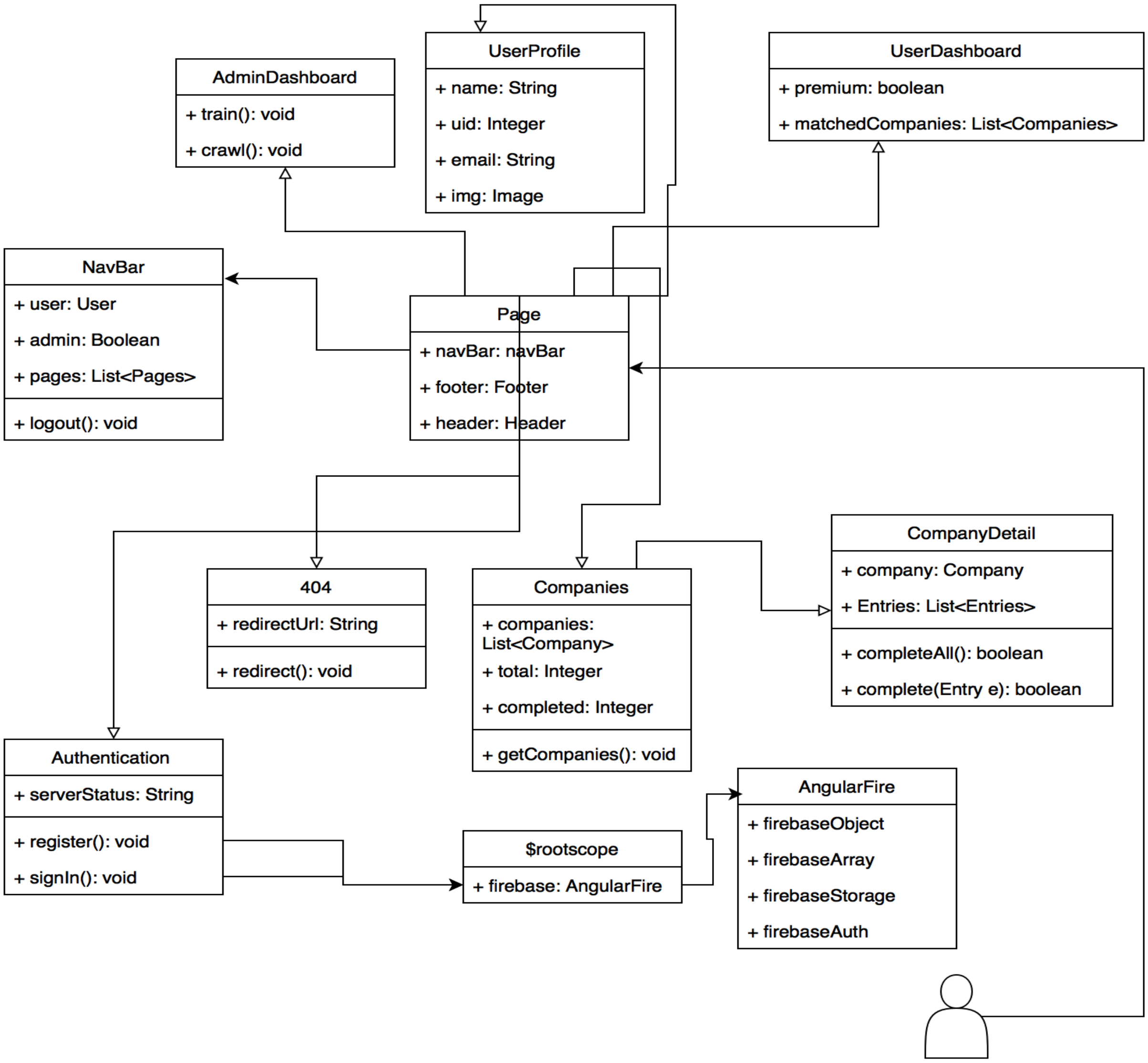
1. Skill
    a. Some interesting skills for certain company. Like C++, java, python, etc.
2. User
    a. User class. Nothing much special.
3. Operation
    a. User operation. Like scroll a mouse, click some button, fill in some entry etc.
4. Util
    a. For utilities. Like some constants like url, keys, config values.
5. Entry
    a. Entry in the form, it could be text box, option, drop down or so on, different user will have different answer for this Entry.
6. Form
    a. Form class will be dealing with literally application forms, it will contains multiple entries.
7. Company
    a. Stores internship company data.

