**CASE STUDY PYTHON**

**NAME: - SUSHANT KUMAR SINGH**

**PROJECT:- CarConnect, a Car Rental Platform**

Create following tables in SQL Schema with appropriate class and write the unit test case for the

application.

SQL Tables:

**1. Customer Table:**

• CustomerID (Primary Key): Unique identifier for each customer.

• FirstName: First name of the customer.

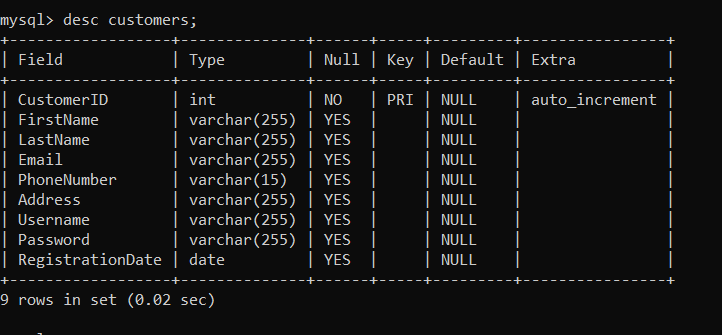
• LastName: Last name of the customer.

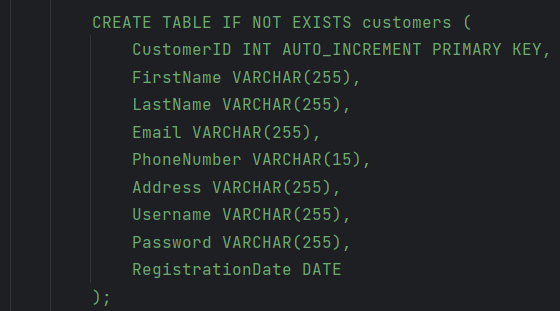
• Email: Email address of the customer for communication.

• PhoneNumber: Contact number of the customer.

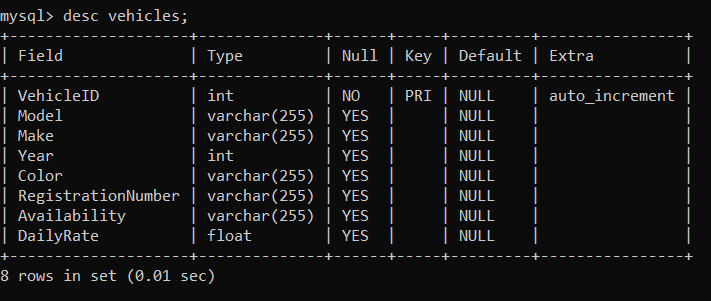
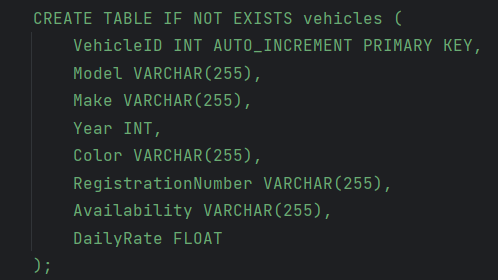
• Address: Customer's residential address.

• Username: Unique username for customer login.

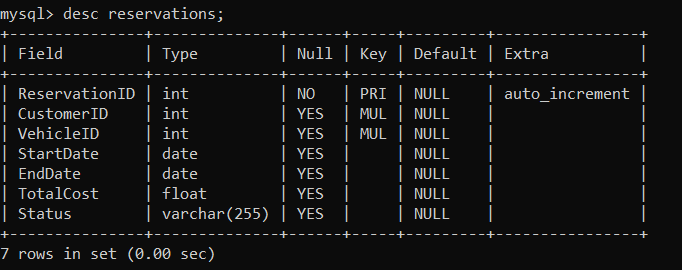
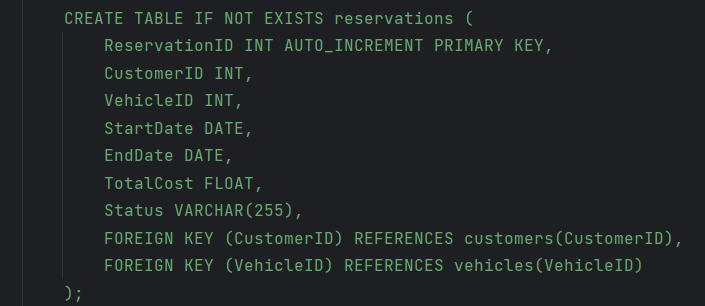
• Password: Securely hashed password for customer authentication.****

****

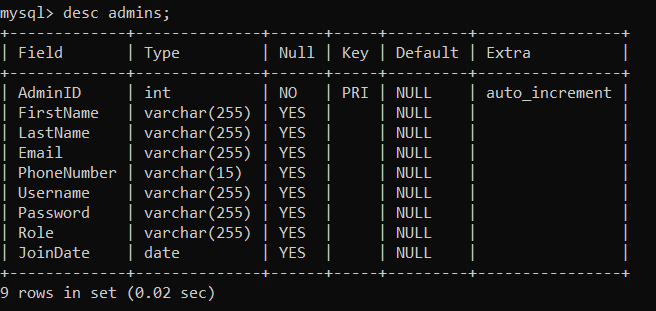
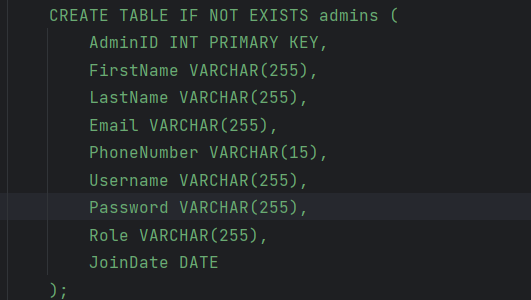
**2. Vehicle Table:**



**3. Reservation Table:**



**4) Admin Table**



Create the model/entity classes corresponding to the schema within package entity with variables

declared private, constructors (default and parametrized) and getters,setters )

