

# Constrained Re-Planning in Spatial Crowdsourcing, PIRM II

Team 51

Steven Sheets (Backend Engineer, Test Engineer)  
Logan Anderson (Frontend Engineer, Test Engineer)  
Nicholas Heger (Frontend Engineer, Progress Manager)  
Jared Weiland (Backend Engineer)  
Jame Volpe (Frontend/Backend Engineer)

Advisor: Goce Trajcevski

[sdmay21-51@iastate.edu](mailto:sdmay21-51@iastate.edu)

# Project Recap - I

## Project Goal:

Create a spatial crowdsourcing algorithm that runs on a mobile and web application that match workers with tasks from consumers.



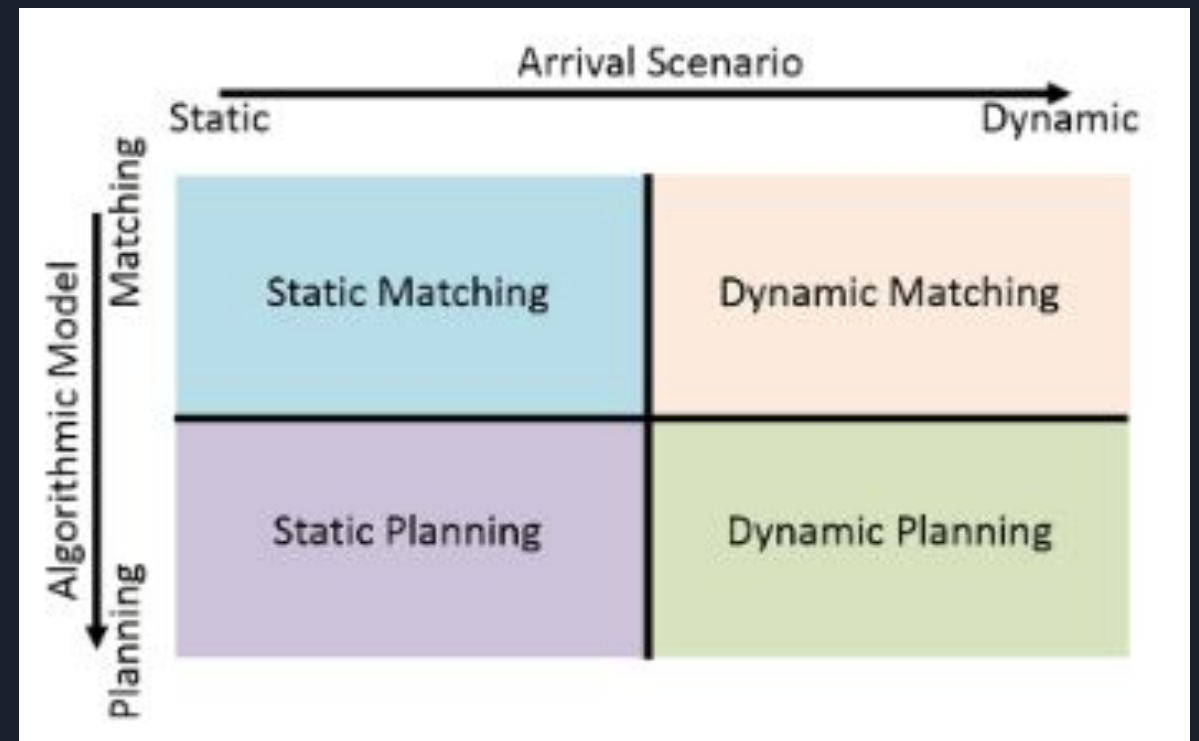
# Project Recap - II

## 4 types of spatial algorithms

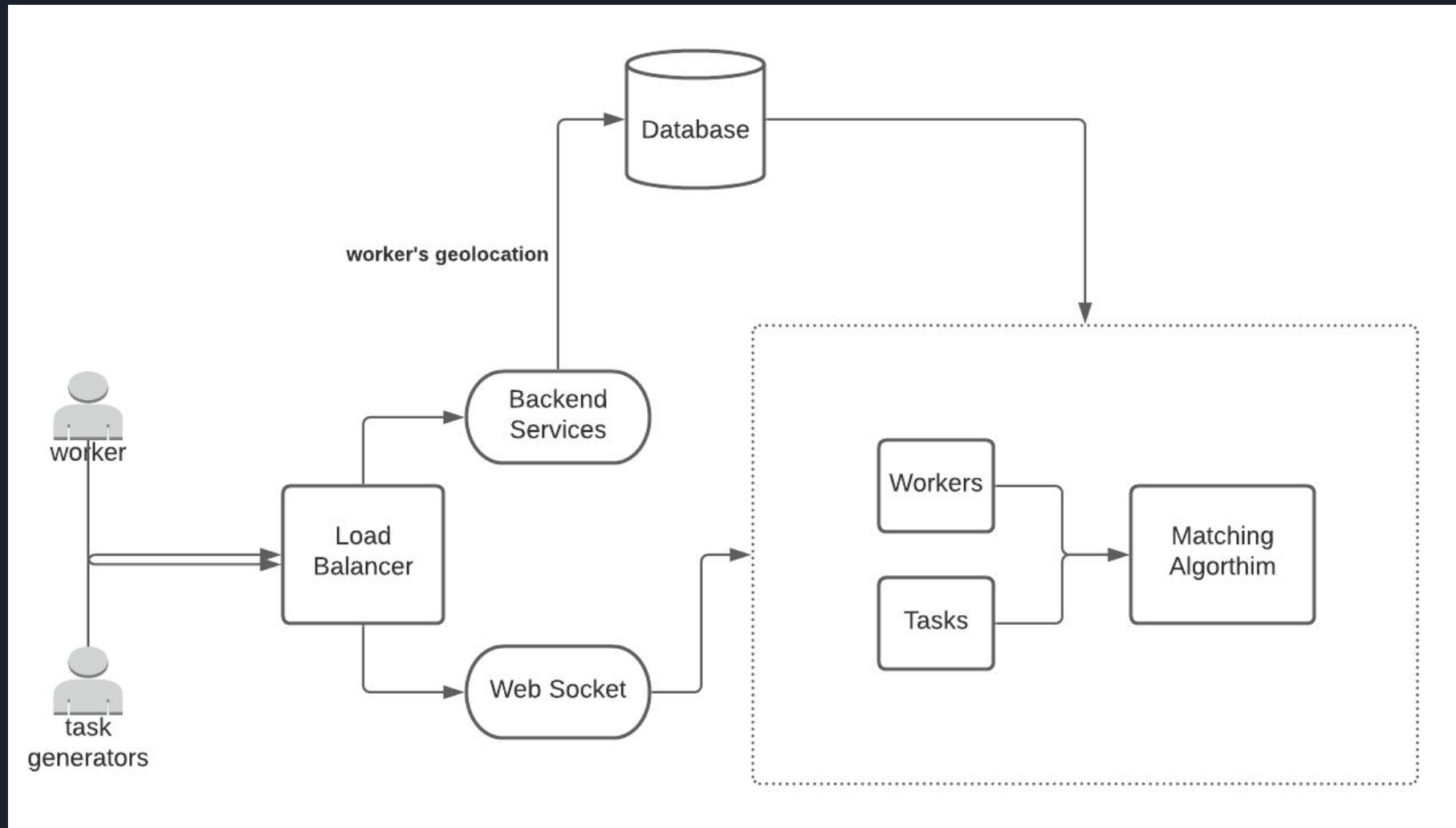
- Static Matching
- Static Planning
- Dynamic Matching
- Dynamic Planning

