

# Constrained Re-Planning in Spatial Crowdsourcing

Team 51:

Logan Anderson, Nicholas Heger, Steven Sheets, James Volpe, Jared Weiland  
Client/Advisor: Goce Trajcevski

## Introduction

### Spacial Crowdsourcing:

A matching problem where there are:  
1. A set of workers, each with skills and a geolocation  
2. A set of job-sites with tasks requiring specific skills at a location

### Solution

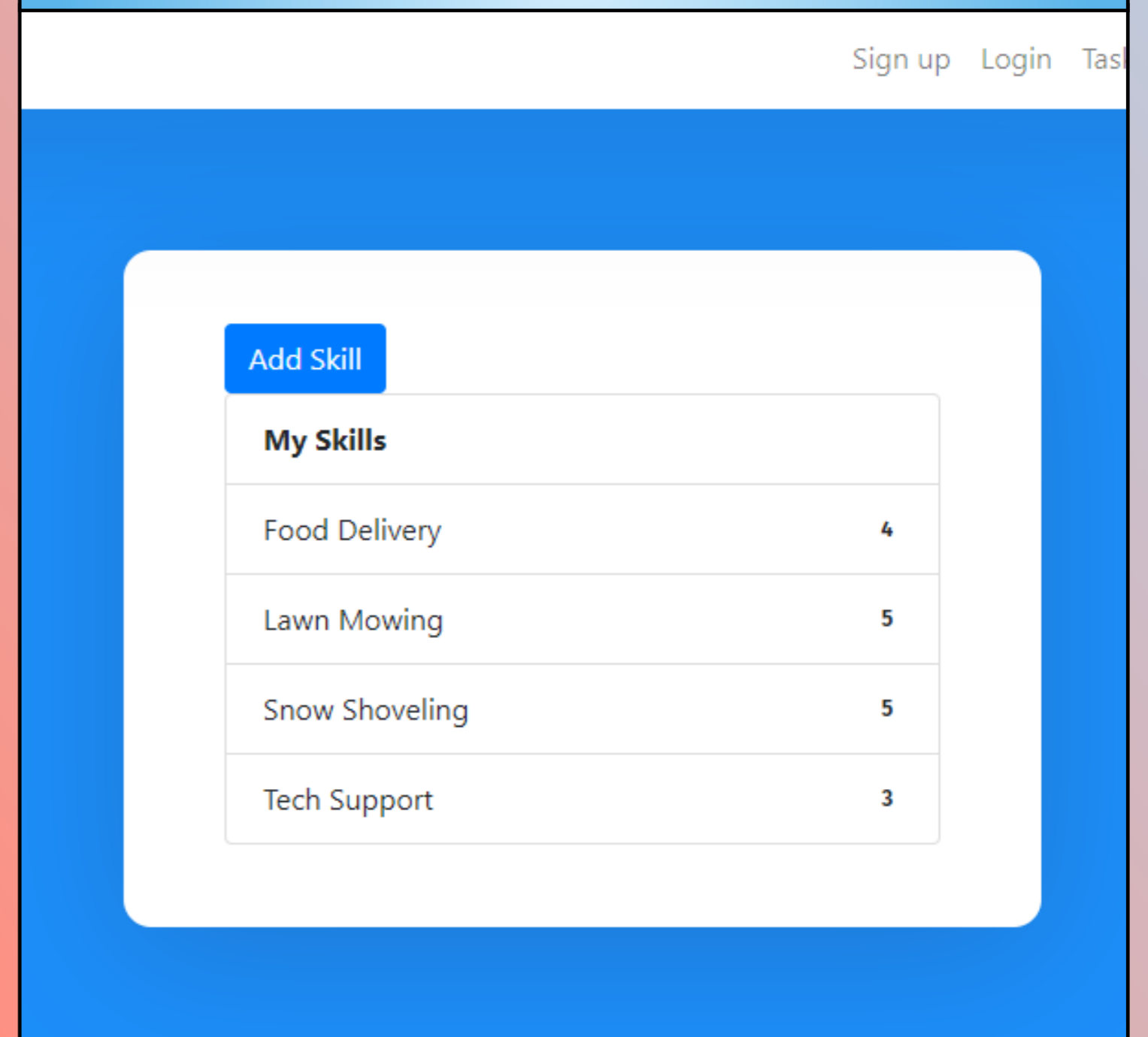
A client/server application that matches works with tasks.

## Intended Use

**Use:** With a web and mobile application, our end products are very versatile. Their intended use is for services similar to Uber or GrubHub.

**Users:** The end products are intended to be used by customers who will create tasks, as well as workers who will complete the tasks submitted by the customers.

