SOFTWARE ARCHITECTURE DOCUMENT

Project Title: ApplyForMe
Team Name: Team Hydraulics
Type of Document: Software Architecture Document

Project Description

You work at a remote company and you earn $1k a month. You are very happy. But you sometimes think - what if there are better jobs out there for me? But it's too much work to apply. In comes: ApplyForMe. A service where you tell us your skills and what you are looking for, and people apply for you for 100s of jobs. All you need to do is attend interviews.

Overview

Working in a remote company is fun and stress free but sometimes people wonder if there are better opportunities out there for them, but there's something else – The stress of job searching. Job searching could be stressful and draining to some people and taking all these into consideration, we developed ApplyForMe. An application that helps the Developers to search for jobs that could be fitting for them. All they need to do is to attend interviews.

ApplyForMe is an application that aids users to apply for a job on their behalf. In this project, there will be two people interacting with the system: the actors (users) and the agents. The actors will register and login using their saved login details. They will be prompted to fill in any necessary information, attach documents or links needed for the application and will await feedback and this is where the agents come in. The agents will login into their dashboard. Seeing a list of applicants, they'll pick one person to work on and go through their details before helping them to apply. Once it's done, they move on to another applicant while the applicant who has been worked on will receive a notification and further instructions on what to do next.

ApplyForMe is a platform that acts as a job search agency provider:

A job search agency also known as a reverse recruiter assumes all the job searching functions for their clients. A reverse recruiter works for a job seeker

A traditional recruiter works for the hiring company while a reverse works for the job seeker.

The product would have two actors including the Job seeker and Reverse recruiter

Terms:

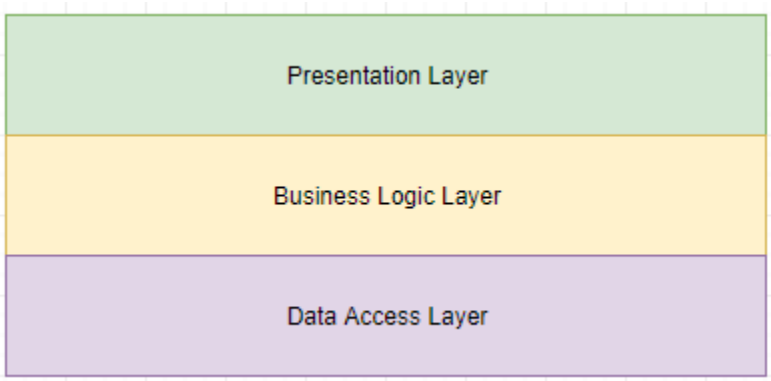
1. **Developer/Professional/Job Seeker/Applicant:** The person looking for better opportunities aside its current role.
2. **Agent or Applier or Reverse Recruiter:** The person who does the actual job application on behalf of the developer.

FUNCTIONAL REQUIREMENT & MINIMUM VIABLE PRODUCT (MVP)

- **Developer** registers on platform.
- Then submits professional details and documents and also indicates the desired job it is soughting for.
- **Applier** is onboarded internally on the platform and is a working member of the company.
- An applier can view profiles of developers and is assigned to one.
- It can view the list of jobs already submitted for the developer by previous appliers' assigned to the developer.
- It makes job application submissions outside the platform on the developer's behalf.
- It creates records of jobs applied to on behalf of that developer.

