

## High level architecture: CRC cards

User	Collaborators
<ul style="list-style-type: none"> <li>• - profile</li> <li>• - listOfAppliedJobs</li> <li>• - listOfTeamsJoined</li> <li>• - int selfID</li> <li>• + edit_picture(png)</li> <li>• + edit_resume(pdf)</li> <li>• + edit_paragraph(string paragraph)</li> <li>• + apply(int jobID)</li> <li>• + remove_application(int jobID, int selfID)</li> <li>• + rate_user(int rating, int ID)</li> </ul>	<p>Profile Jobs Project</p>
Owner – inherits from User	Collaborators
<ul style="list-style-type: none"> <li>• - listOfOwnJobs</li> <li>• - listOfTeamsOwned</li> <li>• + CreateJob(parameters for team (may vary in time))</li> <li>• + CreateProject(parameters for team(may vary in time))</li> <li>• + addTeamToTeamList(parameter)</li> <li>• + addJobToList(job)</li> <li>• + upload(job)</li> <li>• + takeDownListing(job)</li> <li>• + check_applications(job_id, selfID)</li> <li>• + addMemeberToProject(project, int ID)</li> <li>• + DeleteMemeberFromProject(project , int ID)</li> </ul>	<p>Profile Jobs Project</p>
Job	Collaborators
<ul style="list-style-type: none"> <li>• - paragraph</li> <li>• - tag</li> <li>• - dayCreated</li> <li>• - dayEnd</li> <li>• - numberOfPeople</li> <li>• - listOfApplicants</li> <li>• + applyTo(id)</li> </ul>	<p>User Owner</p>

Project	Collaborators
<ul style="list-style-type: none"> <li>• - owner</li> <li>• - listOfJobs</li> <li>• - listOfEmployees</li> <li>• + notify()</li> </ul>	User Owner Job

UI Manager	Collaborators
<ul style="list-style-type: none"> <li>• + createUser()</li> <li>•</li> </ul>	User Owner Job

Profile	Collaborators
<ul style="list-style-type: none"> <li>• - picture</li> <li>• - resume</li> <li>• - paragraph</li> <li>• - name</li> <li>• - tag</li> </ul>	User Owner Job

### Pages & UI Elements:

- Homepage/Sign up page
  - Sign up: right of the page
  - Task bar (with log in changes when user logs in)
  - Info about the site (left)
- Profile page
  - Resume upload
  - Skills (list all your skill)
  - Profile Picture
  - Self paragraph (short introduction)
  - Past Project times lines (sort of like GitHub)

- (If owner) see all going project and past projects
  - all applicants to current projects
  - if owner can rate other people
  - what job you are
- Job listings:
  - List of on going projects
  - Summary of each projects
  - A way to message and apply to job
  - Search bar
- Teams page
  - A page that links a team together
- Applicants Page
- Project Page
  - Has a List of all the people inside the project
  - Has the Owner of the project
  - Has a List of all the jobs in the project
- Successful sign up

### **Modules:**

- Math (imported)
- Login/sign up
- Edit Profile
- Connect
- User
- Owner
- Job
- Project
- UI manager
- Profile

### **Design pattern:**

Model-View-Controller: we used a Model based system where each model has its own unique specifications and functions, through UIs you can change these elements in the models to alter these values and change the display on the website. For example our User, Jobs are all different models that can be changed and the view is changed as well as result.

### **Features:**

- Log in
- Sign up
- Profiling
- Search for jobs under a certain tag

- Searching jobs under a certain owner
- Search for a profile under a tag
- Rating a person out of 5 stars after the job is completed
- Post jobs to database
- Display list of jobs from database
- Apply to Jobs on database
- Unapply to jobs on the database
- Edit personal profile
- Save Picture to Database and display it
- Create a Project with multiple jobs and people
- Google log in
- Contact owner of job

### **Rest API Designs:**

Get:

Main Page:

- Used to retrieve all jobs in database to display, Get will return all the Jobs in the database
- Used to retrieve all jobs in a search, Get will return all the jobs that is associated under a specific tag
- Used to retrieve all profiles in a search, Get will return all the profiles of users that is associated under a certain tag

Profile Page:

- Used to retrieve all resumes, when looking at a persons profile get will retrieve that persons resume

Project Page:

- Used to retrieve everyone in the current project, Get will return everyone in the Project
- Used to retrieve all the jobs in the current Project, Get will return all the Jobs in the Project

Post:

Sign up Page:

- When signing up it will save all of your info, Put will send all of your information such as username, password and profile info into the database and it will return a page saying successful signup

Owner page:

- When creating a post it will save all the job information into the database, Post will send all the job information into the database

Delete:

Profile page:

- User can delete their account through the settings in the profile page, Delete will delete the account from the database and return an alert saying you completed it
- Owner can take down his or her own jobs from their profile page, Delete will delete the job from the database and return an alert stating you completed it

Put:

User Profile page:

- User can change their profile description through their profile page, Put will update the description in the database and will return an alert saying you completed it
- User can change their Profile picture through their Profile page, Put will update the profile picture and return an alert saying you completed it
- User can change their Online resume link, Put will update the online resume in the database and will return an alert saying you completed it

Owner Profile Page:

- Owner can change the description of any of the jobs they have created, Put will update the information within the database and return an alert saying you completed it
- Owner can change the number of wanted people and the end date of their post, put will change these attributes within the database and return an alert saying so

Team Page:

- Owners can change the Team Members of a Team they own, Put will update the team members in the team and return an alert
- Owners can change the team jobs of a team they own, Put will update the team jobs and return an alert