

Ride Sharing

This project implements a simple ride-sharing system with classes for managing users (riders and drivers), rides, vehicles, and ride requests. It allows users to request rides, drivers to accept ride requests, and tracks the ride details including start and end locations, driver, rider, start and end times, and fares.

Classes and their Methods:

1. Ride_Sharing:

- Methods:
 - `__init__(self, company_name)`: Initializes a ride-sharing company with a specified name.
 - `add_rider(self, rider)`: Adds a rider to the ride-sharing system.
 - `add_driver(self, driver)`: Adds a driver to the ride-sharing system.
 - `__repr__(self)`: String representation of the ride-sharing company.

2. User (Abstract Base Class):

- Methods:
 - `__init__(self, name, email, nid)`: Initializes a user with a name, email, and national ID.
 - `display_profile(self)`: Abstract method to display the user profile.

3. Rider (Subclass of User):

- Methods:
 - `__init__(self, name, email, nid, current_location, initial_amount)`: Initializes a rider with name, email, national ID, current location, and initial wallet amount.
 - `display_profile(self)`: Displays rider's profile.
 - `load_cash(self, amount)`: Adds cash to rider's wallet.
 - `update_location(self, current_location)`: Updates rider's current location.
 - `request_ride(self, ride_sharing, destination)`: Requests a ride to a destination.
 - `show_current_ride(self)`: Displays details of the current ride.

4. Driver (Subclass of User):

- Methods:
 - `__init__(self, name, email, nid, current_location)`: Initializes a driver with name, email, national ID, and current location.
 - `display_profile(self)`: Displays driver's profile.
 - `accept_ride(self, ride)`: Accepts a ride request.

5. Ride:

- Methods:
 - `__init__(self, start_location, end_location)`: Initializes a ride with start and end locations.
 - `set_driver(self, driver)`: Assigns a driver to the ride.
 - `start_ride(self)`: Marks the start time of the ride.
 - `end_ride(self, rider, amount)`: Marks the end time of the ride and processes payment.
 - `__repr__(self)`: String representation of the ride.

6. Ride_Request:

- Methods:
 - `__init__(self, rider, end_location)`: Initializes a ride request with a rider and destination.

7. Ride_Matching:

- Methods:
 - `__init__(self, drivers)`: Initializes a ride matcher with available drivers.
 - `find_driver(self, ride_request)`: Matches a driver to a ride request.

8. Vehicle (Abstract Base Class):

- Methods:
 - `__init__(self, vehicle_type, license_plate, rate)`: Initializes a vehicle with type, license plate, and rate.
 - `start_drive(self)`: Abstract method to start driving the vehicle.

9. Car (Subclass of Vehicle):

- Methods:
 - `__init__(self, vehicle_type, license_plate, rate)`: Initializes a car with type, license plate, and rate.
 - `start_drive(self)`: Marks the car as unavailable for driving.

10. Bike (Subclass of Vehicle):

- Methods:
 - `__init__(self, vehicle_type, license_plate, rate)`: Initializes a bike with type, license plate, and rate.
 - `start_drive(self)`: Marks the bike as unavailable for driving.