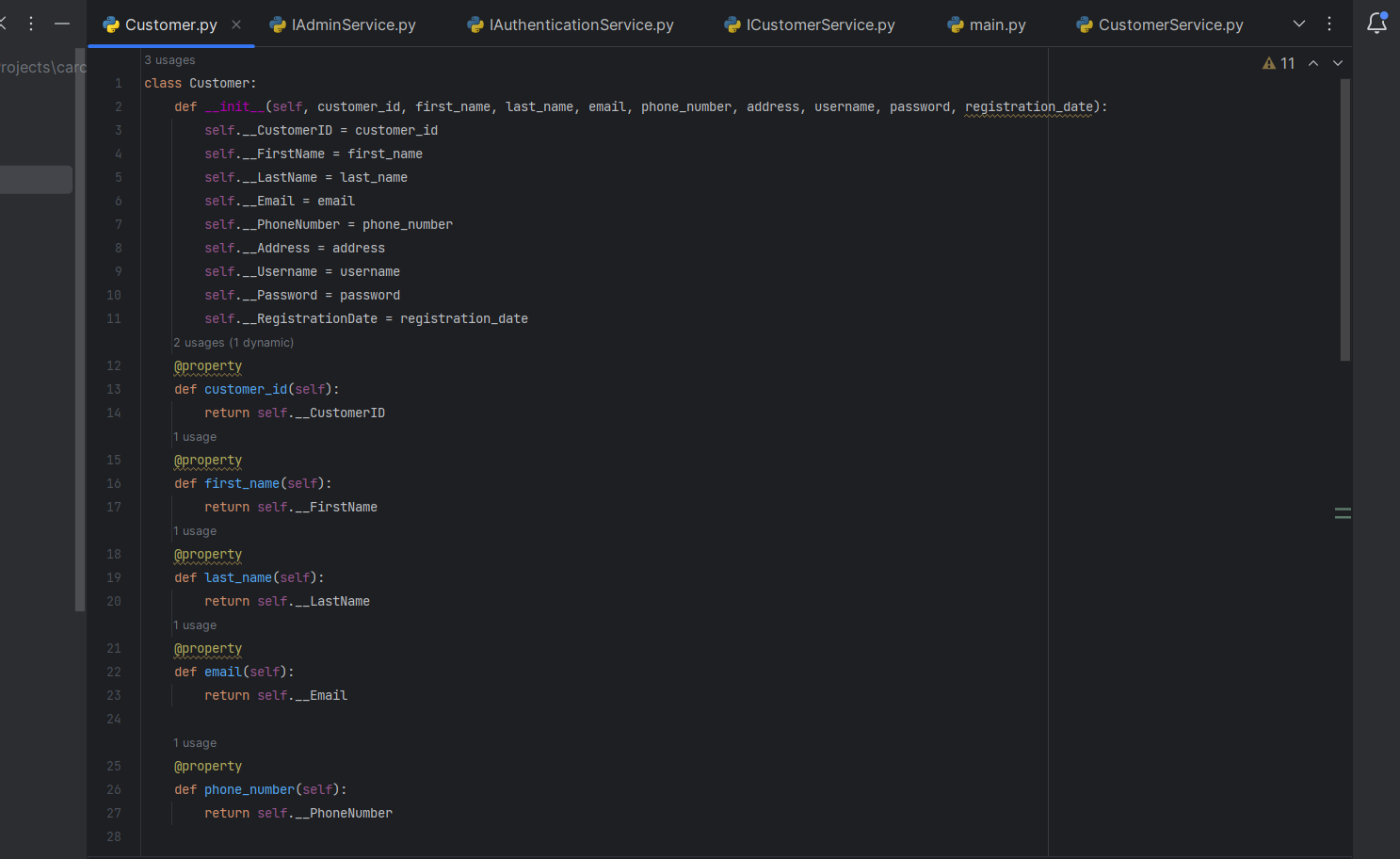
**Classes:**

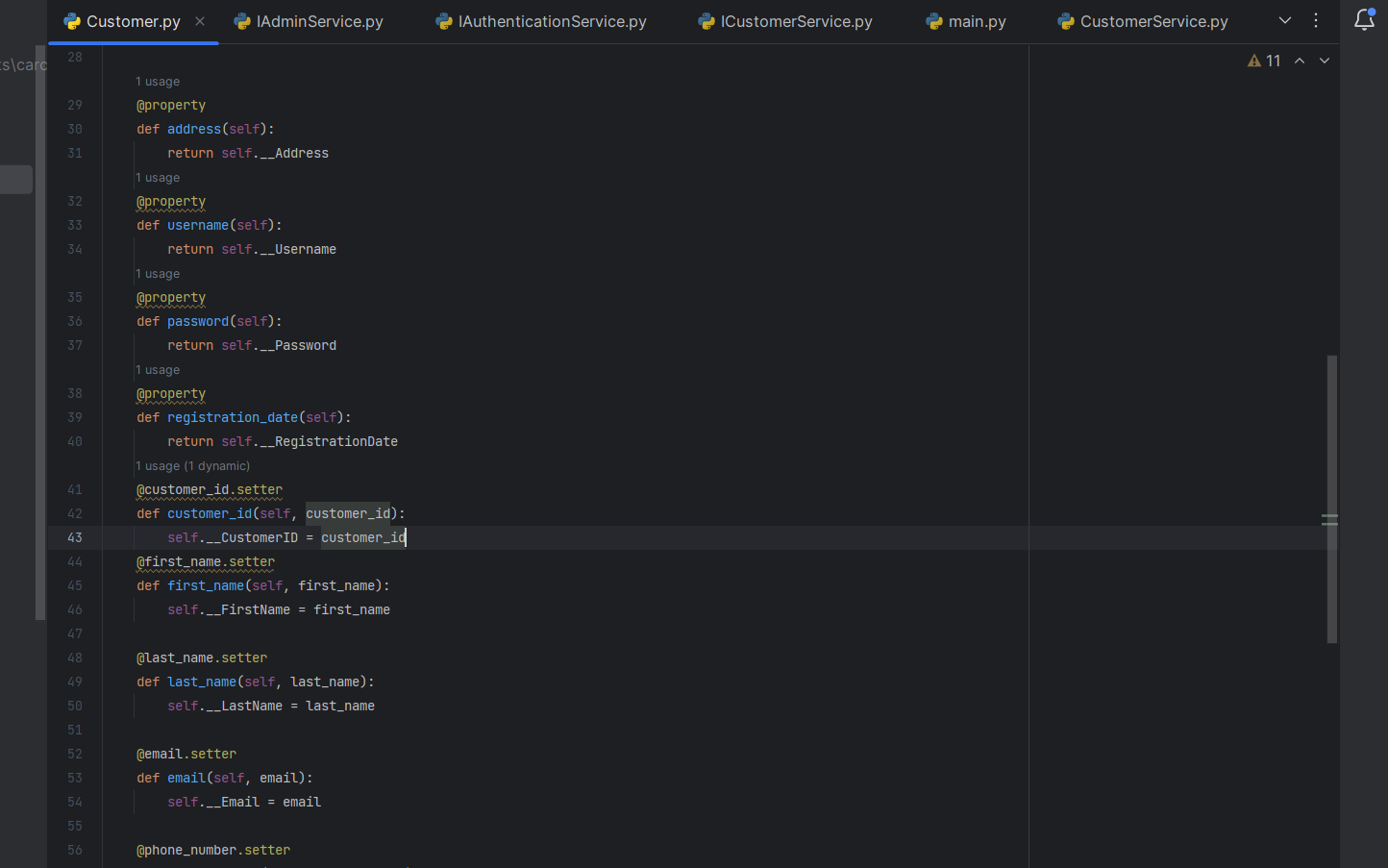
**Customer:**

• Properties: CustomerID, FirstName, LastName, Email, PhoneNumber, Address,

Username, Password, RegistrationDate

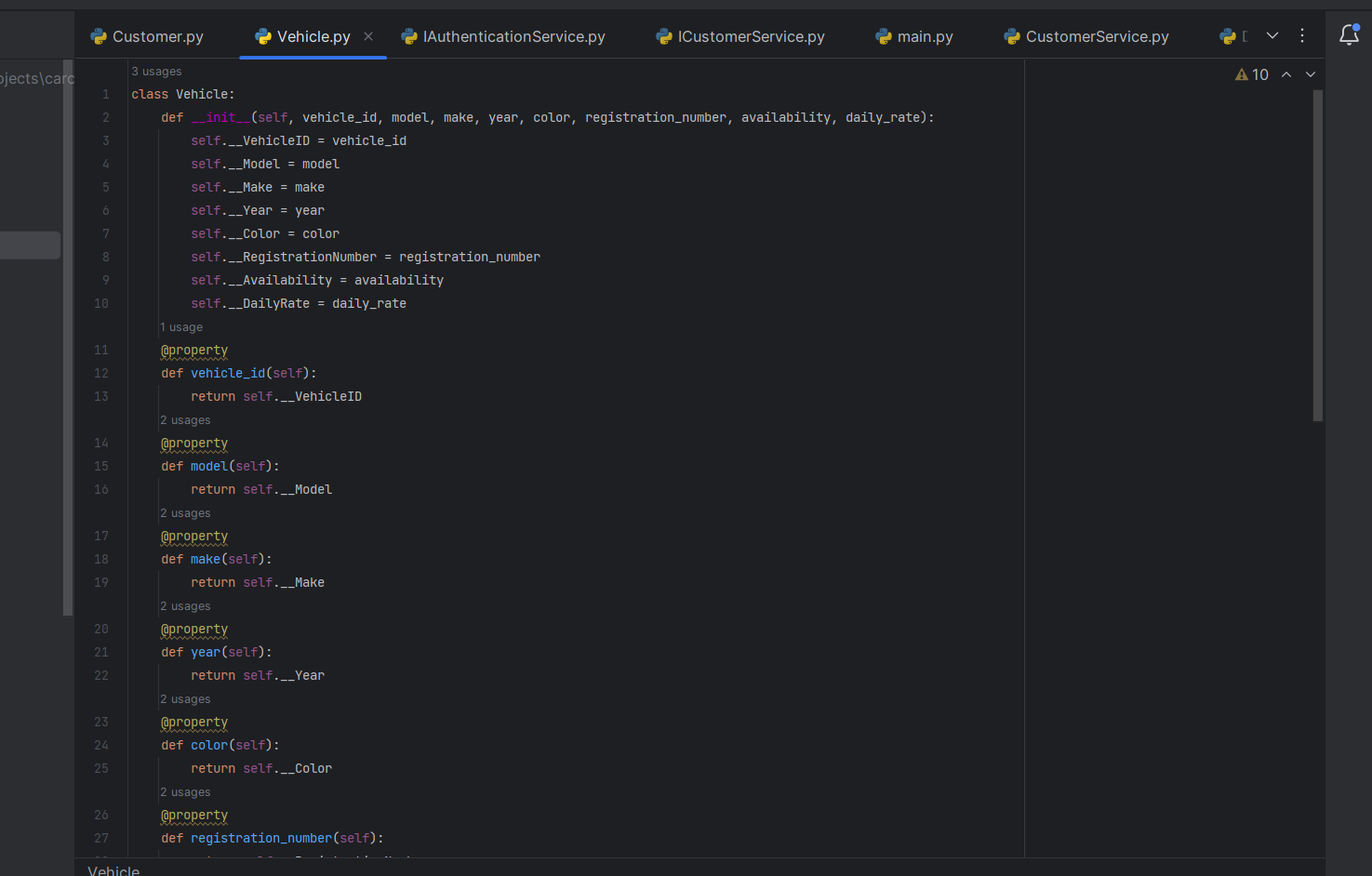
• Methods: Authenticate(password)

