UniGo

Ride Fare Calculation Algorithm

Objective

To calculate the fare for a ride, taking into account the total distance and time, and the additional distance and time required for deviation to pick up the passenger.

Algorithm

1. Define the following fixed rates:
    - baseKmRate: rate per unit distance (e.g., per kilometer)
    - baseMinRate: rate per unit time (e.g., per minute)
    - addKmRate: rate per unit distance for deviation (e.g., per kilometer)
    - addMinRate: rate per unit time for deviation (e.g., per minute)
2. Calculate the fare for the ride based on the total distance and time:
    - Fare = (baseKmRate * totalDistance) + (baseMinRate * totalTime)
3. Calculate the additional fee for the deviation required to pick up the passenger:
    - Additional fee = (addKmRate * additionalDistance) + (addMinRate * additionalTime)
4. Add the additional fee to the total fare to get the final fare for the ride:
    - Final fare = Fare + Additional fee
5. Display the calculated fare to the user (driver) through the application interface, and allow the user to confirm the ride.

Implementation

```
const baseKmRate = 1;
const baseMinRate = 0.1;
const addKmRate = 2;
const addMinRate = 0.2;

// Function to calculate fare and additional fee
const updateCalculation = (s1, d1, s2, d2, trip) => {
    // ... (fetch the distances and durations using Google Maps DistanceMatrixService)

    // Calculate fare
    var fare = ((originalDistance * baseKmRate) / 1000) + ((oldDuration * baseMinRate) / 60);

    // Calculate additional fee
    var addFare = (((newDistance - originalDistance) * baseKmRate) / 1000) + (((newDuration - oldDuration) * baseMinRate) / 60);
    if (addFare <= 0) {
        addFare = 1;
    }

    // Store the calculated data
    setCalculationData({ pickUpLocation, dropOffLocation, pickUpLocation, newDistance, newDuration, fare, addFare, newDurationMin, newDurationSec });
};
```