

Paratransit Fleet Scheduler

Members: Wes Holman, Ant Macchia, Kieran Walsh, Tyler Rambo, Raymond Pickett, Dan Russo

Scrum Master: Tyler Rambo

Team Leader: Ant Macchia

Github: <https://github.com/amacchia/ParatransitFleetScheduler>

Slack: senproject-spring19.slack.com

Project summary:

We will design a paratransit ridesharing service to augment existing public transit resources. Patron requests will be serviced through a variable fleet of vehicles. Patrons will be able to request a ride from one specific address to another specific address at a specific time, and drivers will receive generated schedules at the beginning of their shift through our website interface. These routes will be optimized for minimizing distance travelled, while maximizing patrons served.

Project goals:

- Reduce carbon emissions and transportation costs.
- Increase mobility and independence for underserved populations.
- Implement a static algorithm that will create efficient pickup and drop off routes for drivers.

Project features:

- An easy to use website interface for users to input requests. Passengers input their pick-up location, destination, and the date and time into the website. Passengers must be logged in to add a request.
- A website to display generated routes to drivers. These routes will be shown in Bing maps. Drivers must be logged in to view their routes.
- A database for storing and processing passenger and driver accounts, requests, locations, routes, and driver schedules.
- A scalable backend application for efficiently generating routes over large numbers of patrons and fleet drivers
- Our algorithm generates clusters of locations that are close to each other. The number of clusters is equal to the number of drivers. Our algorithm uses K-means clustering.

Limitations:

- The predetermined nature of the ride requests might disenfranchise users who have urgent transportation needs.
- Traffic, construction, and unforeseen circumstances may cause delays for patrons.
- We may not have sufficient infrastructure to support an inordinate volume of patrons.
- We cannot guarantee timely rides.
- Our algorithm assumes that each vehicle can only carry one passenger and two drivers at most.

Stretch Goals/unimplemented features:

- Include the ability to handle on-the-fly dynamic requests, in addition to planned ones, using an efficient insertion algorithm
- Expand our booking system by allowing patrons to call in to our operators, who will schedule the request
- Live tracking of drivers for waiting patrons
- iOS/Android app