Application Architecture

(3) Three-tier Architecture

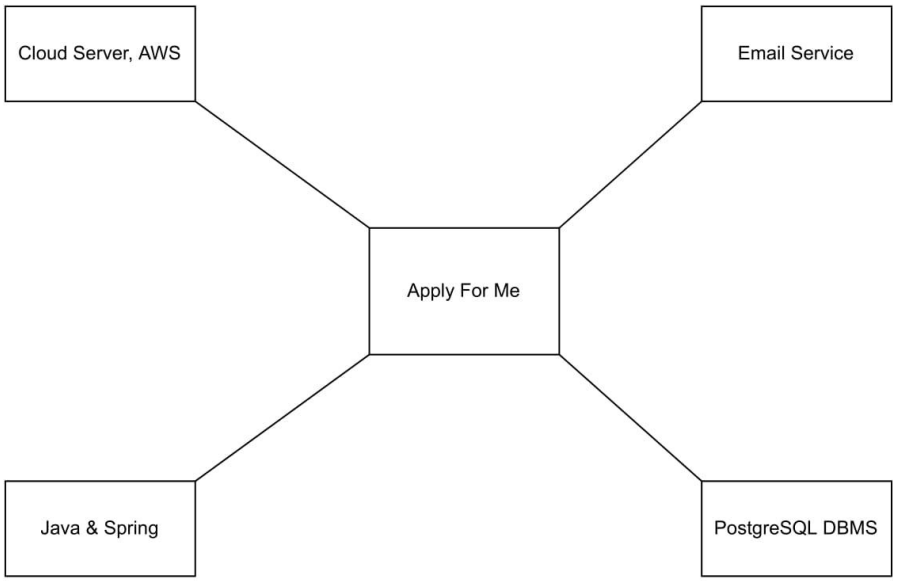


Presentation Layer: The layer at which users interact with the application and the final data will be visible to the users at this interface. It acts as an interface between the user and the application and it will employ a frontend programming framework.

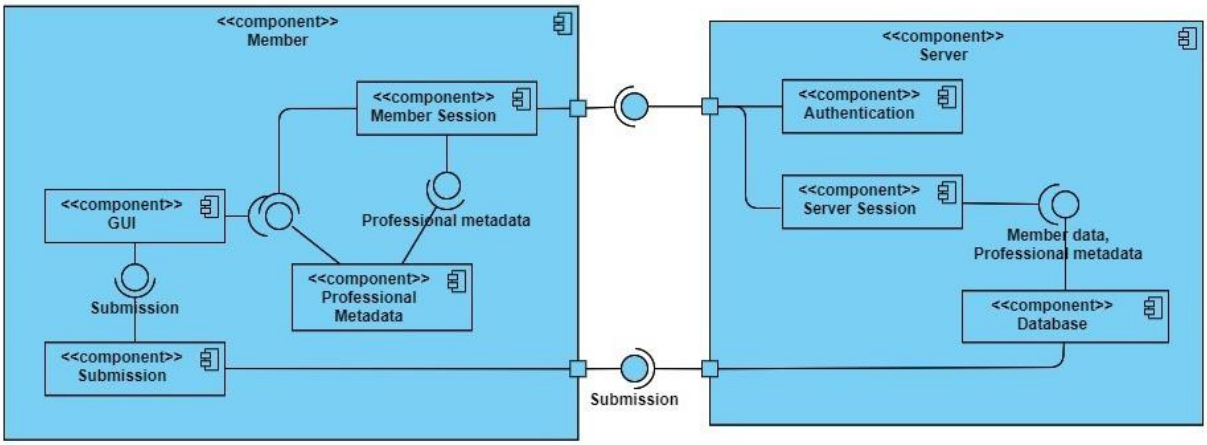
Business Logic: It acts as an intermediate between the Presentation and the Data Access Layer. It uses Java Runtime and Spring boot framework.

Data Repository and Access: The layer at which the data is persisted & managed, it uses PostgreSQL.

SYSTEM CONTEXT DIAGRAM



CONTEXT DIAGRAM



Components:

1. Member
2. Server

Member

This entails the system users – the professional (job applicant) and the applier (one who applies to the jobs on behalf of the professional).

Server

This is the actual computer hosting the ApplyForMe system. It includes the database and code.

Definitions:

- **Provided Interface**

  This indicates what the component provides to the rest of the system or the next component.
- **Required Interface**

  The component takes information/data it requires through the required interface.

Components

1. **Database**

   This serves as the storage for all ApplyForMe data. This component provides member data and professional metadata to the rest of the system.

2. **Server session and Authentication**

   Server session is responsible for starting up and running the server.

   Authentication component performs necessary security checks to verify validity of the system users.

3. **Member Session**

   Responsible for handling user sessions. This component supplies the professional metadata information.