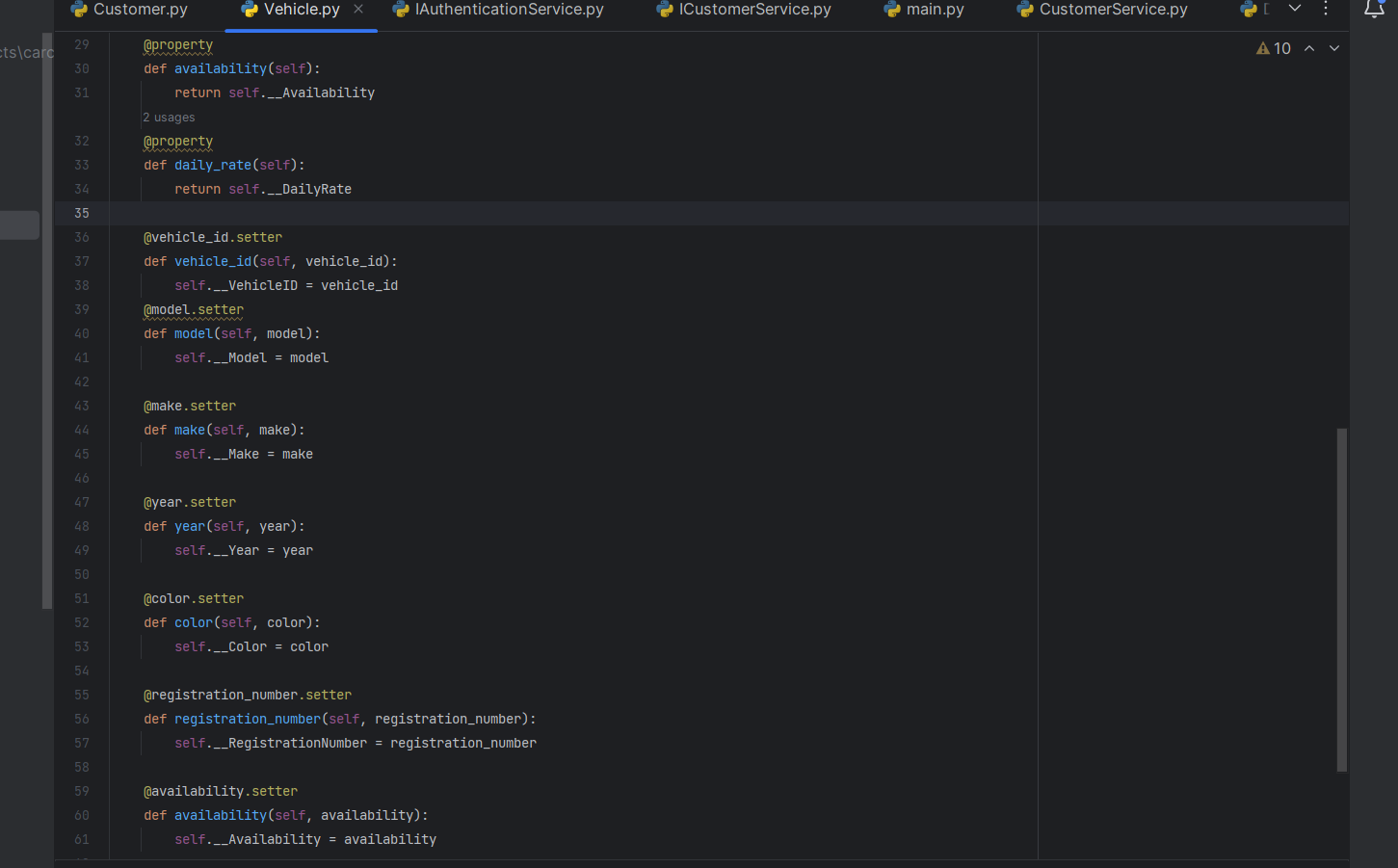


****

**Vehicle:**

• Properties: VehicleID, Model, Make, Year, Color, RegistrationNumber, Availability, DailyRate

