BrainStation®

Program Diploma - Capstone Project

Project Title:	JAS “Job Assignment System”
Author:	Shaymaa Abdelrahman Abdou
Cohort:	*Software Engineering Bootcamp April 11, 2023*

1. Project Overview

1.1 Description

A React application that uses employee information, location, and fault information to assign jobs automatically would be a powerful tool for businesses that need to manage field service operations. The application would allow managers to know information about their employees, such as their skills, availability, and location, as well as information about the jobs that need to be completed, such as the type of work required and the location of the job site.

1.2 Problem

This application will provide an opportunity to explore important issues related to field service management and optimization through:

- Gathering employees’ locations data and communicating with assets’ database.
- Displaying map of geographically distributed employees and assets.
- Ensuring fast and reliable delivery of tasks requests to assigned employees.
- Filtering employees based on their distance, availability, specialization or based on a specific filter developed separately.
- Finding alternative employees in case of declining or reassigning a task.

I got my background information and the idea of this application through my work in an electrical distribution company in which I found that the faults in the network was assigned manually. And I found this application will be very helpful in the field of industry and companies.

1.3 User Profile

The end users of this React application would be managers who are responsible for assigning jobs to employees in a field service operation. The application would allow managers to input and view data related to employee skills, availability, and location, as well as job requirements such as the type of work and location of the job site.

The application would help managers by automatically assigning jobs to the most appropriate employee based on their skills and location, ensuring that jobs are completed efficiently and effectively. The application would also provide real-time updates on the status of jobs, allowing managers to track progress and adjust as needed.

Special considerations for the design of this application would include ensuring that the user interface is intuitive and easy to use, with clear visualizations of employee and job data. The application should also be designed to be responsive and accessible, with support for different devices and assistive technologies. Finally, the application should be designed with security and privacy in mind, with appropriate measures in place to protect sensitive employee and job data.

1.4 Requirements: Use Cases and Features

Use Case: A field service company wants to optimize their job assignment process by using a web application that automatically assigns jobs to employees based on their skills, availability, and location. The application should allow managers to input and view data related to employee skills, availability, and location, as well as job requirements such as the type of work and location of the job site. The application should also provide real-time updates on the status of jobs, allowing managers to track progress and adjust as needed.

Features:

1. Employee Management: The application should allow managers to input and manage employee data, including their skills, availability, and location.
2. Job Management: The application should allow managers to input and manage job data, including the type of work required and the location of the job site.
3. Automatic Job Assignment: The application should use algorithms to automatically assign jobs to the most appropriate employee based on their skills and location.
4. Real-time Updates: The application should provide real-time updates on the status of jobs, allowing managers to track progress and make adjustments as needed.

5. Reporting and Analytics: The application should provide reporting and analytics features that allow managers to analyze job assignment data and optimize their processes over time.
6. User Management: The application should provide user management features that allow managers to control access to the application and assign roles and permissions to users.

1.5 Tech Stack and APIs

A React application that uses employee information, location, and fault information to assign jobs automatically would require integration with several APIs and data sources. Here are some APIs and data sources that could be useful for this application:

Employee information: The application would need to collect information about employees, such as their skills, availability, and location. This information could be stored in a database or accessed through an API, such as the Google Maps API or the OpenStreetMap API.

Job information: The application would also need to collect information about the jobs that need to be completed, such as the type of work required and the location of the job site. This information could be stored in a database.

