# JobQuest Navigator – Week 3 Sprint Document

Maria Soto, Shruti Amit Vasanwala, Zhihuai Wang, Ishan Aakash Patel

Team 9

The Zombies of CAA

Seneca Polytechnic

Course Code: CAA900

David Chan

## 1. Sprint Goals

## Weekly Creation Meetings:

## The team meets weekly to work on the project. Each session is documented with a specific meeting acta (minutes), capturing decisions, action items, and progress.

Please find the attached link for the minutes document here: [Minutes Document](https://drive.google.com/drive/folders/1_QNxhZVV_Nq5ksKc8DEf4A7CWohQlDN2?usp=drive_link)

## Design Diagrams & Technical Information:

## All technical diagrams (system architecture, data flow, etc.) and supporting technical documentation are maintained and updated as the project evolves.

Please find the attached link for the diagrams here: [Diagrams](https://drive.google.com/drive/folders/1o9tQsh2Taw5iWUV7RmbJD0ATAra7NKQA?usp=drive_link)

## 2. Tasks

List of the tasks on Jira in Progress. [List](https://myseneca-team-pi6s3gm8.atlassian.net/jira/software/projects/SM/list?sortBy=customfield_10015&direction=ASC&atlOrigin=eyJpIjoiYzQ5MTE5YjAxZGExNGM1YmE2ODgyZjg3MGFkMjhiNTMiLCJwIjoiaiJ9)

A screenshot of a computer

AI-generated content may be incorrect.

## 3. Key Features Being Developed

## Technical Design Document (TDD)

## Technology Stack Selection Based on the Product Requirements Document (PRD), the following technology stack has been selected:

## Frontend: React.js

## Backend: Django (Django REST Framework)

## Database: MongoDB (for unstructured data, AI suggestions), MySQL (for structured data)

## Map Service: Google Maps API (job mapping functionality)

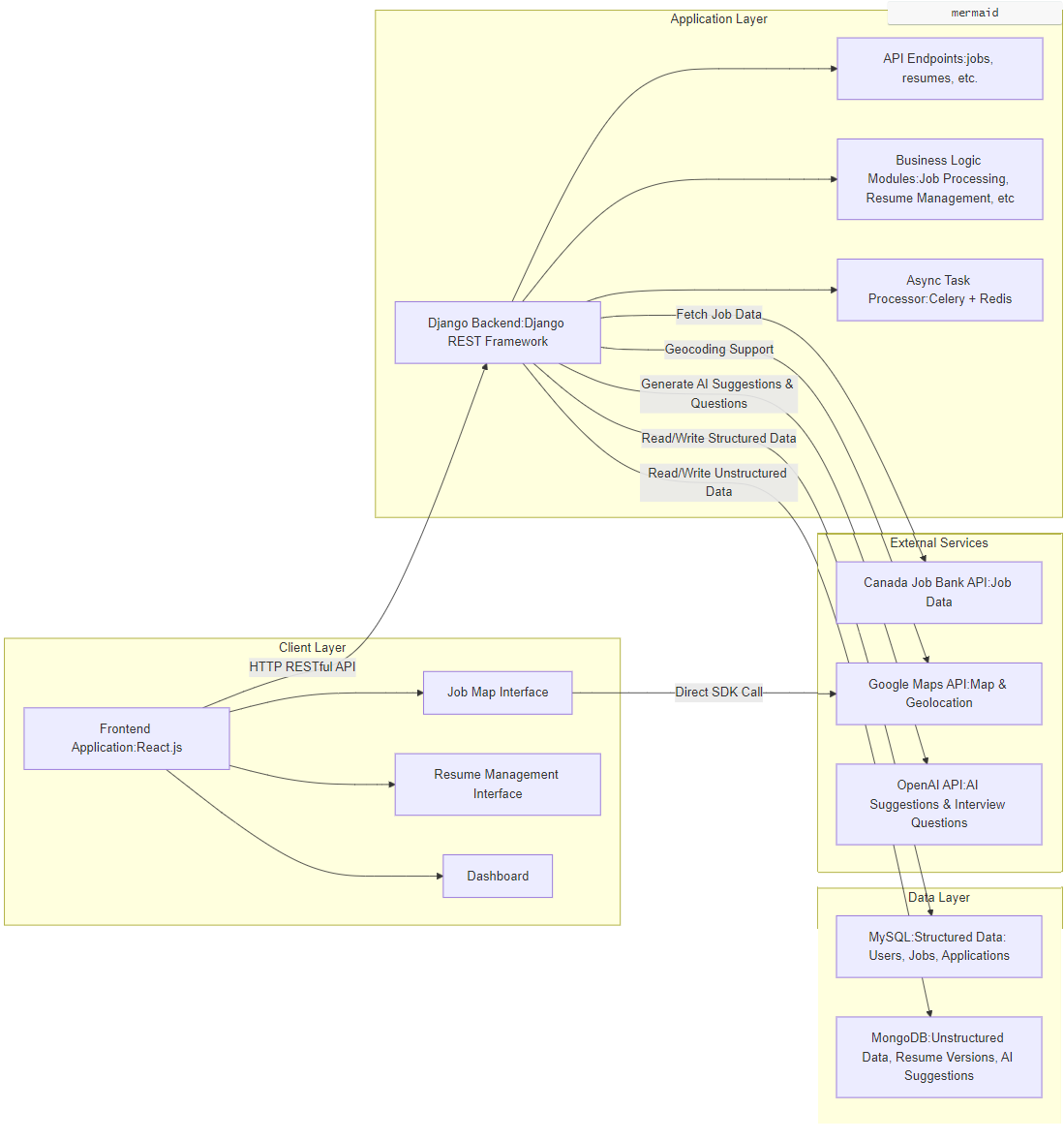
## AI Service: OpenAI API (AI suggestions, interview questions)