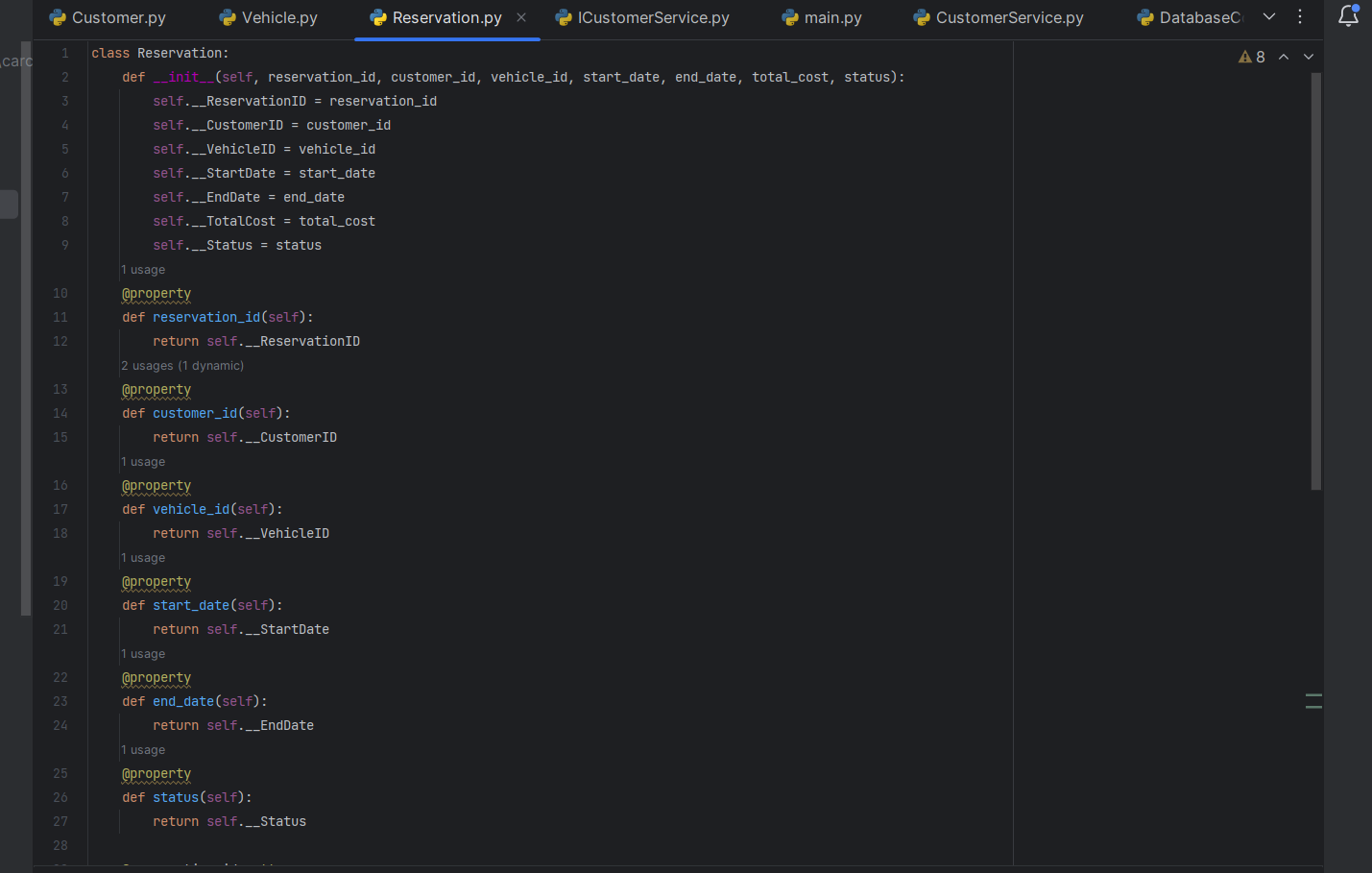


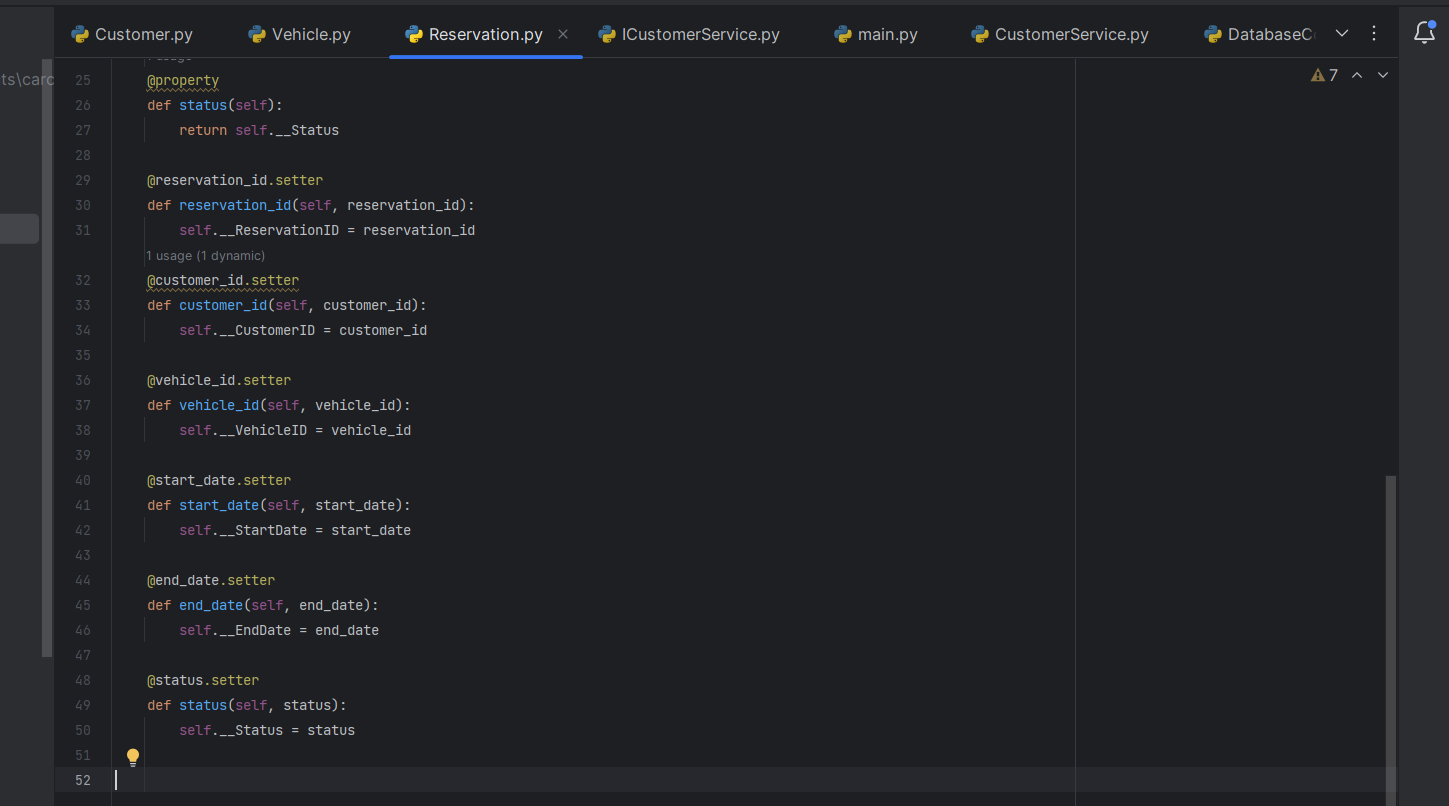
****

**Reservation:**

• Properties: ReservationID, CustomerID, VehicleID, StartDate, EndDate, TotalCost, Status

• Methods: CalculateTotalCost()

