Low Level System Design - ride_sharing

Problem Statement

Develop a ride-sharing application with the following features.

Features:

- User Roles: Users can either offer a shared ride (Driver) or consume a shared ride (Passenger).
- Ride Selection: Users can search and select from multiple available rides on a route with the same source and destination.

Requirements:

- User Management:

  Onboarding: Implement functionality to add user details add_user(user_detail)

- Vehicle Management:

  Implement functionality to add vehicle details for users.

  add_vehicle(vehicle_detail)

- Ride Offering:

  Offer Ride: Allow users to offer a shared ride on a route with specific details.

  offer_ride(ride_detail)

- Ride Selection:

  Select Ride: Users can select a ride from multiple offered rides using a selection strategy based on preferred vehicle or most vacant seats select_ride(source, destination, seats, selection_strategy)

- Ride Management:

  End Ride: Implement functionality to end a ride. Users can offer a ride for a given vehicle only when there are no active offered rides for that vehicle.

  end_ride(ride_details)

- Statistics:

  Print Ride Stats: Retrieve and display total rides offered/taken by all users.

  print_ride_stats()

Bonus Question:

- Multiple Rides Output: If a direct route is not available, output multiple rides that can facilitate the journey.

Other Notes:

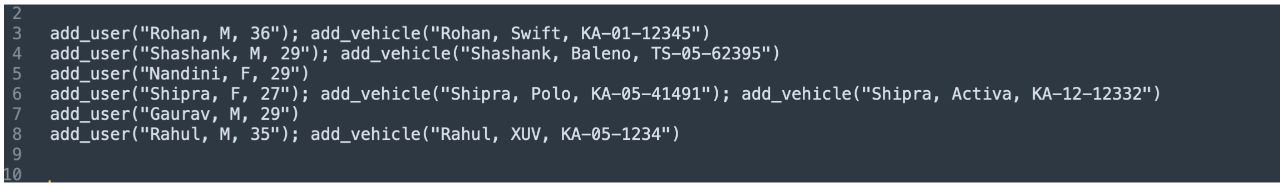
- Develop a single class for demo purposes to execute all commands and test cases in one place within the code.
- Use in-memory data structures instead of databases or NoSQL stores.
- Avoid creating any user interface for the application.
- Prioritize code compilation, execution, and completion.
- Focus on achieving expected output first before adding additional features.

Expectations:

- Preferred languages for the interview question are Ruby, Golang, Node.js, and Python
- Ensure the code is functional and demonstrable.
- Maintain correctness and functionality throughout the code.
- Utilize proper abstraction, modeling, and separation of concerns.
- Write modular, readable, and unit-testable code.
- Ensure the code can accommodate new requirements with minimal adjustments.
- Implement proper exception handling for robustness.

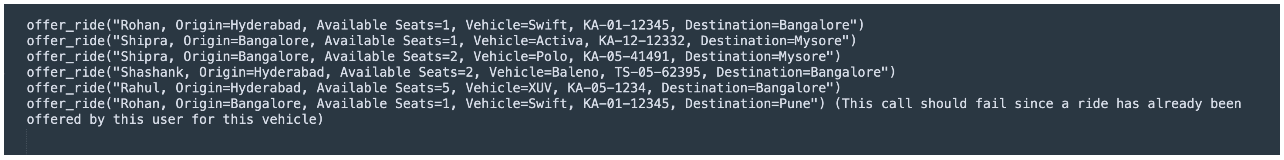
Sample Test Cases:

- Onboard 5 users:
  - Add users and their vehicles:



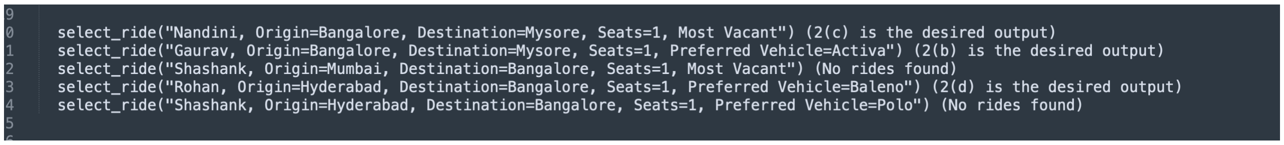
```
add_user("Rohan, M, 36"); add_vehicle("Rohan, Swift, KA-01-12345")
add_user("Shashank, M, 29"); add_vehicle("Shashank, Baleno, TS-05-62395")
add_user("Nandini, F, 29")
add_user("Shipra, F, 27"); add_vehicle("Shipra, Polo, KA-05-41491"); add_vehicle("Shipra, Activa, KA-12-12332")
add_user("Gaurav, M, 29")
add_user("Rahul, M, 35"); add_vehicle("Rahul, XUV, KA-05-1234")
```