## Image of Skills Page



## Testing

Testing Environment

Frontend  
• Jest  
• Enzyme  
Backend  
• JUnit

Testing Strategies

Frontend  
• Unit testing  
• Component testing  
Backend  
• Unit tests  
• CIT

## Technical Details

Technology Used:

- Javascript, HTML
- ReactJS, Bootstrap
- Apollo, GraphQL
- Springboot, Java
- MongoDB, Mapbox
- IntelliJ, Gitlab
- Jest, Enzyme

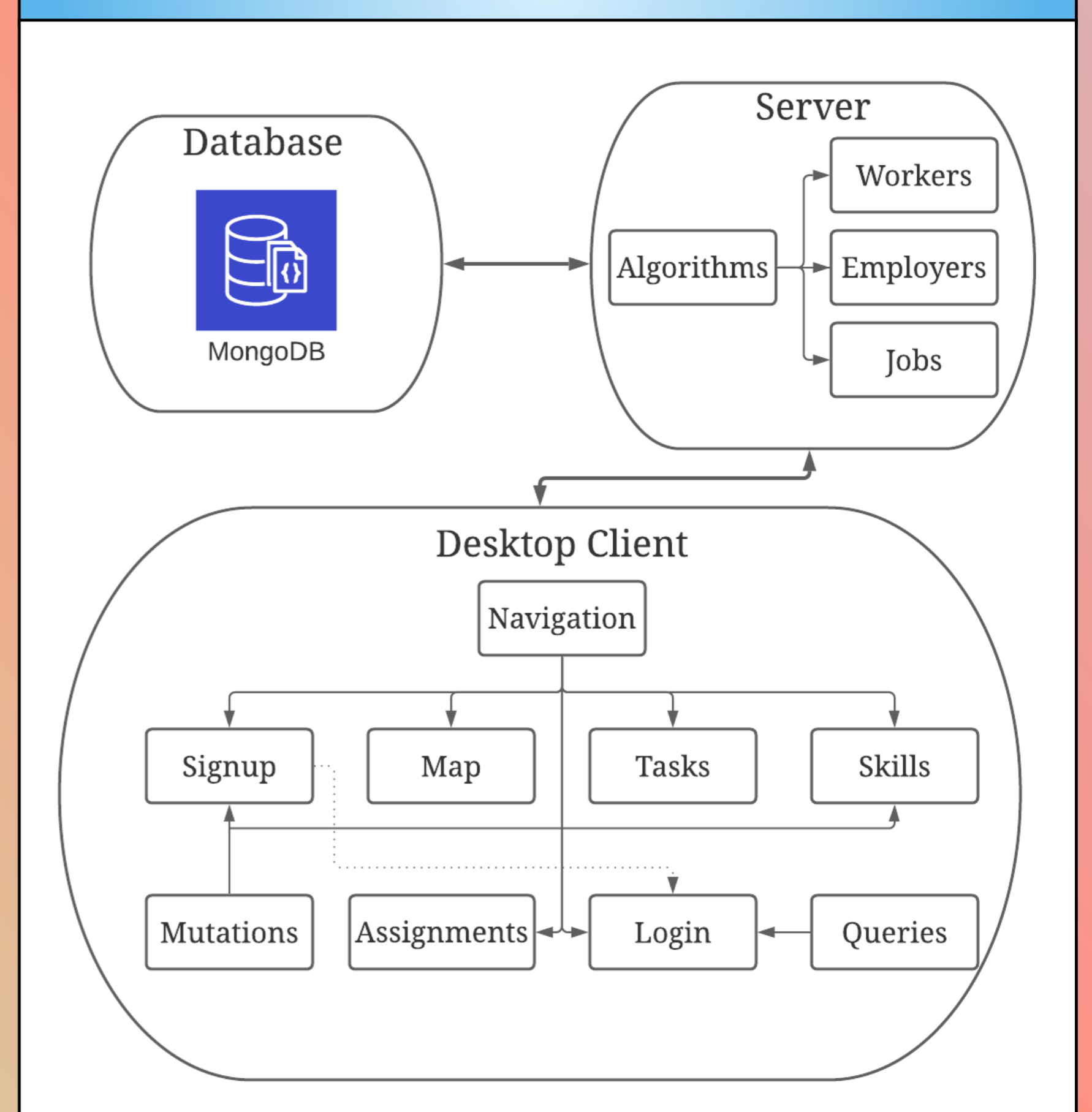
## Functional Requirements

- Allow customers and workers to be able to create accounts
- Take worker inputs of skills and location
- Take task inputs of skills required and location
- Optimize a schedule based on worker and task inputs
- Re-optimize this schedule in the event of new information
- Alert workers of tasks to complete
- Web UI for the addition of tasks and visualization of work schedule