## Current Implementation plan

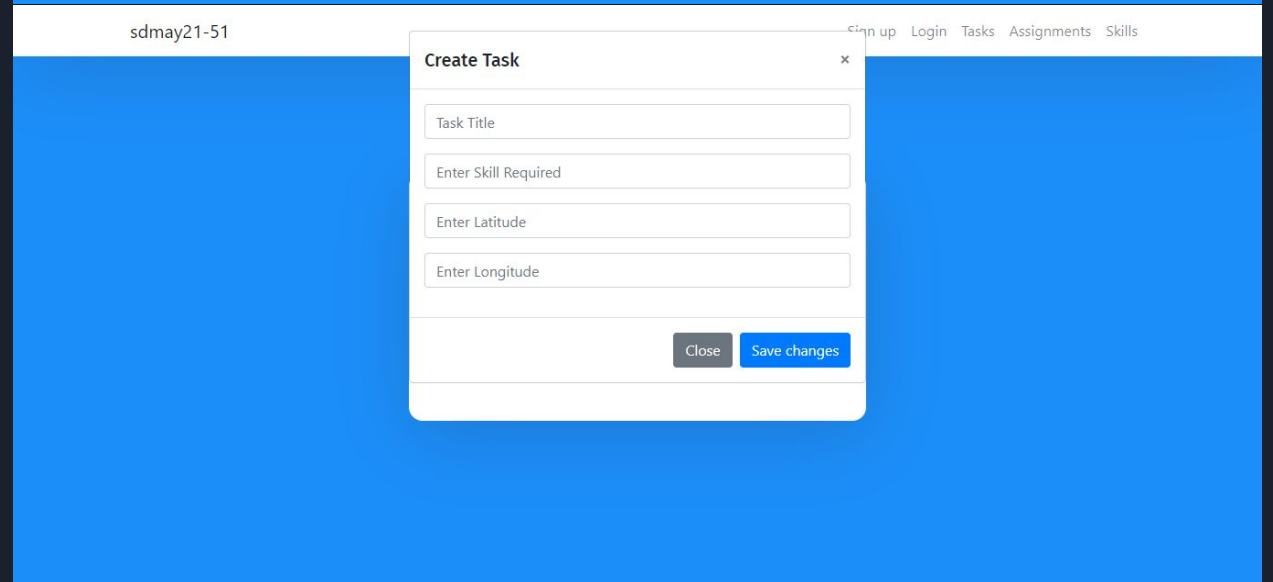
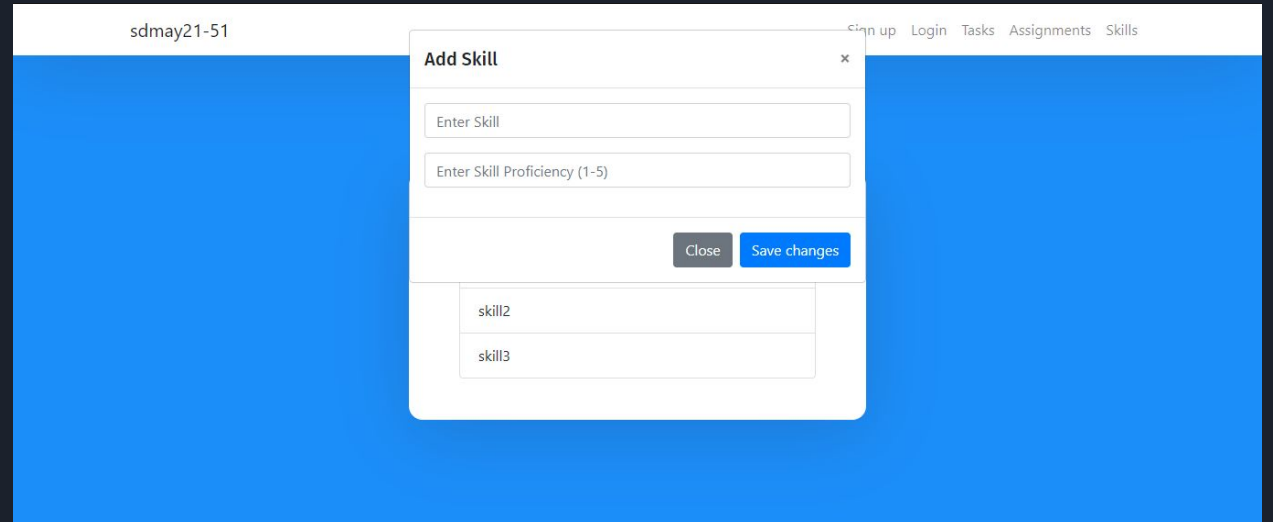