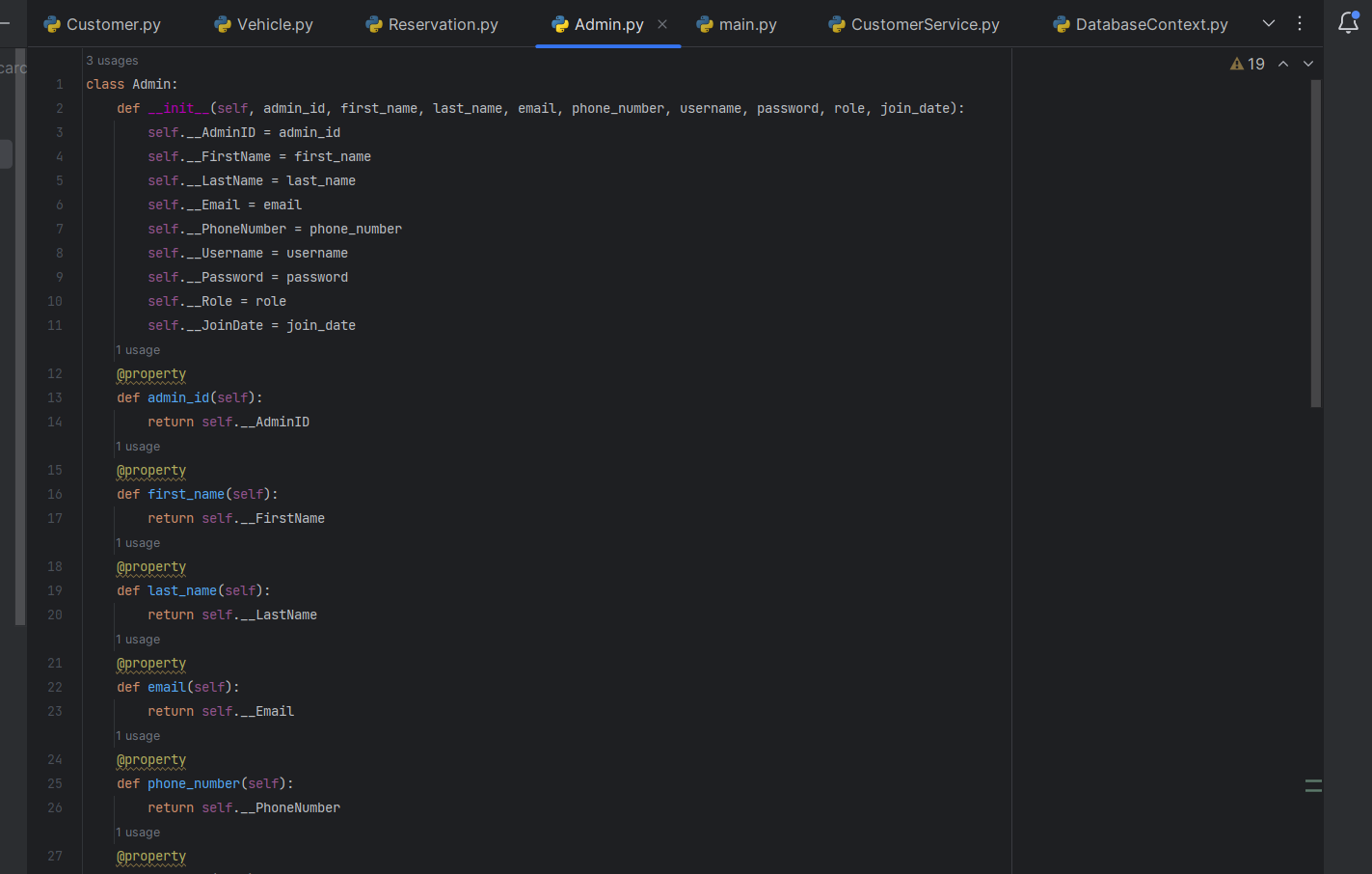


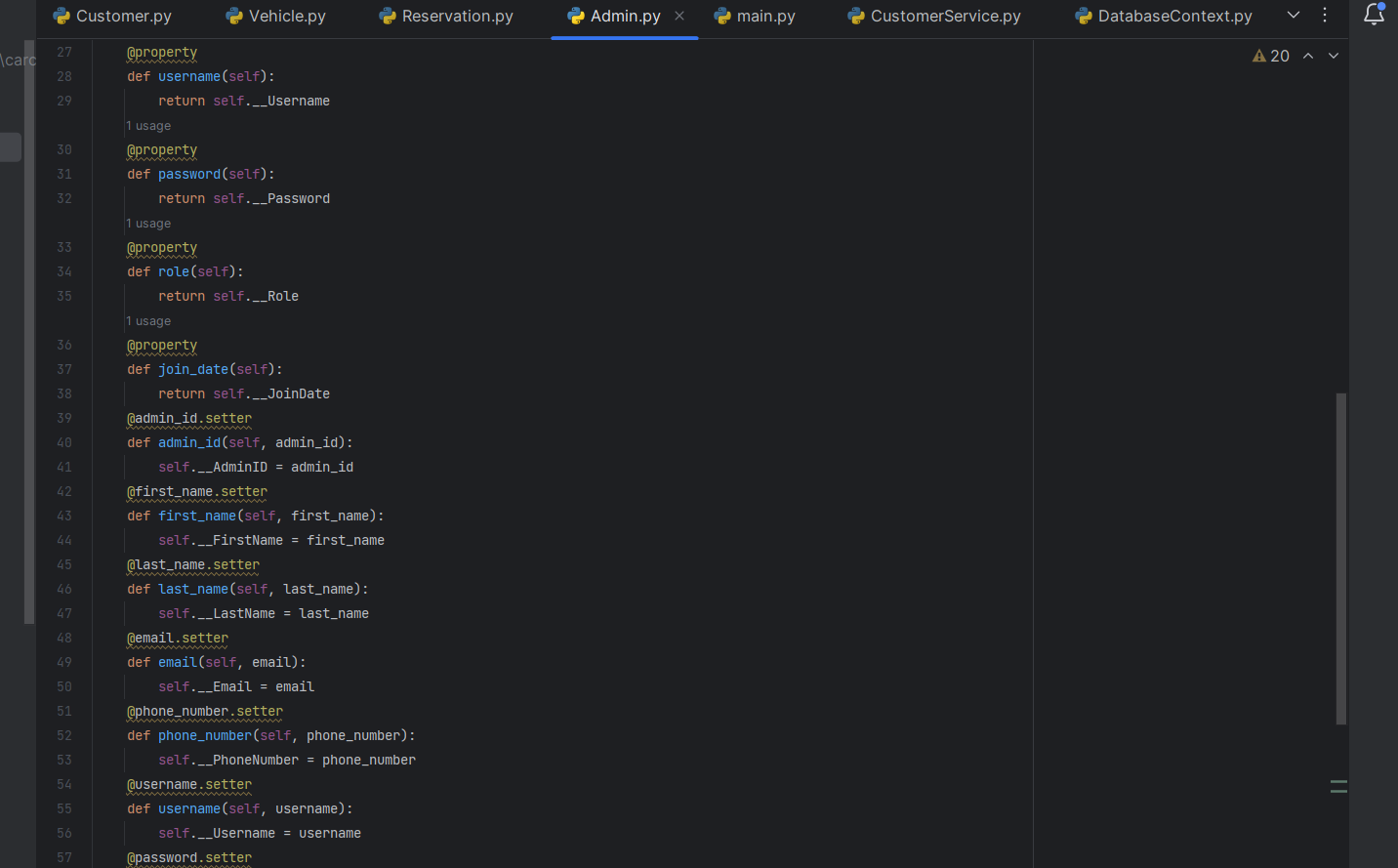


**Admin:**

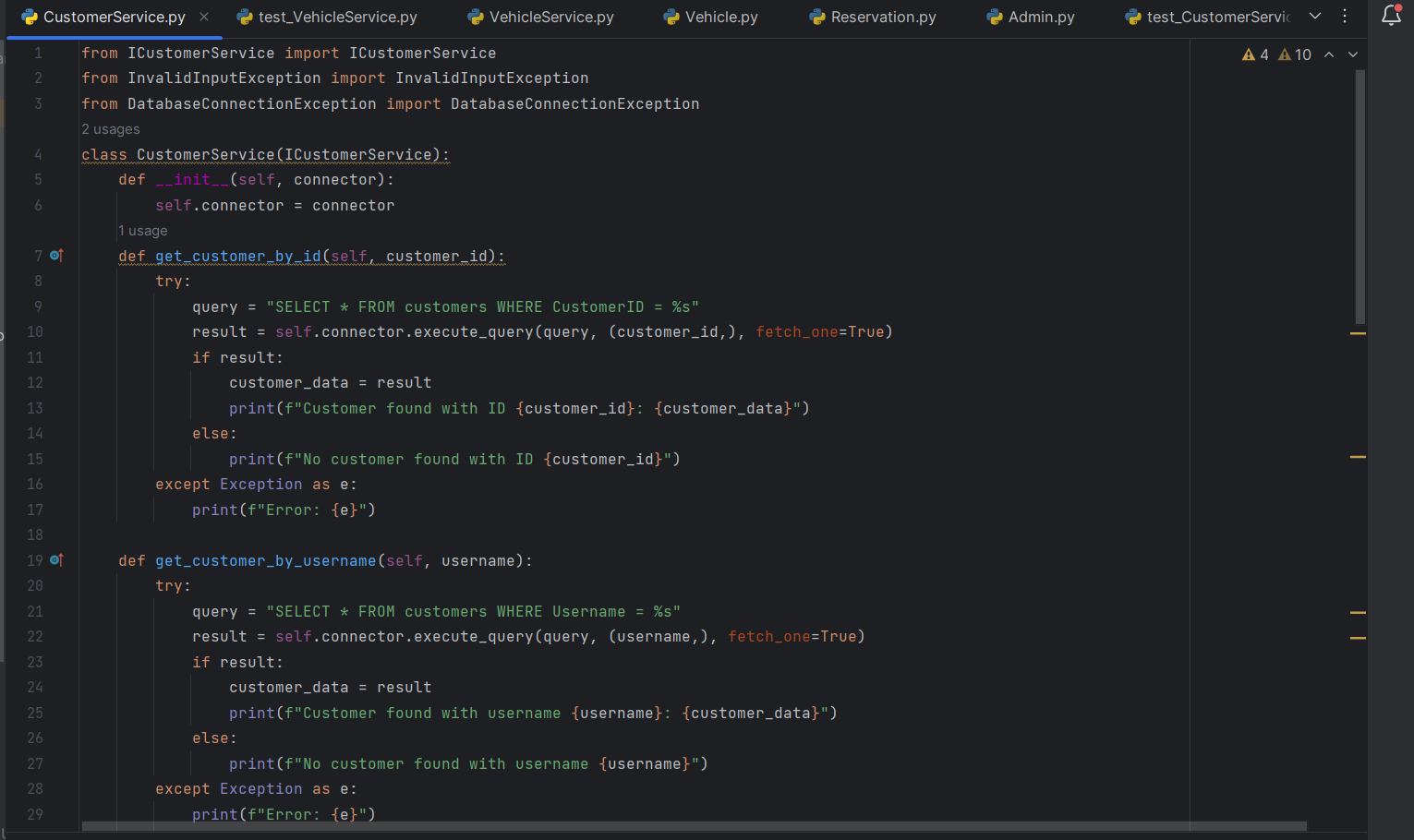
• Properties: AdminID, FirstName, LastName, Email, PhoneNumber, Username, Password, Role, JoinDate