## Third-party API: Canada Job Bank (job data integration)

1. Technology Stack Selection

The system is designed for modularity and scalability, supporting independent development across different epics and clear separation between frontend and backend.

1. **System Architecture Diagram**



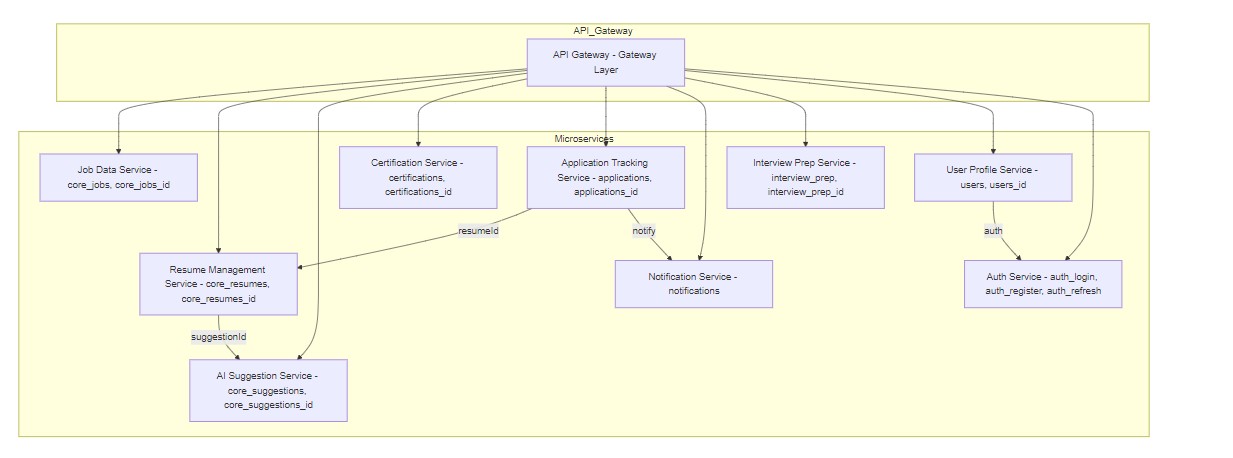
1. **Environment Setup**

## Development, Testing, Production Environments: All environments are containerized using Docker for consistency.

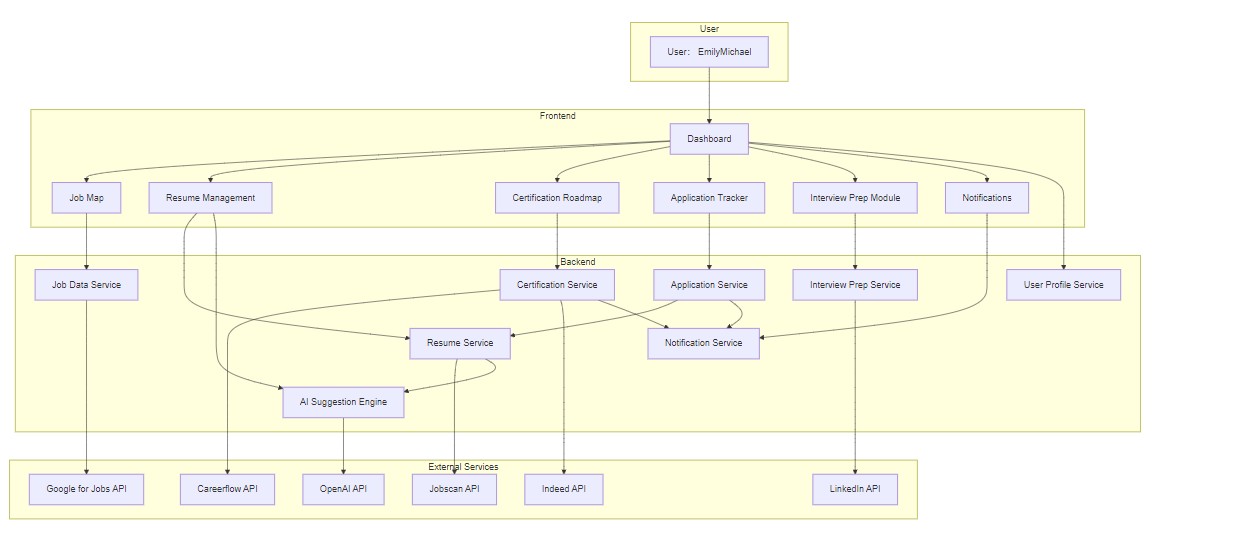
## CI/CD Pipeline: Continuous integration and deployment are managed via Jenkins or GitHub Actions.