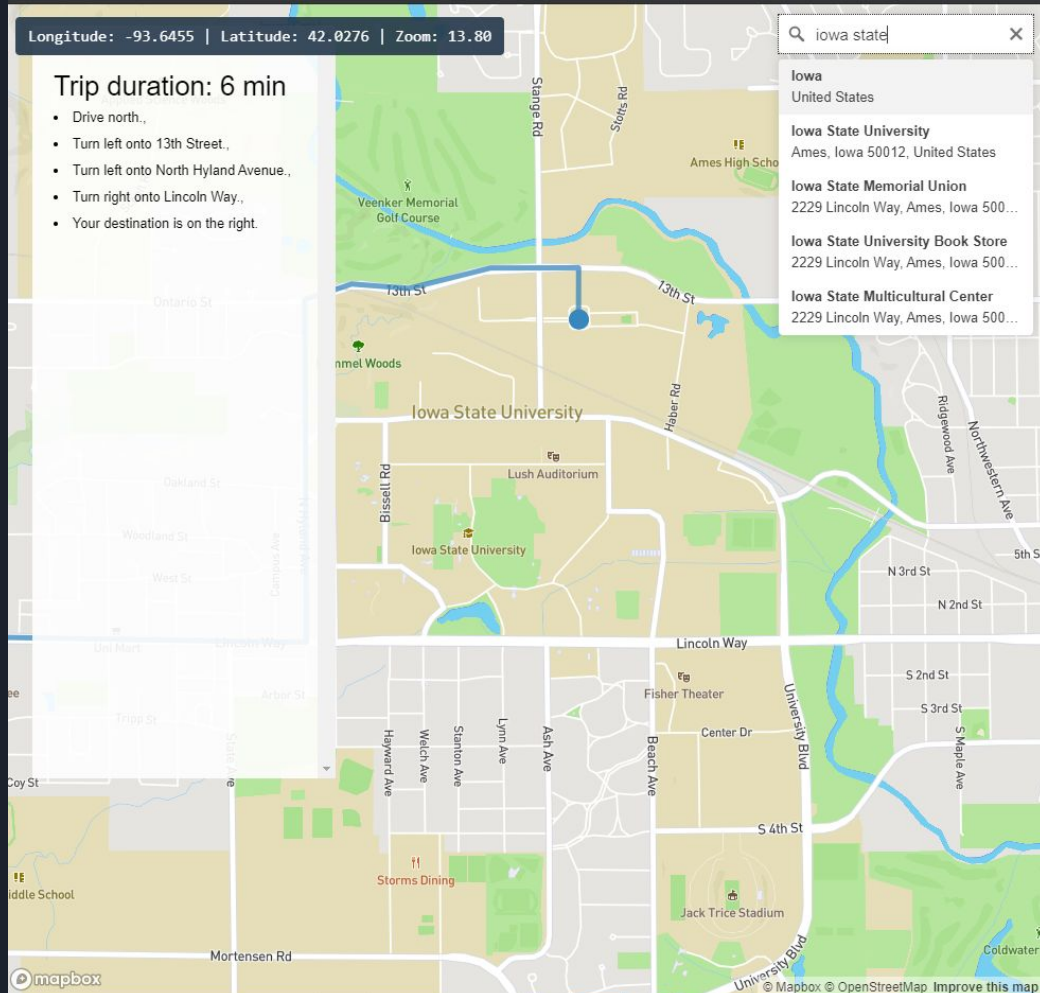
- Initial: Static Matching
- Final: Dynamic Planning



