**I. Data Definitions V2**

| **Name** | **Definition & Examples** | **Usage** |
| --- | --- | --- |
| Resume | a user’s resume that is uploaded for business owners to view | A list of these will be shown on a registered business owner with tags selected |
| Skills | categories that associate with jobs the users skilled at, e.g., *Management, Marketing, Java, Full Stack* | Registered users select skills as they register their account and registered business owners use skills to filter out specific ones they are looking for |
| Job | attributes that include job title, skills, salary, location, and type  *e.g, job title: bartender, skills: good communication, mixology, salary: $20 an hour, location: 1234 temp street SF, CA 94116, type: in-person* | Jobs will be posted by employers and contractors will be able to view these jobs and submit their applications for employers to view. |
| User: Contractor | A person who signed up with the application, e.g., *Ben Li, age: 26, gender: male, skills: Databases, Java, Python* | Information from the user is collected and stored. The users information will be made available for business owners to view |
| User: Employer | A person who signed up with the application looking to hire new people, e.g, *Rocky Wade, age: 34, gender: male, business name: Foodverse* | Information from the business owner that is collected and stored. Admins can view this information to verify the business before allowing the business owner to view resumes |
| Account | An account with set permissions on whether it is a contractor or employer. | Users use an account to upload resumes and interact with business owners.  Business owners are able to view these resumes and contact the users |

Data Definitions continued

| **Primary Data Name** | **Sub Data** |
| --- | --- |
| Resume | * resume\_id * user\_id * resume\_path |
| Skills | n/a |
| Job | * job\_id * title * skills |
| User: Contractor | * user\_id * name * username * password * email * usertype |
| User: Employer | * user\_id * name * username * password * email * usertype |
| Account | * username * password * email |

**IV. High-level Architecture, Database Organization**

User

| **Users** |
| --- |
| user\_id int PK |
| name varchar(255) |
| username varchar(255) |
| email varchar(255) |
| password varchar(255) |
| usertype int |

Resume

| **Resume** |
| --- |
| resume\_id int PK |
| user\_id FK -> users.user\_id |
| resume\_path varchar(4096) |

Job

| **Job** |
| --- |
| job\_id int PK |
| title varchar(255) |
| skills varchar(4096) |
| salary int |
| street\_address varchar(255) |
| city varchar(255) |

| **UserSkills** |
| --- |
| user\_skills PK |
| user\_id FK -> users.user\_id |
| skills varchar(4096) |

| **JobApplications** |
| --- |
| jobapplication\_id int PK |
| job\_id int FK -> job.job\_id |
| user\_id FK -> users.user\_id |

| zipcode int |
| --- |
| type varchar(255) |
| user\_id FK-> users.user\_id |

*Add/Delete/Search architecture Functional Requirement*

**Add/Delete/Search for Users** When users register

**Add/Delete for UserSkills** When a user adds skills

**Add/Delete/Search for Resume** When a contractor uploads or deletes their resumes

**Add/Delete/Search for Job** When a employer adds or deletes a job

**Add/Delete/Search for JobApplications** When a user adds or deletes a job application

**Search/Display for JobApplications** When a contractor looks at the users that applied for  
 their job

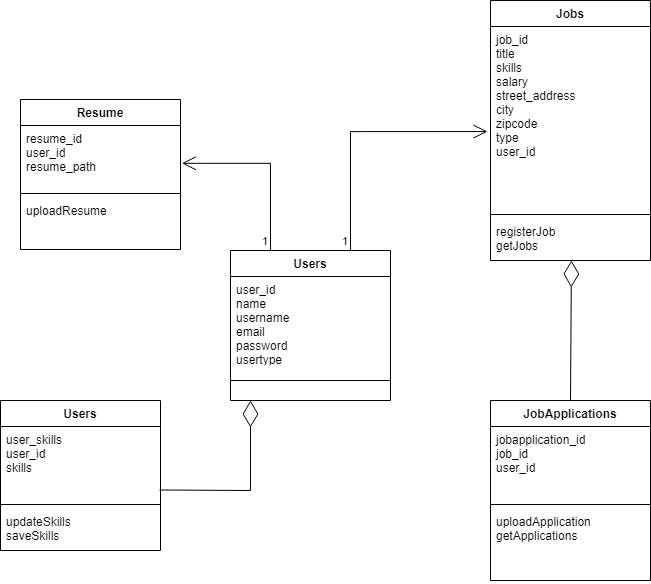
Technical Feasibility of DB operations

We considered the functional requirements for users when using database operations. An example would be when a userfilters out jobs based on skills. We would then look up the **Jobs** table, get any **Job\_ids’** that match the skills and display the jobs for the user to view and apply. When the user submits an application, a new row is added to the **JobApplications** table. Contractors can then view who applied to their job by getting the applications that match the **Job\_id** that they have selected.

*API’s*

Our app will use an API created with express, allowing us to add, delete, and search through our database.

**V. High-Level UML Diagrams**



High Level Sequence Diagram