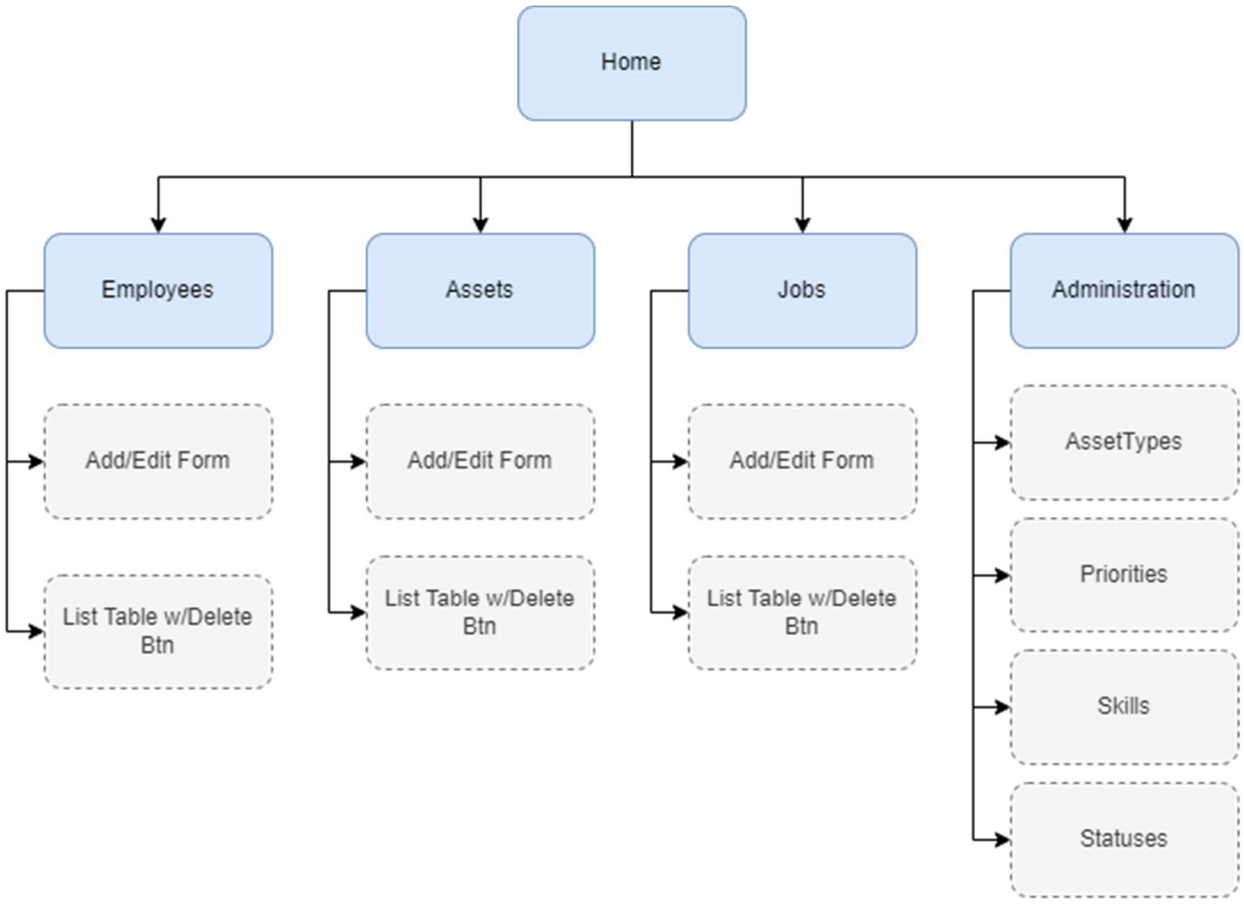
Fault information: The application would need to collect information about faults or issues that need to be addressed. This information could be collected through sensors or other IoT devices or entered manually by employees or customers.

Algorithms: The application would need to use algorithms to automatically assign jobs to the most appropriate employee based on their skills and location. This could involve machine learning algorithms or other optimization techniques.