## Version Control: GitHub is used for code management and collaboration.

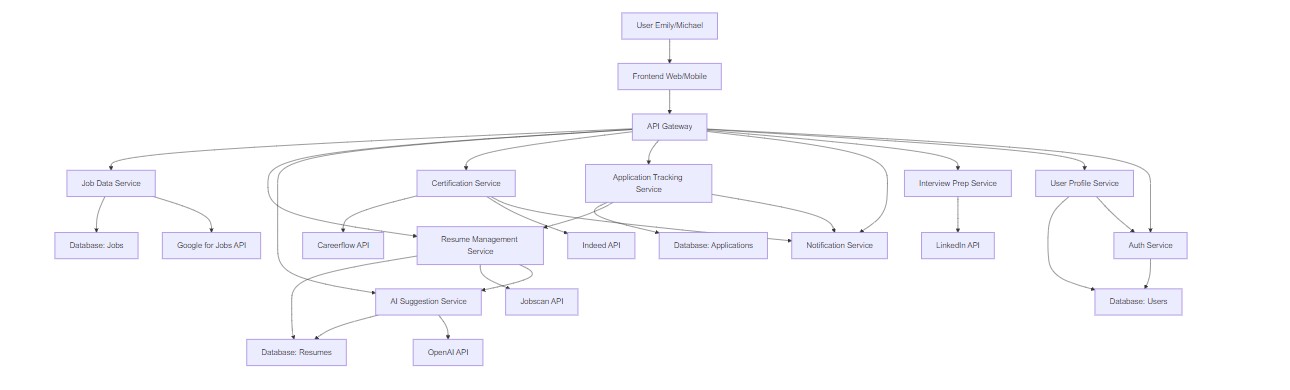
1. **API Architecture Diagram**



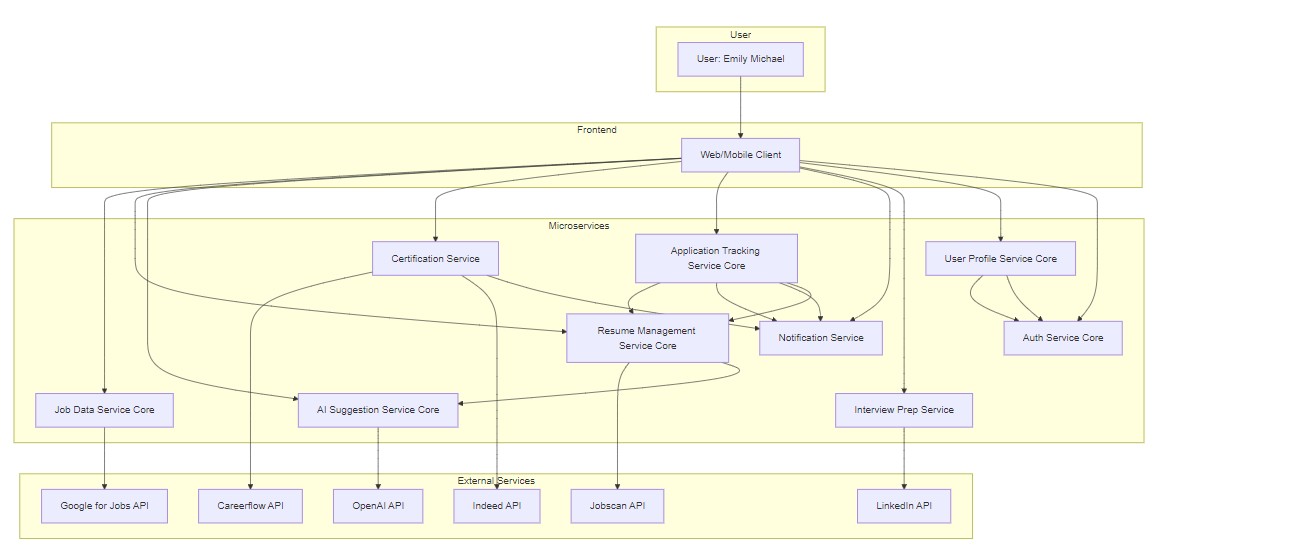
## Business Architecture Diagram



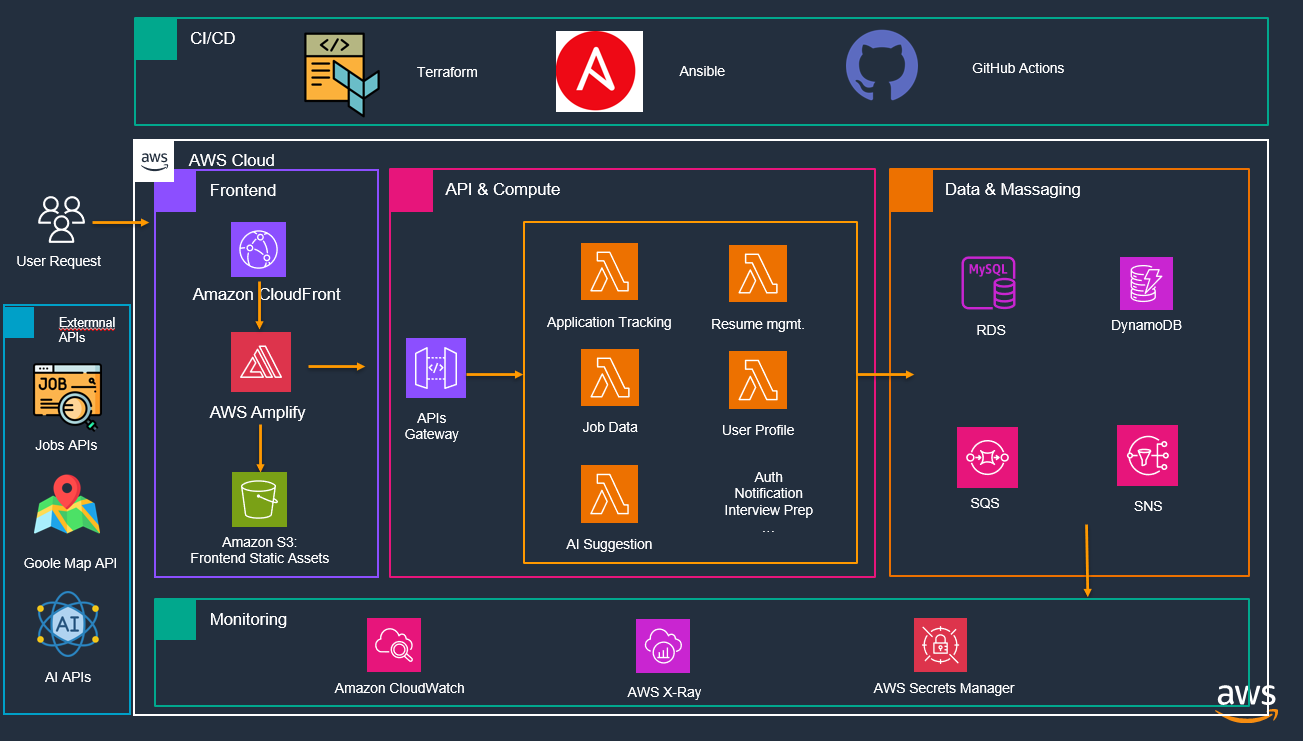
1. **Data Flow Diagram**

****

1. **Microservices Architecture Diagram**

****

1. **AWS Architecture based on System Architecture Diagram**

****

## 4. Progress Overview

* Reviewing and finalizing the logical architecture diagrams to ensure all system components and interactions are accurately represented.
* Detailed epic diagrams are being developed to clearly define the scope of each epic, facilitating effective task planning.
* The initial template for the Job Mapping Interface (UI) under Epic 1 was successfully created, laying the groundwork for Basic Login Signup
* The Data Dictionary was completed to standardize data definitions and support consistent development across the project.

## 6. Workflow and Tools Used

- Jira is used for requirement traceability.

- GitHub Projects is used for issue tracking, sprint boards, and automation.

- All user stories and tasks have GitHub Issues with labels and links to the PRD.

- Backend: Django (Django REST Framework)

- MongoDB (for unstructured data, AI suggestions), MySQL (for structured data)

- Google Maps API (job mapping functionality)

- OpenAI API (AI suggestions, interview questions)

- Canada Job Bank (job data integration)