4. **Graphical User Interface (GUI)**

   This provides the user interface a way through which system users interact with the system. It requires professional metadata and member data and provides submission data to the submission component.

5. **Professional Metadata**

   Supplies professional metadata to the GUI.

6. **Submission**

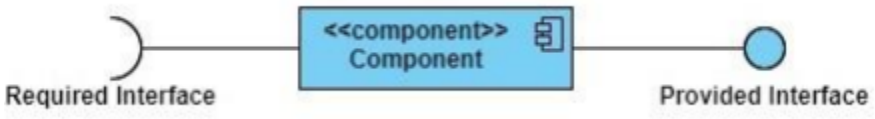
   Receives submission data from the GUI and proceeds to provide it to the database component for persistence.

   All users interact with the system through the GUI, in turn interacting with either member data, professional metadata, or submission.

   A Developer can submit, edit, or view member data and professional metadata. He/she can also view submission details.

   Applier interacts with professional metadata and uses it to create a submission.

Definition:



- **Provided Interface**

  This indicates what the component provides to the rest of the system or the next component.

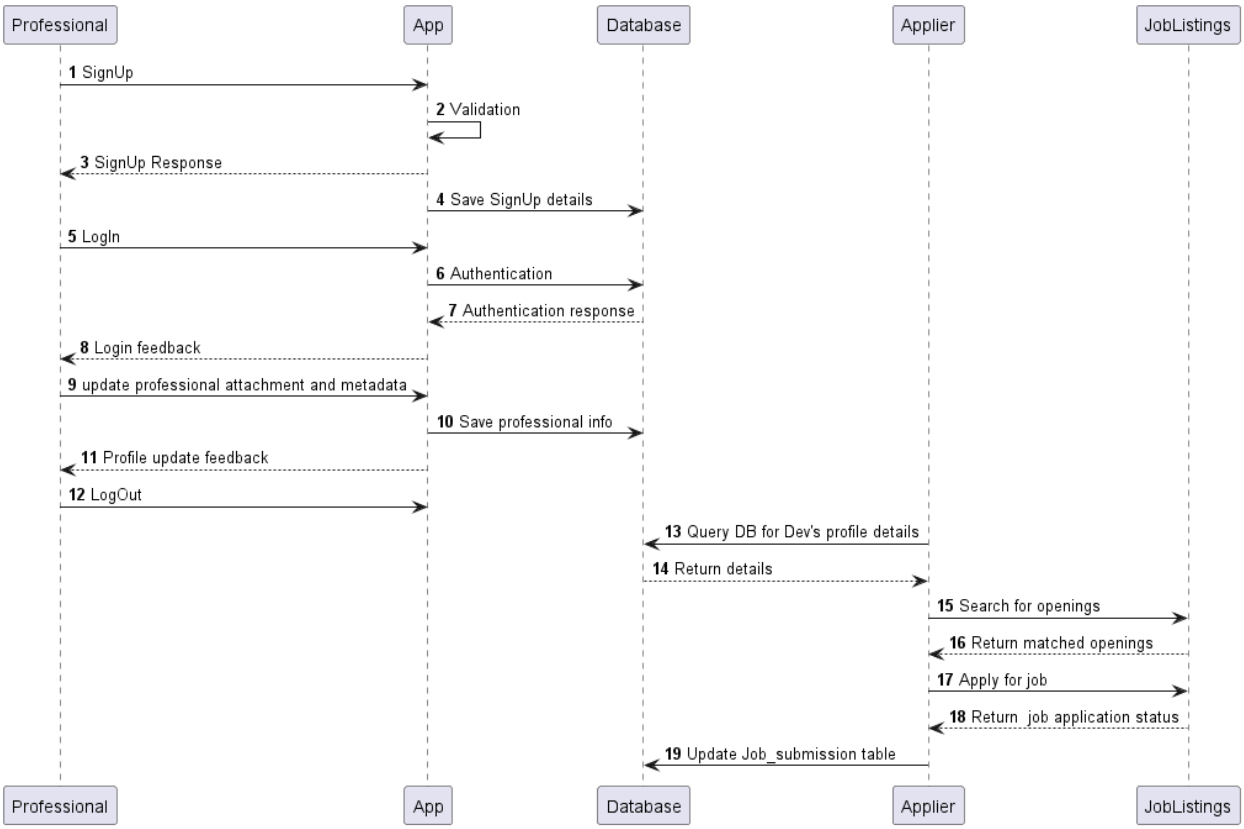
- **Required Interface**

  The component takes information/data it requires through the required interface.