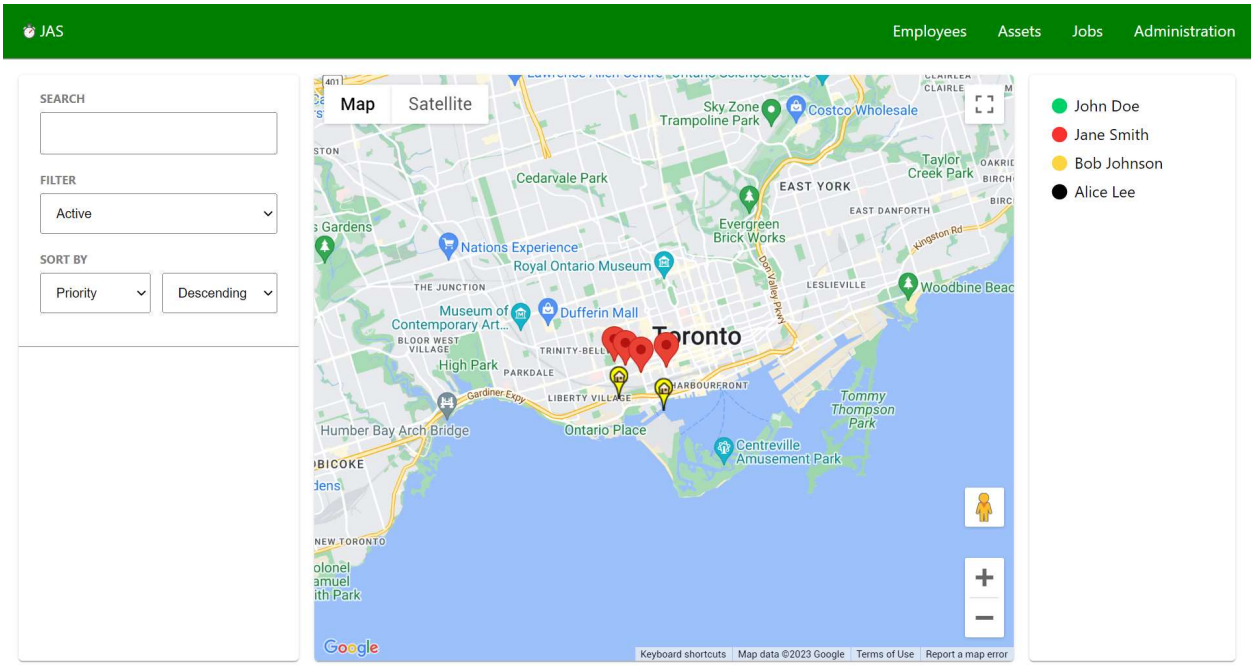
BrainStation®

2. Client-Side Implementation

Site Map



Screen Details



- Graphical User Interface.
- Developed as HTML, CSS and JavaScript.
- Contains forms to submit assignments.
- Provides reports for employees, assignments and assets.
- Displays map of assets and employees.

3. Server-Side Implementation

3.1 End-Point Descriptions

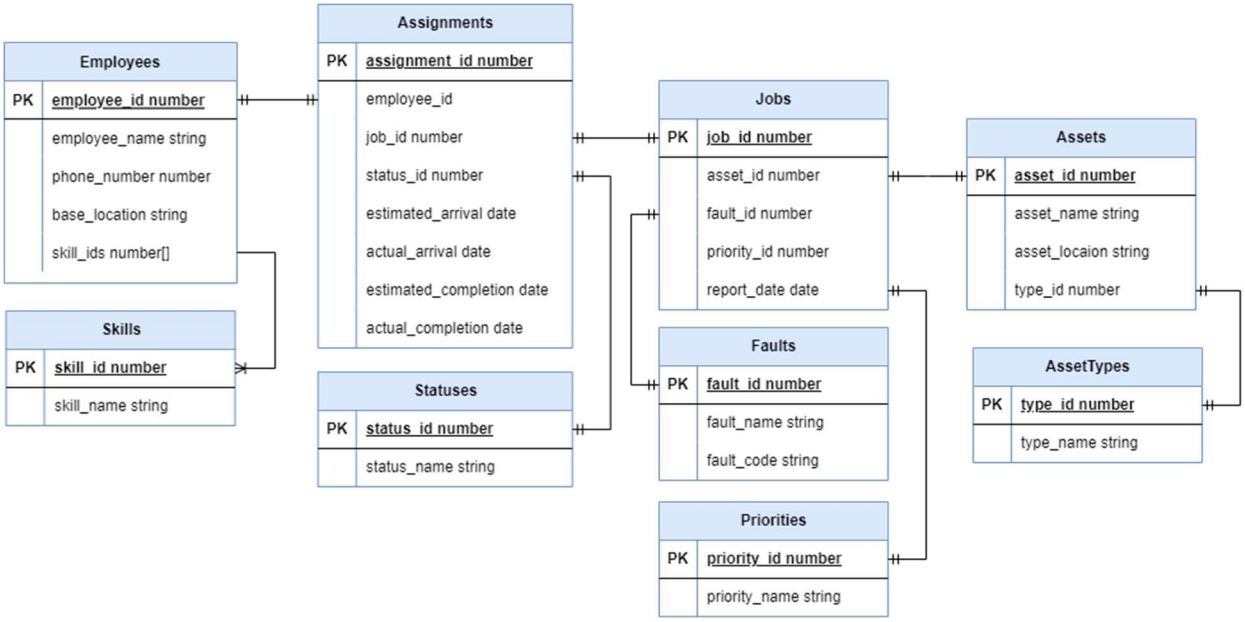
Endpoint	HTTP Methods	Description
/employees	GET, POST	Retrieve list of employees/Add new employee
/employees/:id	GET, PUT, DELETE	Retrieve/Update/Delete employee record
/assets	GET, POST	Retrieve list of assets/Add new asset
/assets/:id	GET, PUT, DELETE	Retrieve/Update/Delete asset record
/jobs	GET, POST	Retrieve list of jobs/Add new job