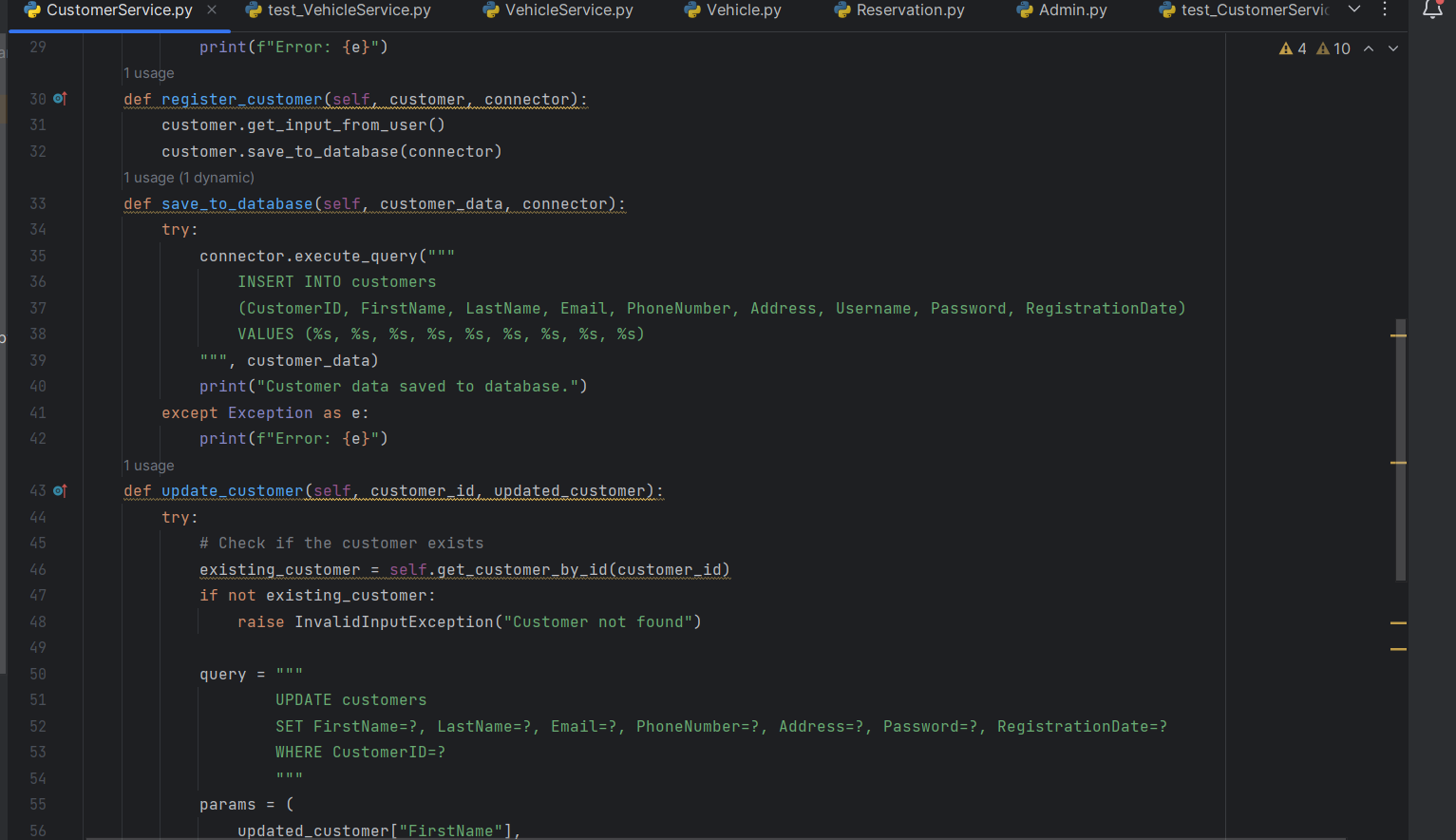
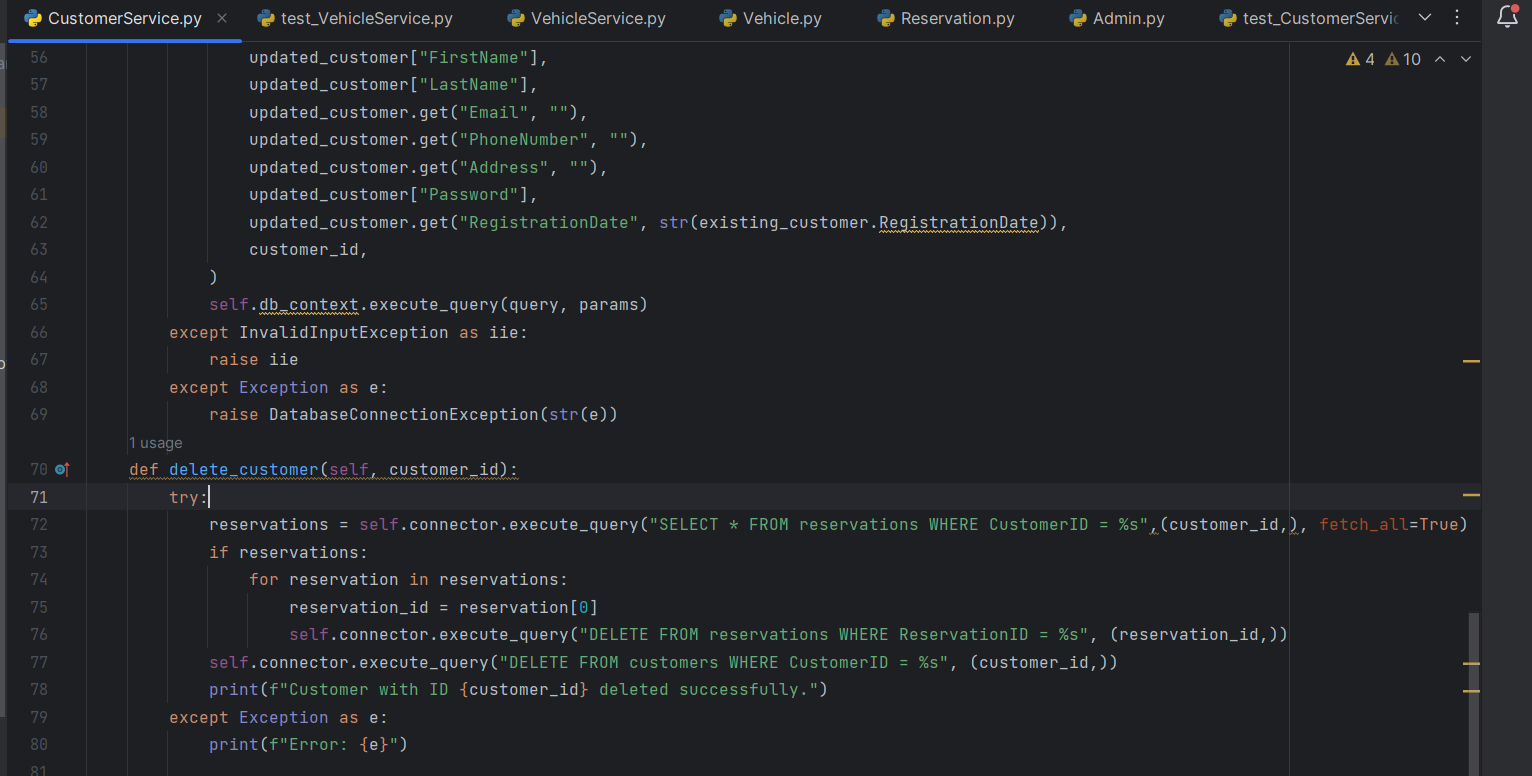
• Methods: Authenticate(password)