- **Component**

  Component diagrams are called black-box views. Ports are shown as squares bordering the component, these indicate how the *interfaces* of the component are used internally. Objects implementing a *required interface* are received via a port and objects implementing a provided interface are shared via a port.

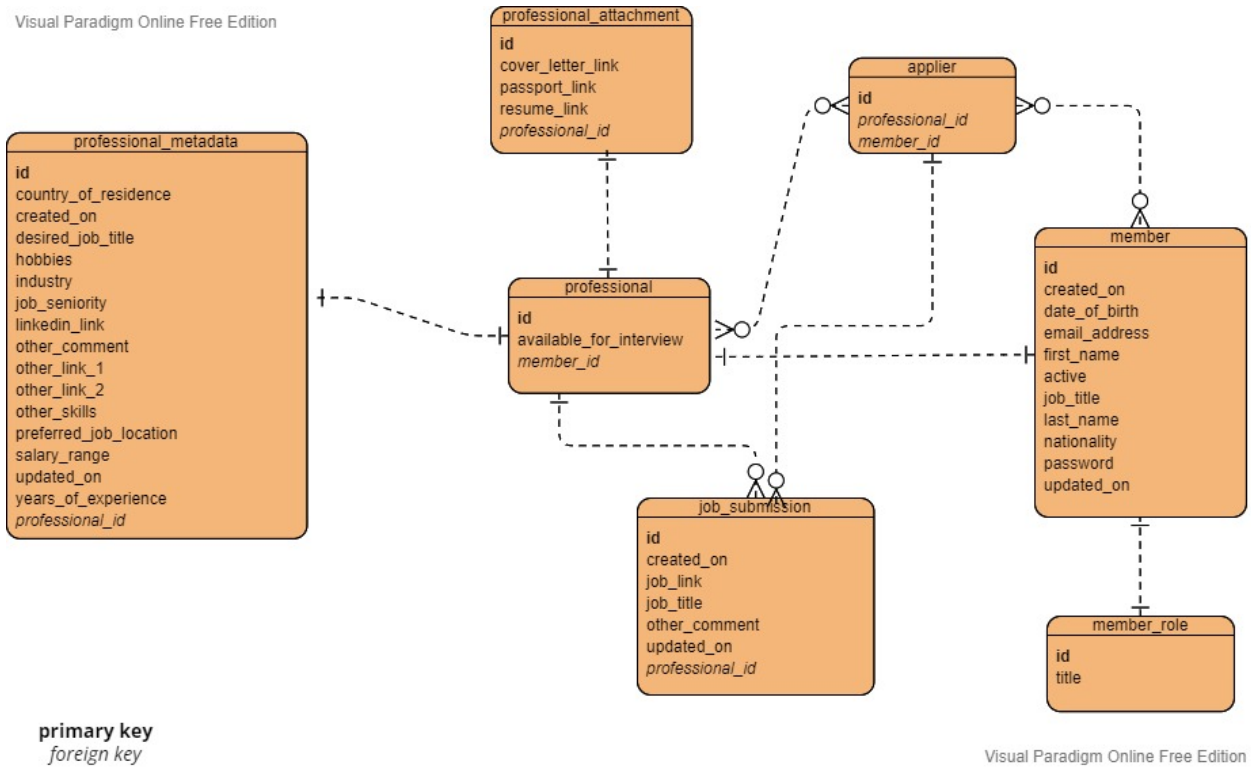
SEQUENCE DIAGRAM



ENTITY RELATION DIAGRAM (ERD)