Backend Class Design



1. User (LinkedInInfo, Location, Position, Education)
    a. The user has attributes of firstName, lastName, email, etc in the LinkedInInfo class as well as the legal questions in the application form.
2. Crawler
    a. This is the web scraper that gathers the jobs to apply to and the necessary application form information required to fill it out.
3. Application Form
    a. This is the application that is filled out automatically with the data retreived from the LinkedInInfo as well as the predicted data that is trained from previous instances of the application form.
4. Firebase_admin
    a. This stores the information for the jobs as well as the user's information.
5. Train
    a. This is the class that will be used to train based off the application form.
6. Predict
    a. This class will take the trained data and use it to predict what to fill in on the application form.

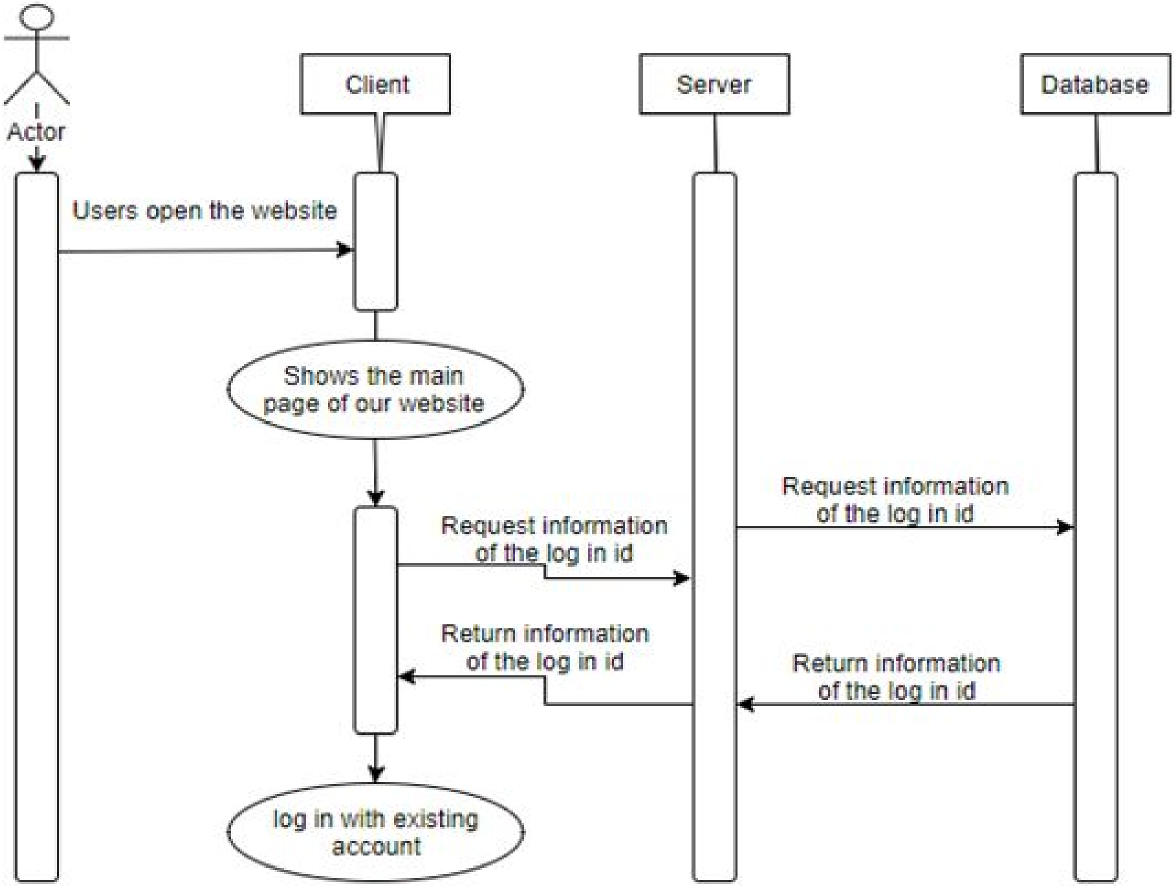
Frontend Class Design



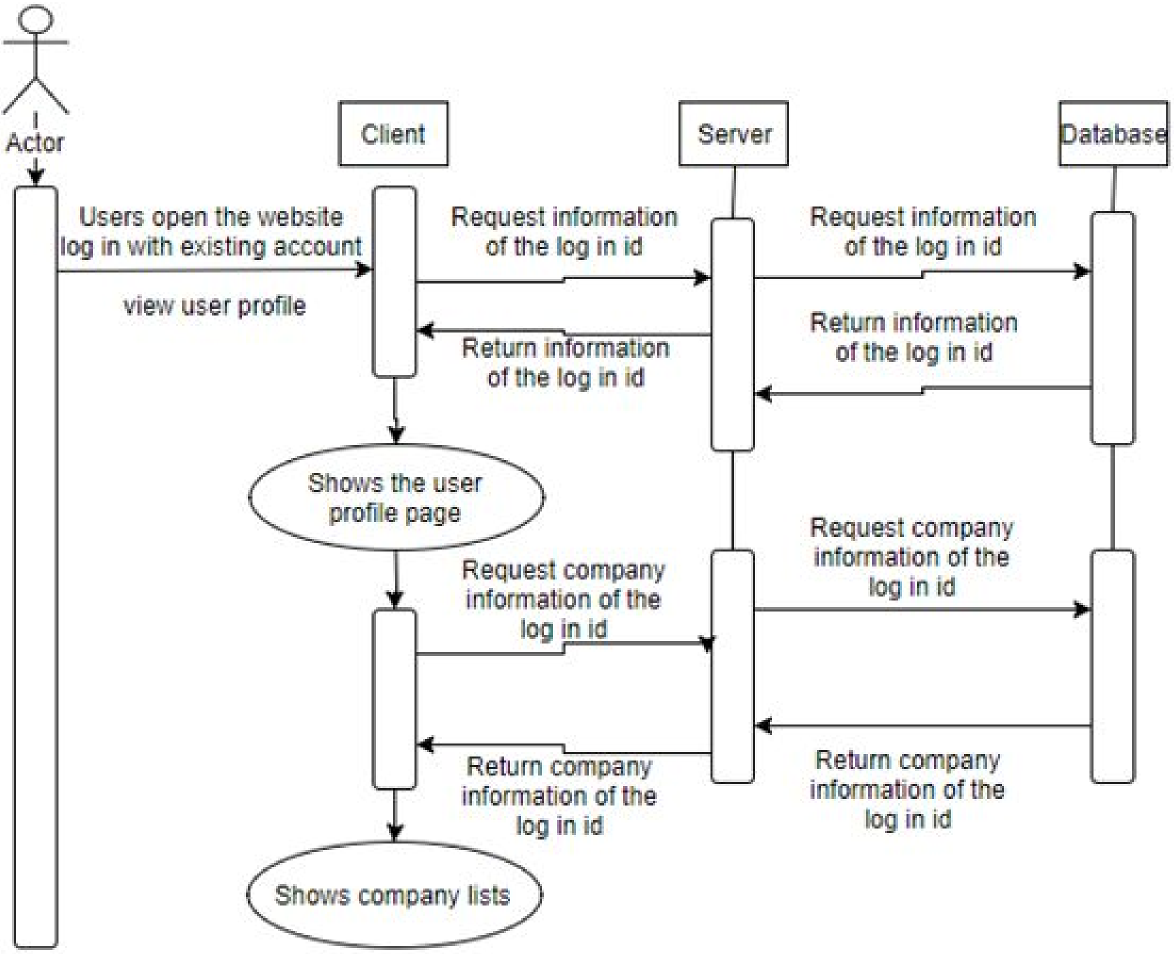
1. NavBar
    a. Navigation bar.
2. Page
    a. The abstract page class, which is good for sticking navbar, footer, header together without changing them in case we wanna change body content.
3. Authentication
    a. Subclass of page, authentication like sign in sign out.
4. AngularFire
    a. Angular friendly class to deal with firebase API.
5. $rootscope
    a. the root scope in angularjs is more like a global variable in the whole app.