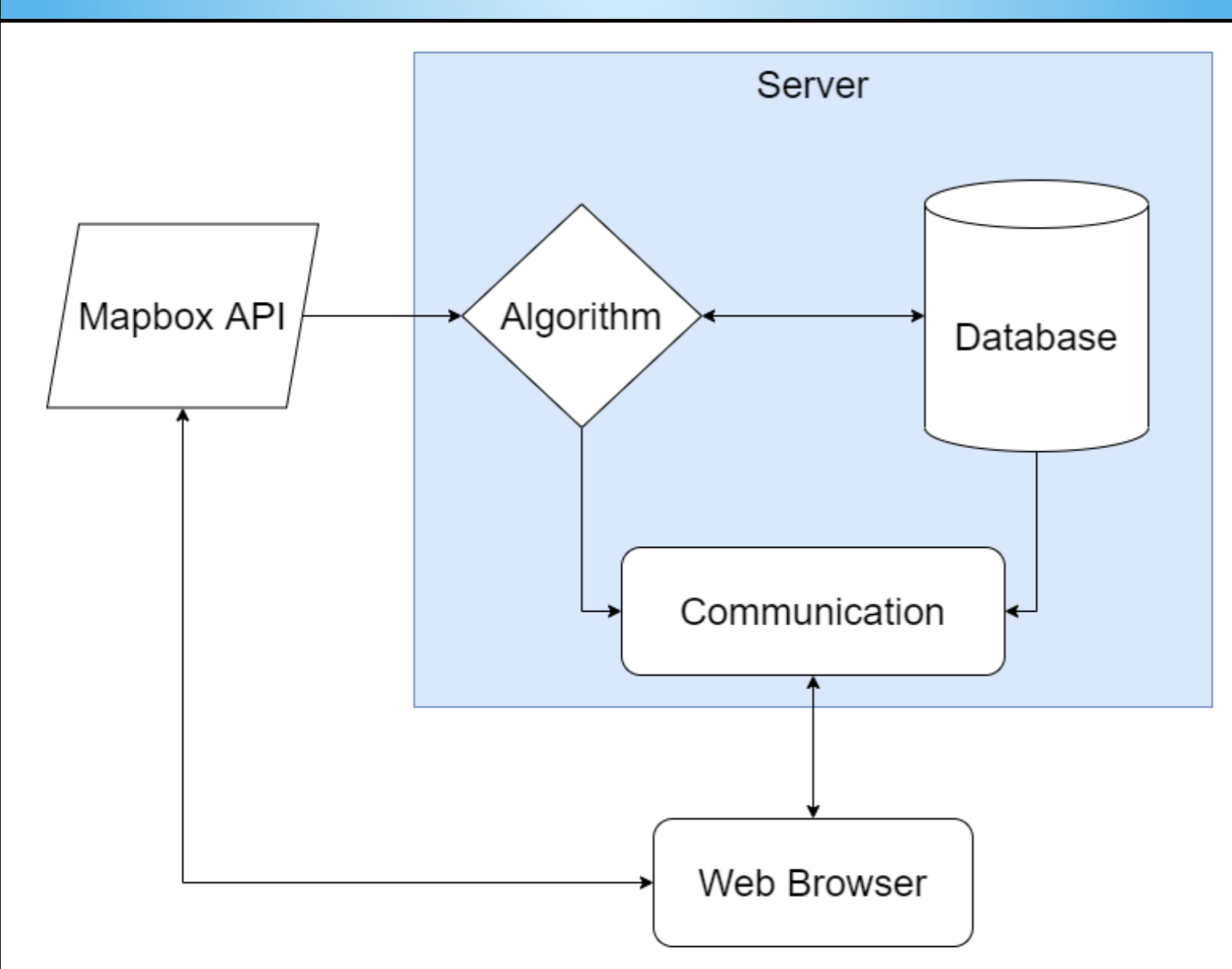
## Design Approach

- Client/server based
- Server contains algorithm module and database
- Client is web browser
  - UI for worker and employer
  - Requests required information from server

## Block Diagram



## Concept Design



## Non-Functional Requirements

- Function with few bugs or issues that impede the users' experience
- Protect users' personal information from others
- Optimized applications to run efficiently on mobile devices
- Be able to be used by a large number of users at one time

## 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 load
  - 100,000 monthly direction requests
  - 100,000 monthly geocoding requests
- Project must work without a budget
- Project must be completed within the semester

## Standards

- IEEE/ISO/IEC 29919-2-2013 - ISO/IEC/IEEE International Standard - Software and systems engineering - Software testing
  - Part 2: Test processes
  - Part 3: Test documentaion
  - Part 4: Test techniques

## Image of Map page