# System Design - Component Diagram



# Current Screenshots of Desktop App



# Engineering Standards

- [IEEE/ISO/IEC 29119-2-2013 - ISO/IEC/IEEE International Standard - Software and systems engineering – Software testing – Part 2: Test processes](#)
- [IEEE/ISO/IEC 29119-3-2013 - ISO/IEC/IEEE International Standard - Software and systems engineering – Software testing – Part 3: Test documentation](#)
- [29119-4-2015 - ISO/IEC/IEEE International Standard - Software and system Engineering -- Software testing --Part 4: Test techniques](#)

# Engineering Constraints

- Must run as a mobile and desktop app
- Server needs to be able to handle algorithm processing
- Application requires internet connection
- Free Mapbox API
  - 50,000 monthly map loads
  - 100,000 monthly direction requests
  - 100,000 monthly geocoding requests
- Project must work without a budget
- Project must be completed within the semester

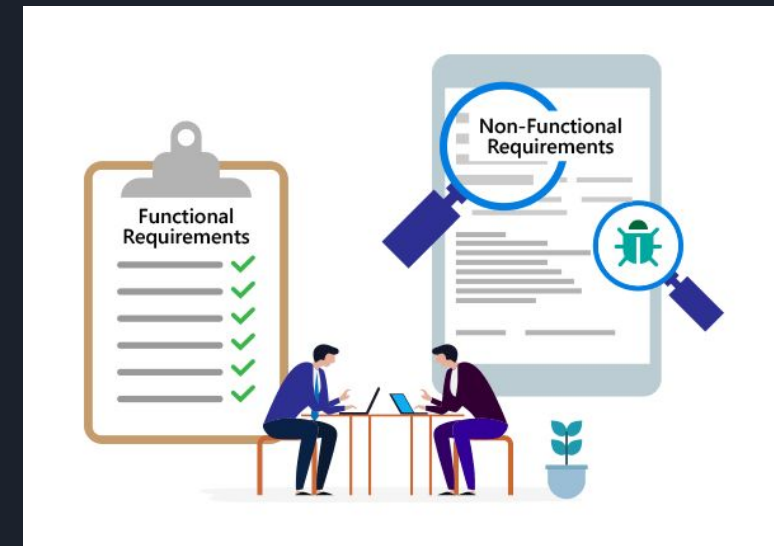
# Functional Requirements

- Allow employers and workers to be able to create accounts (stored in DB)
- Take input of workers: skills, location, and reputation
- Take input of tasks: location(s) and skills required
- Optimize a schedule based on task and worker input
- Alert workers of incoming tasks
- Re-optimize schedule in the event of new constraints
- User interface for visualization of work schedule



# Non-Functional Requirements

- Reliability - few bugs or issues that impede user experience
- Performance - algorithm is polynomial time and app is optimized for web/mobile
- Scalability - able to be used by a large number of users simultaneously
- Maintainability - readable code with documentation
- Usability - intuitive/easy to use
- Modularity - be able to switch between components (like algorithm used)