Sequence Diagrams

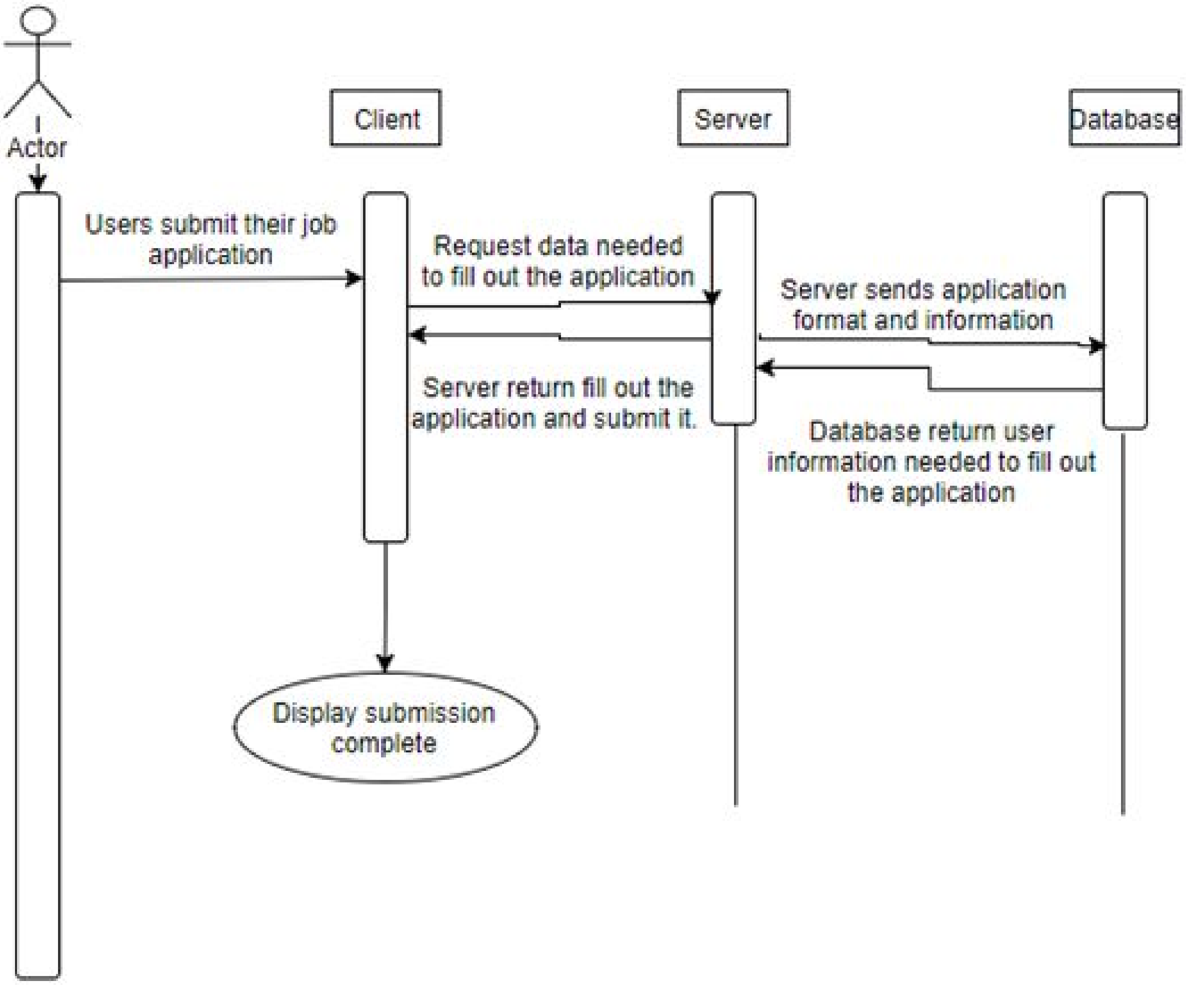
- Sequence of events when the user first log in to our website


