```
# car_rental.py
from datetime import datetime, timedelta

class CarRental:
    def __init__(self, total_cars):
        self.total_cars = total_cars
        self.available_cars = total_cars
        self.rented_cars = {}


    def display_available_cars(self):
        return f"Available Cars: {self.available_cars}"

    def rent_hourly(self, num_cars):
        return self._rent_car(num_cars, "hourly")

    def rent_daily(self, num_cars):
        return self._rent_car(num_cars, "daily")

    def rent_weekly(self, num_cars):
        return self._rent_car(num_cars, "weekly")

    def return_cars(self, rental_start_time, rental_mode, num_cars):
        if rental_mode not in ["hourly", "daily", "weekly"]:
            return "Invalid rental mode"

        if rental_start_time not in self.rented_cars:
            return "Invalid rental start time"

        rented_period = datetime.now() - rental_start_time
        rented_period_hours = rented_period.total_seconds() / 3600

        if rental_mode == "hourly":
            bill = 5 * rented_period_hours * num_cars
        elif rental_mode == "daily":
            bill = 20 * rented_period_hours / 24 * num_cars
        elif rental_mode == "weekly":
            bill = 50 * rented_period_hours / (24 * 7) * num_cars

        self.available_cars += num_cars
        del self.rented_cars[rental_start_time]

        return f"Rental period: {rented_period}, Total bill: {bill}$"

    def _rent_car(self, num_cars, rental_mode):
        if num_cars <= 0:
            return "Invalid number of cars"
```

```
if num_cars > self.available_cars:
    return "Not enough cars available"

self.available_cars -= num_cars
self.rented_cars[datetime.now()] = (num_cars, rental_mode)

return f"Rented {num_cars} cars for {rental_mode} mode"
```

```
# customer.py
from car_rental import CarRental

class Customer:
    def __init__(self, name):
        self.name = name

    def request_cars(self, rental, num_cars, rental_mode):
        return rental._rent_car(num_cars, rental_mode)

    def return_cars(self, rental, rental_start_time, rental_mode, num_cars):
        return rental.return_cars(rental_start_time, rental_mode, num_cars)
```

```
# main.ipynb
from car_rental import CarRental
from customer import Customer

def main():
   total_cars = 10
   rental = CarRental(total_cars)
   customer = Customer("John Doe")

   while True:
      print("1. Display available cars")
      print("2. Rent a car")
      print("3. Return a car")
      print("4. Exit")

      choice = int(input("Enter your choice (1-4): "))

      if choice == 1:
         print(rental.display_available_cars())
      elif choice == 2:
         num_cars = int(input("Enter the number of cars to rent: "))
         rental_mode = input("Enter rental mode (hourly/daily/weekly): ")
         print(customer.request_cars(rental, num_cars, rental_mode))
      elif choice == 3:
         rental_start_time = datetime.strptime(input("Enter rental start time (YYYY-MM-DD
HH:MM:SS): "), "%Y-%m-%d %H:%M:%S")
         rental_mode = input("Enter rental mode (hourly/daily/weekly): ")
         num_cars = int(input("Enter the number of cars to return: "))
         print(customer.return_cars(rental, rental_start_time, rental_mode, num_cars))
      elif choice == 4:
         print("Exiting program.")
         break
      else:
         print("Invalid choice. Please enter a number between 1 and 4.")

if __name__ == "__main__":
   main()
```