Updated Reference to SQL File: Schema SQL File

Reference to ORM Models



List of Tables

- member
- member_roles
- roles
- professional_metadata
- professional_attachment
- applier
- job_submission
- professional

member: The member table stores the personal data and login details of new users (both job-seekers and agents) on registration. It has a one-to-one relationship with the appliers' and professionals.

role: This table details the list of enumerated or predefined roles or privileges that can be assigned to a member of the system.

member_roles: This table maps new users and assigns them to given roles and privileges in the application. Here a user is assigned the role of a job-seeker, an agent or an admin. It has a many-to-many with the member table because a user can have more than one role.

professional_metadata: This table holds all relevant professional details of the job seeker. Information stored in this table is used by an ApplyForMe agent to match job seekers with related jobs to their qualifications and sends job applications on behalf of the job seeker. It has a one-to-one relationship with the professional table.

professional_attachment: Links to attachments relevant to a job seeker profile such as resume, cover letter, passport etc. are stored in this table. It shares a one-to-one relationship with the professional table.

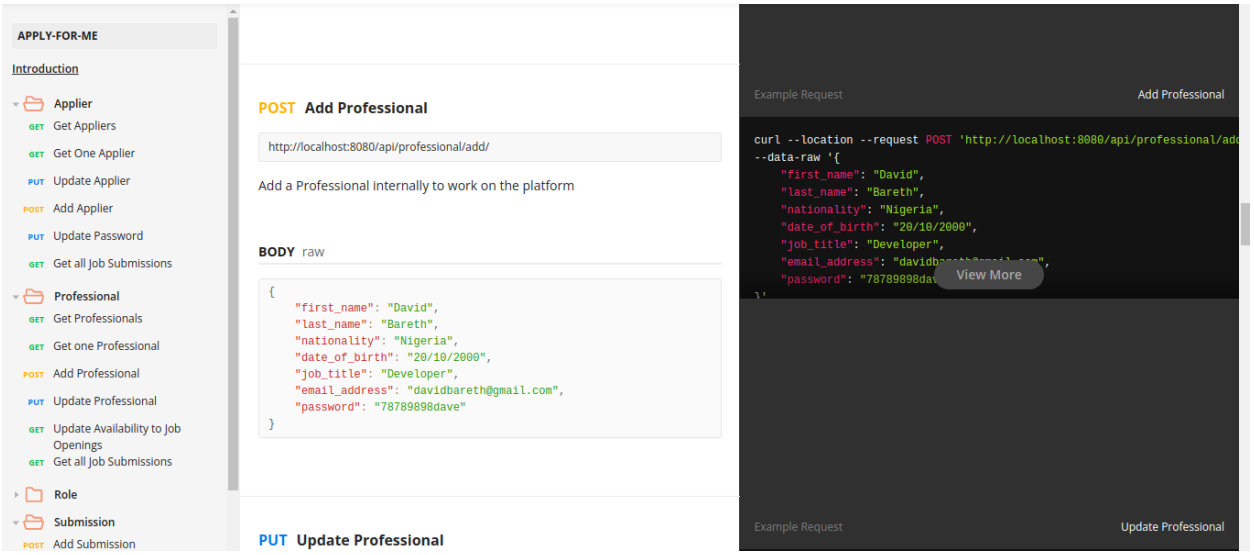
applier: The applier table stores a list of job seekers whose applications have been sent to job openings relevant to their qualifications. It shares a many-to-many relationship with the member and professional tables and a one-to-many relationship with the job_submission table.