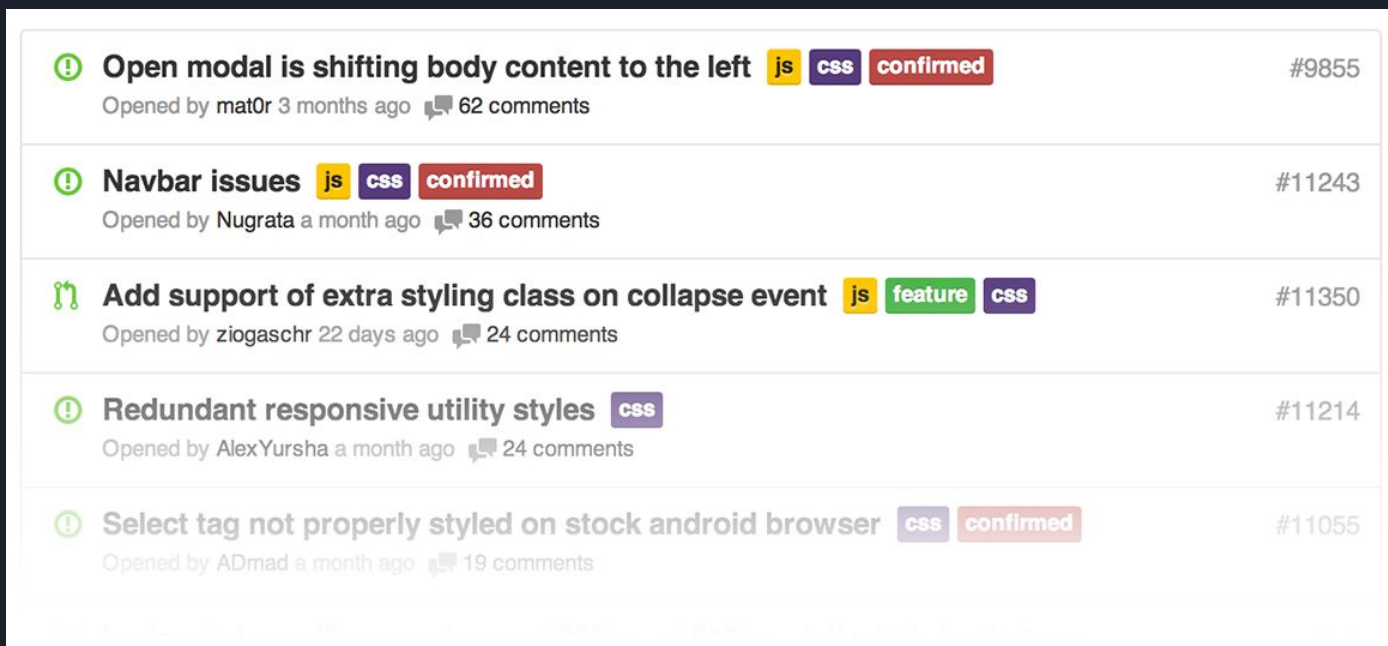


# Technical Challenges

- Implementing spatial crowdsourcing algorithm
- Establishing a server
- Frontend-backend communication/familiarizing ourselves with Apollo GraphQL
- Familiarization with Mapbox API
- Working without in-person team interaction for much of the project
- Makes sure there is not memory bloat in the client

# Immediate Next Steps

- Expand frontend-backend communication
- Work on implementing alternate spatial crowdsourcing algorithms
- Run our backend on a server and set up CI/CD.



A screenshot of a GitHub issues page showing five open issues. Each issue entry includes a green circular icon with an exclamation mark, the issue title, a list of labels (js, css, confirmed, feature), the issue number, the author's name, the time since it was opened, and the number of comments.

Issue Title	Labels	Issue Number	Author	Time	Comments
Open modal is shifting body content to the left	js, css, confirmed	#9855	mat0r	3 months ago	62
Navbar issues	js, css, confirmed	#11243	Nugrata	a month ago	36
Add support of extra styling class on collapse event	js, feature, css	#11350	ziogaschr	22 days ago	24
Redundant responsive utility styles	css	#11214	AlexYursha	a month ago	24
Select tag not properly styled on stock android browser	css, confirmed	#11055	ADmad	a month ago	19





# Thank you!

Any questions, concerns, or  
comments are greatly appreciated!  
Looking forward to feedback!