

EE / CPR E / S E 492 – [sdmay21-51](#)

Team 51: Constrained Re-Planning in Spatial Crowdsourcing

Bi-Weekly Report 4: 03/01 – 03/15

Client &/Advisor: Goce Trajcevski

Team Members:

- Logan Anderson: Frontend/Test Engineer
- Steven Sheets: Backend/Test Engineer, Report Manager
- Nicholas Heger: Frontend Engineer, Progress Manager
- Jared Weiland: Backend Engineer
- James Volpe: Frontend/Backend Engineer

Past Week Accomplishments:

- Implemented the log in and sign-up pages as components in react. Added UX for both, implemented with bootstrap forms and inputs, with a select element for a dropdown indicating the account type. - Nicholas
- Researched and implemented a system for routing. Added react-router-dom to the project and began using it to route different pages. Added a top navigation bar to the web application that allows navigation between each component. - Nicholas
- Began creating a skills component to display the skills a worker has. Planning to use a list component to display current skills and a modal to add new skills. Started testing GET requests to retrieve current skills for a given worker. - Nicholas
- Rewrote code base for map system due to memory bloat causing system crash. Map can process routes from a static start location to dynamic end locations and display estimated time as well as instructions for route. Started experimenting with Geocoding API, which allows transforming addresses into coordinates. -Logan
- Created Employers and Workers, with their respective repository interface. Created Skill and Location, to be tied to workers. Tested this legitimate mock-user in the database. - Steven

Pending Issues:

- Frontend and backend still to be connected.

Individual Contributions:

| Name | Contributions | Bi-Weekly Hours | Cumulative Hours |
|----------------|--|------------------------|-------------------------|
| Logan Anderson | React, Report, Map | 8 | 26 |
| Steven Sheets | Meeting Organization, React/SB, SB/Mongo, Report | 6 | 25 |
| Nicholas Heger | React (sign in/up pages, routing, skills page), Report | 12 | 30 |
| Jared Weiland | Server and Database, Report | 3 | 19 |
| James Volpe | React Research, Report | 4 | 17 |

Plans for Coming Week:

- Add create account / sign in client-side validation - Nicholas
- Hook up create account page to POST to backend - Nicholas + Steven/Jared
- Hook up sign in page to GET from backend - Nicholas + Steven/Jared (same as above)
- Allow for user entered start location for routes using Geocoder API. Integrate map from independent React application into client side React as a web page. - Logan

Sign-up Page

sdmay21-51 Login Sign up Skills

Sign Up

First name **Last name**

Email address

Password

Account Type

[Already registered sign in?](#)

Worker Document in DB

MongoDB Compass - cluster0.pl2aq.mongodb.net/ames.workers

Connect View Collection Help

Local

- 5 DBS 13 COLLECTIONS
- HOSTS
 - cluster0-shard-00-02.pl2a...
 - cluster0-shard-00-00.pl2a...
 - cluster0-shard-00-01.pl2a...
- CLUSTER
 - Replica Set (atlas-n5srmw-...)
 - 3 Nodes
- EDITION
 - MongoDB 4.4.4 Enterprise

Filter your data

- ames
 - customers
 - employers
 - traffic
 - workers

MongoSH Beta

ames.workers Documents

| DOCUMENTS | TOTAL SIZE | AVG. SIZE | INDEXES | TOTAL SIZE | AVG. SIZE |
|-----------|------------|-----------|---------|------------|-----------|
| 1 | 245B | 245B | 1 | 36.0KB | 36.0KB |