Actor
Client
Server
Database
Users open the website
Shows the main page of our website
Request information of the log in id
Request information of the log in id
Return information of the log in id
Return information of the log in id
log in with existing account

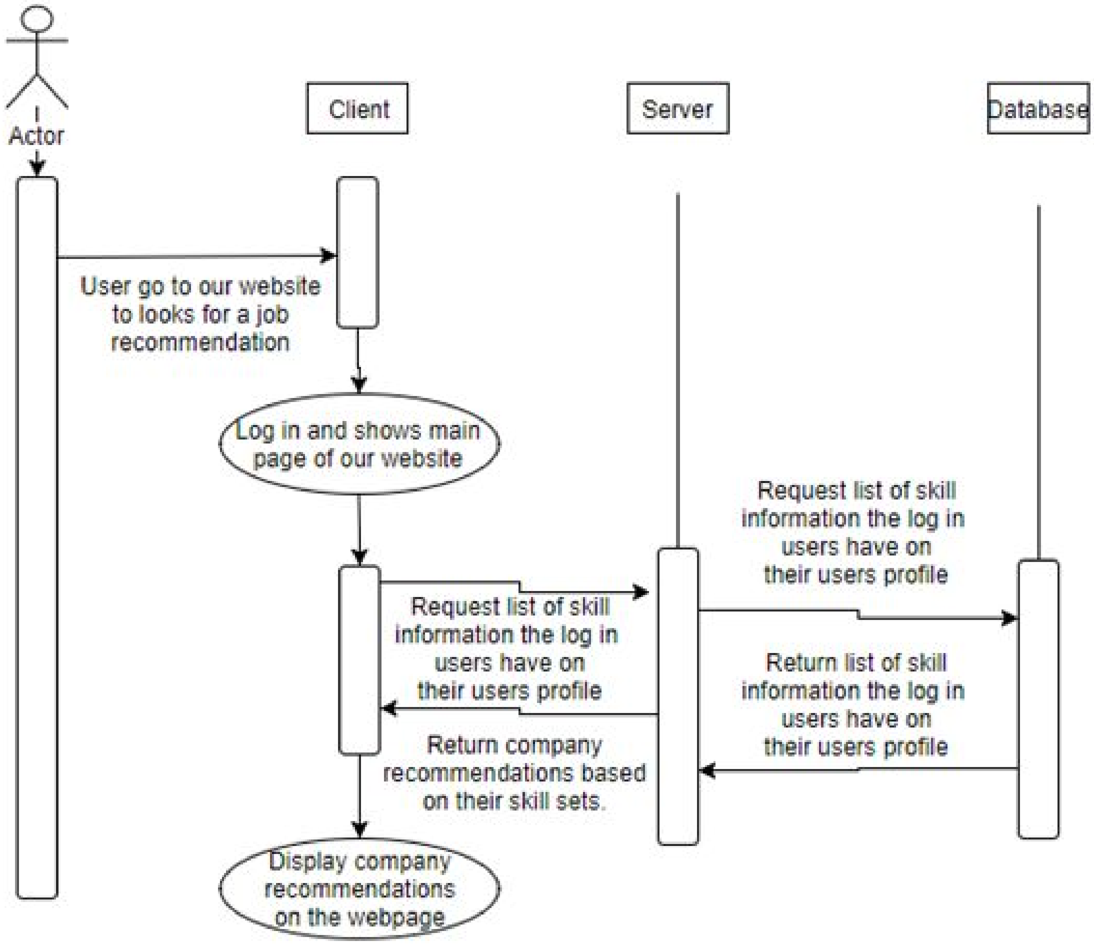
- Sequence of events when the user view their user profile in our website



