# Dynamic Reallocation of Trucks

sdmay22-32(Spring 2022)

Client & Advisor: Dr. Goce Trajcevski

Team Members: Nolan Slimp, Joshua Heroldt, Indrajeet Aditya Roy, Bernard Fay, Asma Gesalla, Matt Medley

## Introduction

- Problem
  Delivery services have many unpredictable issues, such as a truck breaking during a delivery.
- Solution
  Implement a web application using MapBox API that routes trucks to multiple destinations, with the added constraint that a truck will break at some point.

## Intended Users and Uses

- The primary users of our project are customers, truck dispatchers, and truck drivers.
- The project will be more directly used by dispatchers as it will aid them in deciding initial routes for trucks as well as making decisions and adjustments in the case of changing circumstances (traffic, new orders, truck breakdowns, etc.).

## Design Requirements

Functional requirements:

- Trucks should pick-up locations.
- Trucks should be able to deliver to the picked location.
- Find the nearest truck in case of any breaks.
- Find the optimal truck to take care of the broken truck
- Set of orders and trucks w/ given capacity.

Non-functional requirements:

- Application should have access to all truck info databases.
- Application should have enough information about all locations, type of loads and capacity.

Constraints:

- Response time (<1 min to react to dynamic events)
- Assuming the availability of road network maps and other traffic distribution data (traffic density) -> Needed for any assignment (both initial and dynamic)
- Economics:
- Minimize delivery delay as a result of a dynamic update
- Minimize idle time of trucks

Operating Environment: Windows, Mac, & Linux

Relevant Standards:

- Scrum methodologies
- IEEE 610.12, Standard Glossary of SE Terminology
- IEEE 1540: Software Risk Management

## Design Approach

**Application flow:**

- Place order in the frontend (Calls API to change address to lat,long)
- View order page (Frontend calls backend to get the order database to display the orders)
- Go to visualization page for a specific order.(Visualization page calls the backend to get the number of trucks and the lat, long coordinates per truck for the stops on its route).
- Routes are calculated and displayed on the visualization page.
- Once every route is calculated they are put into a single json and sent to the backend.
- Backend calculates and returns locations of all trucks including the broken truck.
- Frontend recalls the optimization and navigation api using a dummy route, including the location of the broken truck in order to find out which routes are optimal (also using load).
- Routes are then updated and displayed on the visualization page.

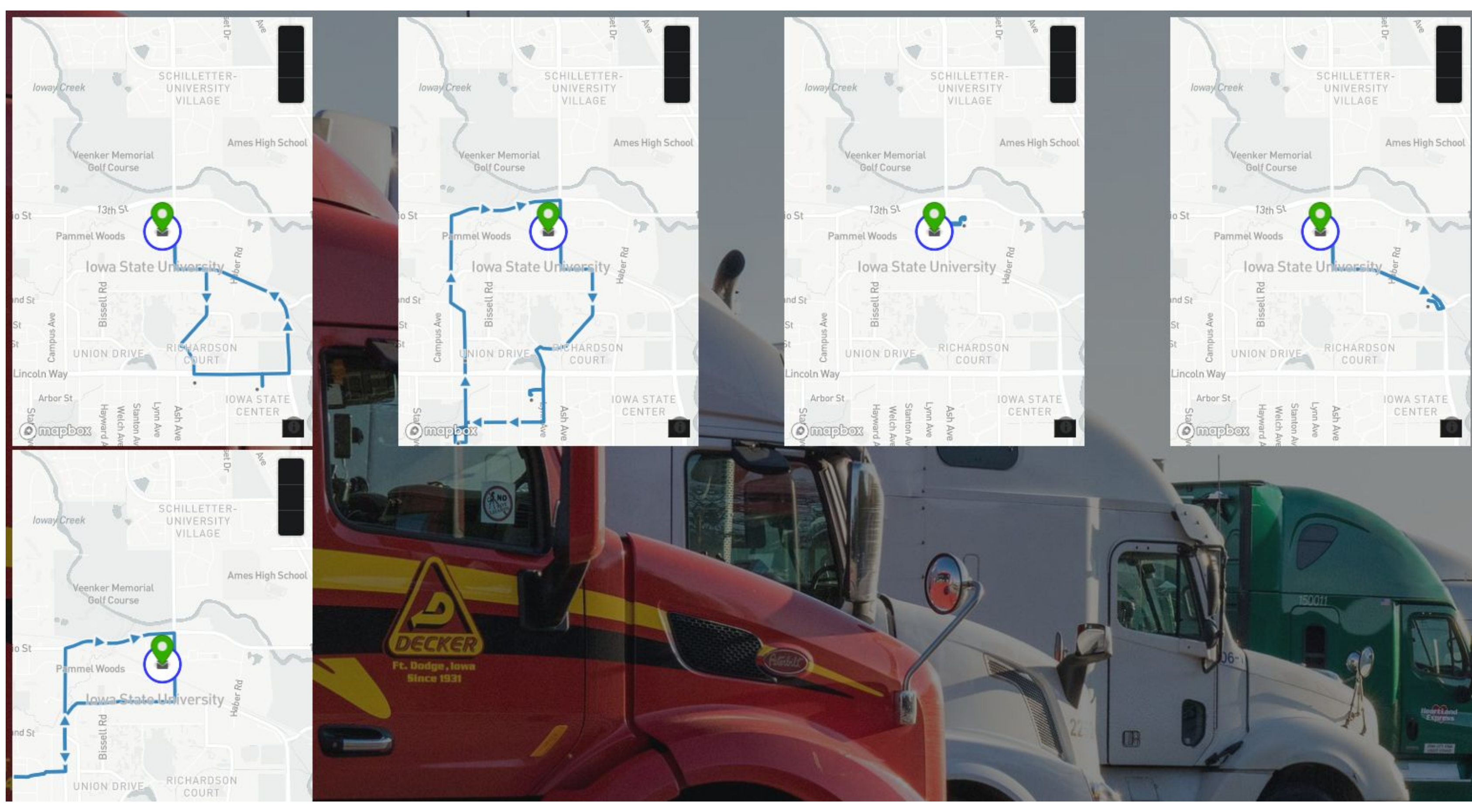
## Testing

- Postman collections – A suite of tests and API calls
- JUnit unit tests
- Interface and System Testing
- Acceptance Testing

All tests succeeded and our client was satisfied with the results of our project.

## Technical Details

- Frontend:
  - Web App UI: Angular
- Backend:
  - API: Spring Boot
- Programming Languages:
  - Typescript, Java
- Libraries: PrimeNG
- External API : Mapbox
  - Enables our project to route vehicles in any city, taking into account closed roads and traffic density
  - Can use multiple vehicles with starting and ending locations
  - Endpoints for viewing map and routes in our web application
  - Also allows for geocoding of addresses that are entered into the UI from the customer.


Iowa State University
SCHILLETTER-UNIVERSITY VILLAGE
Ames High School
RICHARDSON COURT
UNION DRIVE
IOWA STATE CENTER
DECKER