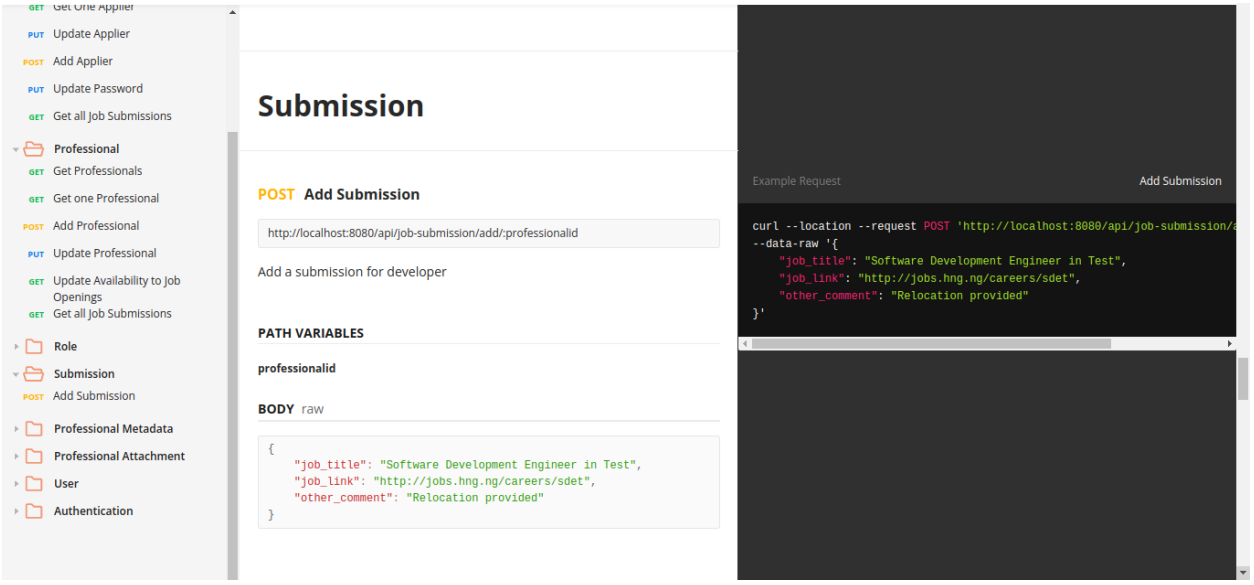
job_submission: Details of all the jobs applied to on behalf of a job seeker are stored in this table including the date and time the applications were sent. It shares a many-to-one relationship with the applier and professional tables.

professional: This table stores all successful job applications that received a response from the hiring managers or companies requesting for an interview with the job seekers.

API Documentation

Postman Documentation for ApplyForMe -





SCENARIOS

Jane is a mid-level backend developer at HNG. Jane earns $1k per month. She's very happy with her job but is considering greener pastures. Jane is so busy she chooses to submit her details to ApplyForMe for other people (Agents) to apply for jobs on her behalf.

Developer's Journey

1. Jane registers on the ApplyForMe platform. During the registration, she provides the following details:

   Personal details or Bio data

   - First name
   - Last name
   - Nationality
   - Date of birth
   - Job title (current job)
   - Email address
   - Password

2. Jane then proceeds to login using her email address and password as login credentials.
3. She submits her professional details. After which they get saved to the database.

   Professional details

   - Salary range
   - Country of residence
   - Preferred job location
   - Job seniority
   - Desired job title
   - Other skills
   - Other comments
   - Passport picture*
   - Resume/CV*
   - Cover letter*

   NB: - CV and cover letter can be updated later.

4. Jane logs out and awaits feedback concerning the jobs applied to on her behalf.
   For instance, say Agent 1 applied to Google and Amazon, and Agent 2 applied to Facebook and Twitter. So therefore, Jane awaits feedback from those companies since the application was done on her behalf.

   Let's assume it's Jane who applied to their job listings, not an ApplyForMe agent.

Agent Journey

David is an agent. His work is to apply to jobs on behalf of developers (job seekers), Jane in this scenario.

1. David registers on the apply for me platform.
2. David logs into the ApplyForMe platform using his email and password as login credentials.
3. He gets to see a list of developer submissions (professional details). He picks on one job seeker and proceeds to view their professional details.
4. Based on the professional details, David proceeds to search for matching jobs on sites like Indeed and LinkedIn and applies to them on behalf of Jane.
   He records the jobs applied to – company name (e.g Google), position details. (This is to ensure the next agent doesn't apply to the same jobs at the same company.)
5. David either proceeds to another developer or logs out.