/jobs/:id	GET, PUT, DELETE	Retrieve/Update/Delete job record
/assignments	GET, POST	Retrieve list of assignments/Add new assignment
/assignments/:id	GET, PUT, DELETE	Retrieve/Update/Delete assignment record
/faults	GET, POST	Retrieve list of faults/Add new fault
/faults/:id	GET, PUT, DELETE	Retrieve/Update/Delete fault record
/priorities	GET, POST	Retrieve list of priorities/Add new priority
/priorities/:id	GET, PUT, DELETE	Retrieve/Update/Delete priority record
/skills	GET, POST	Retrieve list of skills/Add new skill
/skills/:id	GET, PUT, DELETE	Retrieve/Update/Delete skill record
/statuses	GET, POST	Retrieve list of statuses/Add new status
/statuses/:id	GET, PUT, DELETE	Retrieve/Update/Delete status record

3.2 External APIs that will be consumed

Google Maps API: Provides access to Google's mapping and location-based services, including geocoding, directions, and place search.

3.3 Database Structure



3.4 Authentication/Authorization and Security

Authentication and authorization are important considerations for any application that handles sensitive data, such as employee and job information.

- Role-based access control (RBAC): Use RBAC to control access to different parts of the application based on user roles. This would involve defining roles for different types of users, such as managers and employees, and restricting access to certain features based on those roles.

In addition to authentication and authorization, security is also an important consideration for this type of application:

- password policies: Enforce strong password policies, such as minimum length and complexity requirements, to prevent unauthorized access.

BrainStation®

4. Project Roadmap

Phase 1

Milestones	Goal
1	*Create database and design models*
2	*Build server boilerplate and hook up to database*
3	*Define all Endpoints in express, connect endpoints to any external web api*
4	*Implement functionality of endpoints - create JSON response and verify endpoint responses*
5	*Build React App overall structure and high-level components*
6	*Connect smart components to end-point APIs*
7	*Test/debug end-to-end functionality of app*
8	*Implement CSS/Styling*
9	*Code Clean up, last minute testing/debugging*
10	*DEPLOYMENT*
11	*DEMO DAY*

Phase 2

I will work on Mobile App *next after Phase 1 is complete. Because* develop a mobile app version of the application allow managers to assign and track jobs on the go.

Phase 3

As a future enhancement that are more complex, or take the project to a more completed, production-ready state:

1. Chatbot Integration: Integrate a chatbot into the application to allow employees to communicate with managers and receive job solving help through a conversational interface.
2. Advanced Reporting and Analytics: Develop advanced reporting and analytics features to provide deeper insights into job assignment data and help managers make more informed decisions.
3. Machine Learning: Implement machine learning algorithms to improve job assignment accuracy and efficiency.
4. Integration with other systems: Integrate the application with other systems, such as accounting or inventory management, to streamline business processes.

5. Demo Day Information

Please fill out the Google Forms that will be sent out via Slack. This information will be used to complete your presentation and to showcase your profile on the BrainStation website.