- Offer 4 rides by 3 users:
  - Offer rides with details:



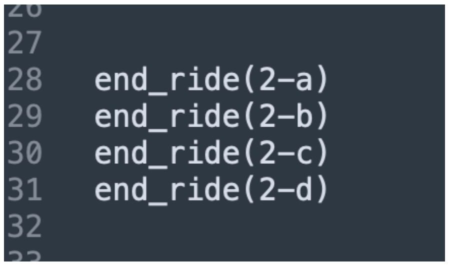
```
offer_ride("Rohan, Origin=Hyderabad, Available Seats=1, Vehicle=Swift, KA-01-12345, Destination=Bangalore")
offer_ride("Shipra, Origin=Bangalore, Available Seats=1, Vehicle=Activa, KA-12-12332, Destination=Mysore")
offer_ride("Shipra, Origin=Bangalore, Available Seats=2, Vehicle=Polo, KA-05-41491, Destination=Mysore")
offer_ride("Shashank, Origin=Hyderabad, Available Seats=2, Vehicle=Baleno, TS-05-62395, Destination=Bangalore")
offer_ride("Rahul, Origin=Hyderabad, Available Seats=5, Vehicle=XUV, KA-05-1234, Destination=Bangalore")
offer_ride("Rohan, Origin=Bangalore, Available Seats=1, Vehicle=Swift, KA-01-12345, Destination=Pune") (This call should fail since a ride has already been
offered by this user for this vehicle)
```

- Find rides for 4 users:
    - Search and select rides for users:



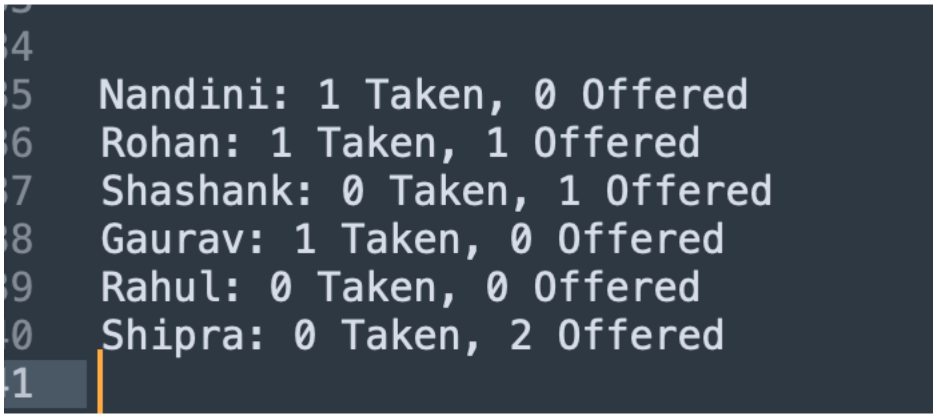
```
select_ride("Nandini, Origin=Bangalore, Destination=Mysore, Seats=1, Most Vacant") (2(c) is the desired output)
select_ride("Gaurav, Origin=Bangalore, Destination=Mysore, Seats=1, Preferred Vehicle=Activa") (2(b) is the desired output)
select_ride("Shashank, Origin=Mumbai, Destination=Bangalore, Seats=1, Most Vacant") (No rides found)
select_ride("Rohan, Origin=Hyderabad, Destination=Bangalore, Seats=1, Preferred Vehicle=Baleno") (2(d) is the desired output)
select_ride("Shashank, Origin=Hyderabad, Destination=Bangalore, Seats=1, Preferred Vehicle=Polo") (No rides found)
```

- End Rides:
    - End selected rides:



```
end_ride(2-a)
end_ride(2-b)
end_ride(2-c)
end_ride(2-d)
```

- Find total rides by user:
    - Display total rides taken and offered by each user: print_ride_stats()



```
Nandini: 1 Taken, 0 Offered
Rohan: 1 Taken, 1 Offered
Shashank: 0 Taken, 1 Offered
Gaurav: 1 Taken, 0 Offered
Rahul: 0 Taken, 0 Offered
Shipra: 0 Taken, 2 Offered
```