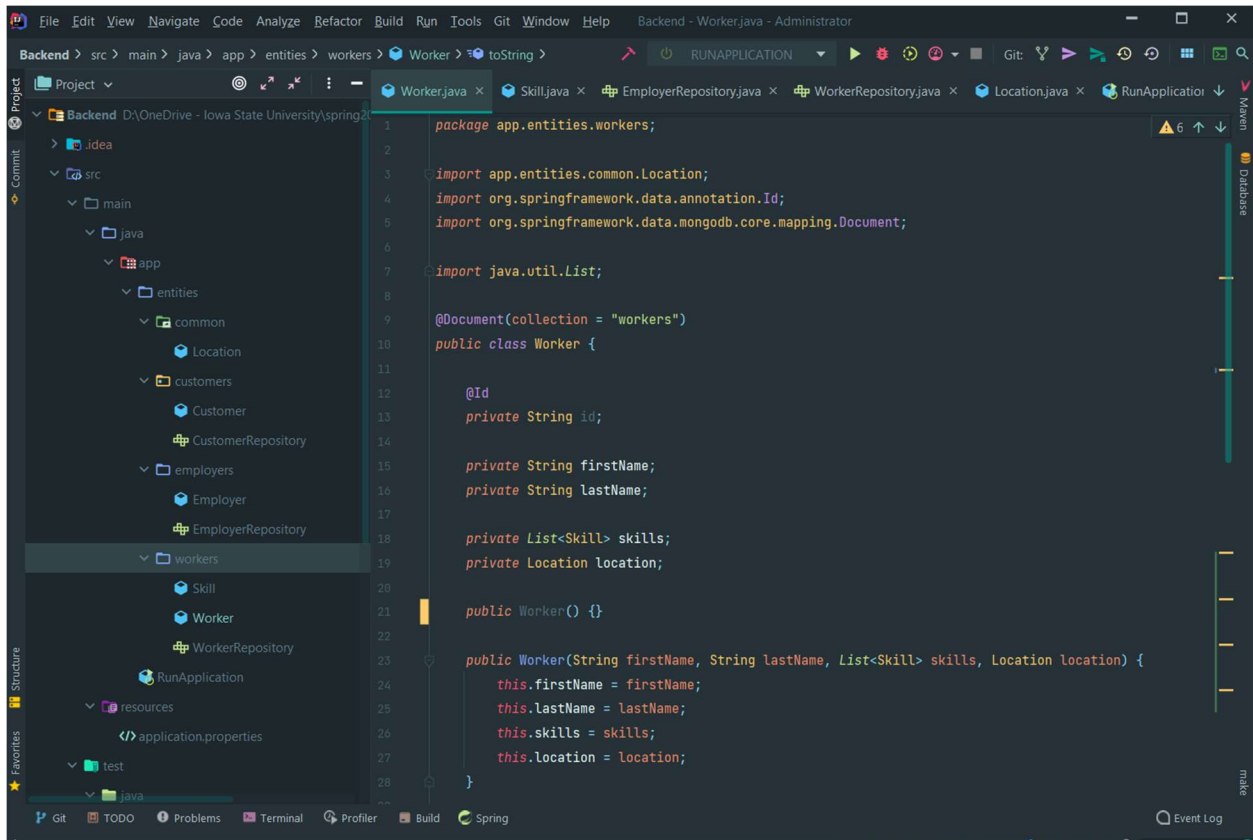
Documents Aggregations Schema Explain Plan Indexes Validate

FILTER { field: 'value' } OPTIONS FIND RESET

ADD DATA VIEW Refresh

```
{
  "_id": ObjectId("604834a95f3ee7023cac9d82")
  "firstName": "Steven"
  "lastName": "Sheets"
  "skills": Array
  > 0: Object
  > 1: Object
  "location": Object
    "longitude": 42.02216
    "latitude": -93.642873
  "_class": "app.entities.workers.Worker"
}
```

Backend, General File Structure



The image shows an IDE window with a project structure on the left and a code editor in the center. The project structure is as follows:

- Backend > src > main > java > app
 - entities
 - common
 - Location
 - customers
 - Customer
 - CustomerRepository
 - employers
 - Employer
 - EmployerRepository
 - workers
 - Skill
 - Worker
 - WorkerRepository
 - RunApplication
 - resources
 - application.properties
 - test > java

The code editor shows the following Java code for `Worker.java`:

```
1 package app.entities.workers;
2
3 import app.entities.common.Location;
4 import org.springframework.data.annotation.Id;
5 import org.springframework.data.mongodb.core.mapping.Document;
6
7 import java.util.List;
8
9 @Document(collection = "workers")
10 public class Worker {
11
12     @Id
13     private String id;
14
15     private String firstName;
16     private String lastName;
17
18     private List<Skill> skills;
19     private Location location;
20
21     public Worker() {}
22
23     public Worker(String firstName, String lastName, List<Skill> skills, Location location) {
24         this.firstName = firstName;
25         this.lastName = lastName;
26         this.skills = skills;
27         this.location = location;
28     }
29 }
```