

- Communicate from the application server with the Route table of the database after performing route generation
- Completely deploy and run all project components on our AWS instance
- Display routes to drivers upon sign-in on the driver page
- Read from the Database's Schedule table to determine amount of drivers for the given day
- Pull in Requests from the database in time chunks for a given day, and unite their paths together after calculation
- Set the Java application on a fixed daily schedule for Route calculation
- Implement an all-pairwise average distance metric to score clusters on
- Implement a scalable pathfinding algorithm that provides close-to-minimal results
- Include page redirects to login page after password validation
- Add styling to driver page
- Add styling to passenger page
- Sign out
- Add screenshots to manual
- Allow drivers to input a schedule of days they can work
- Restrict input for user requests for sensible values with a specified granularity
- Work around Bing Maps API limit of 25 waypoints
- Deploy front end to AWS server
- Only allow users to request rides in time slices