

# Car Pool Match

Damian Davila  
@damianpdavila on GitHub

# Description

Unlike public schools, parochial and private schools draw students from miles away. Car pools are a great way to help families cope with the travel and expense. But how do you find someone to pool with?

Car Pool Match enables families to find nearby car pool “matches”, that fit their individual needs and ensure personal privacy.



# Features

- Registration/Login: Users can create accounts and log in to the application. Each user has a profile page. Users will specify their match criteria/preferences in their profile.
- Match: Matches are based on weighted criteria. Matched profiles are displayed in sorted order as a list and also as pins on a map.
- Contact: User is able to contact the matched profile via email through the app, without direct access to the matched profile's email (for privacy reasons).



# Planning - User Stories

- As a user I want to specify my match criteria so I can find good matches
- As a user, I would like to see potential matches displayed on a map so I can quickly narrow the list of potential matches.
- As a user I want to view the profile of a potential match, so I can decide if they are a good fit for car pool partner
- As a user I want to contact/would like to be contacted by other users so we can discuss potential car pooling, while ensuring my privacy.



# Planning - Database

Persisted tables:

User – used for registration, authentication, and contains profile data

MatchCriteria – contains the user's match criteria/preferences

DaysAvailable – contains DaysAvailable map from MatchCriteria class

Other models:

MatchedUser, RideType (enum), MapConnector, several Forms



# Technology Stack

- Java v8
- Spring Boot v2
- ThymeLeaf templating
- Hibernate, MySQL
- Google Maps API
- Bootstrap, Javascript/JQuery



# Demo



# What I Learned

- Spring Boot is an iceberg in a very dark ocean
- How to use Spring Security (mostly) and Google Static Maps
- I learned how to calculate the distance between two street addresses using geocoding and interesting geometric equations
- Actually built and used a Comparator (no, really)





# What's Next

- Notifying all users of a potential match upon new registration
- Deploy
- Add “forgot password” and un-register features
- Implement unit tests
- Add OAuth registration using Google acct
- Complete the site: home page, T&C, privacy