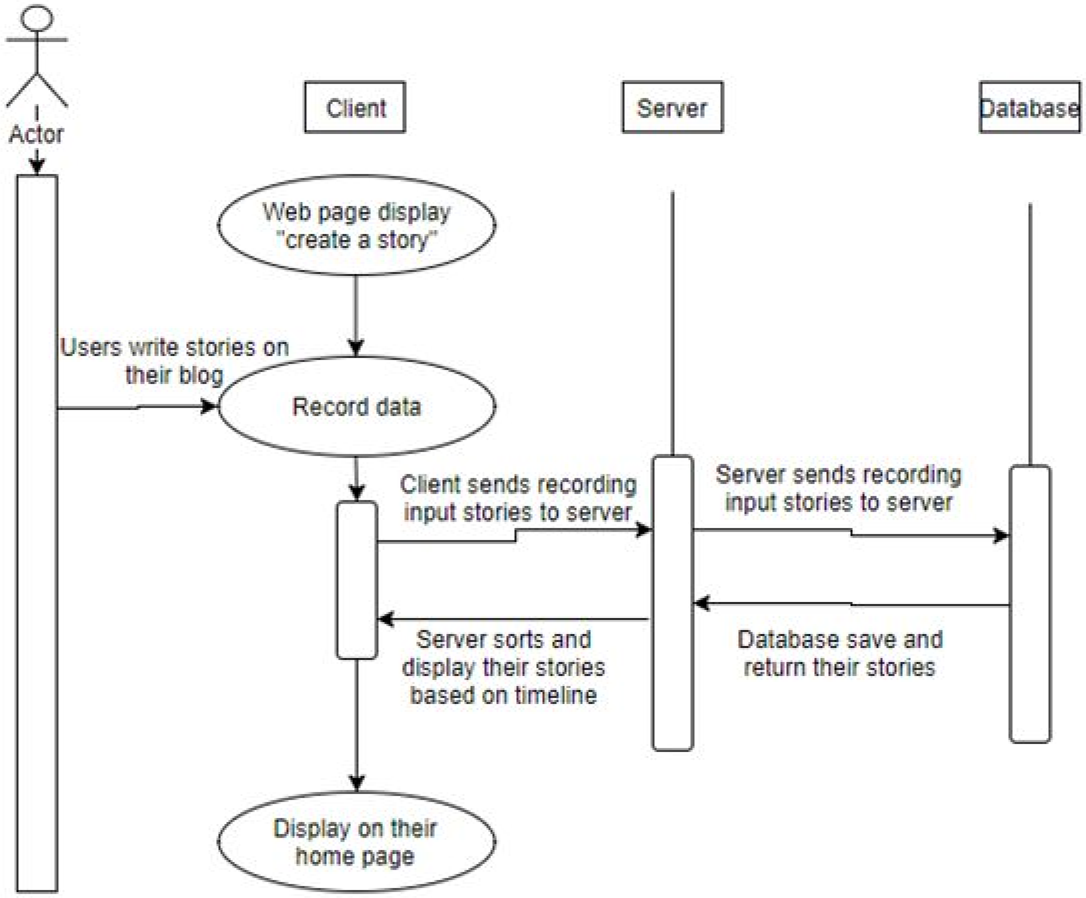
- Sequence of events when the user apply for a job.


Actor
Client
Server
Database
Users submit their job application
Request data needed to fill out the application
Server sends application format and information
Server return fill out the application and submit it.
Database return user information needed to fill out the application
Display submission complete

- Sequence of events when the users want to see job recommandations.


Actor
Client
Server
Database
User go to our website to looks for a job recommendation
Log in and shows main page of our website
Request list of skill information the log in users have on their users profile
Request list of skill information the log in users have on their users profile
Return list of skill information the log in users have on their users profile
Return company recommendations based on their skill sets.
Display company recommendations on the webpage

- Sequence of events when the users want to write and share their stories.



Actor
Client
Server
Database
Web page display "create a story"
Users write stories on their blog
Record data
Client sends recording input stories to server
Server sends recording input stories to server
Server sorts and display their stories based on timeline
Database save and return their stories
Display on their home page