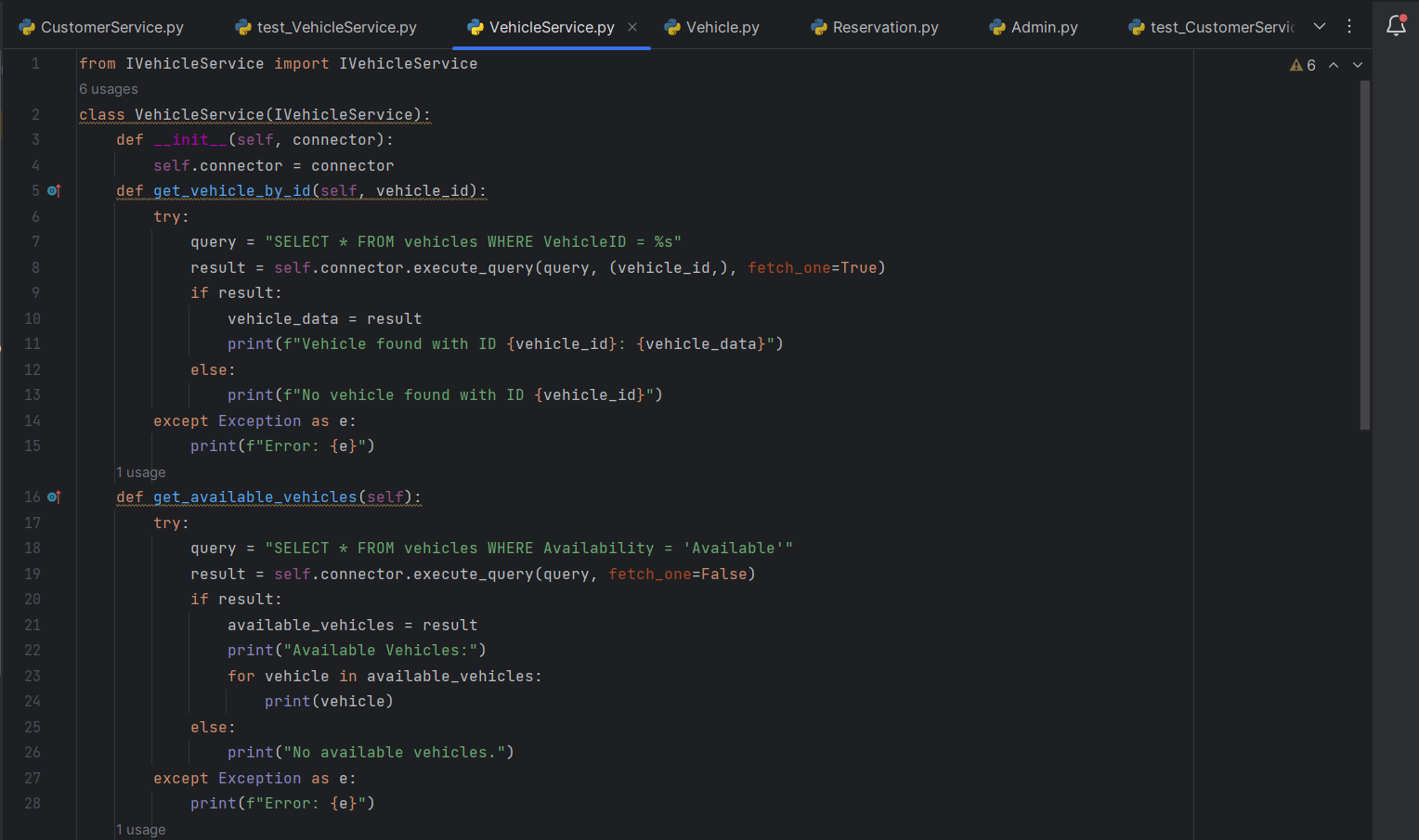
**CustomerService (implements ICustomerService):**

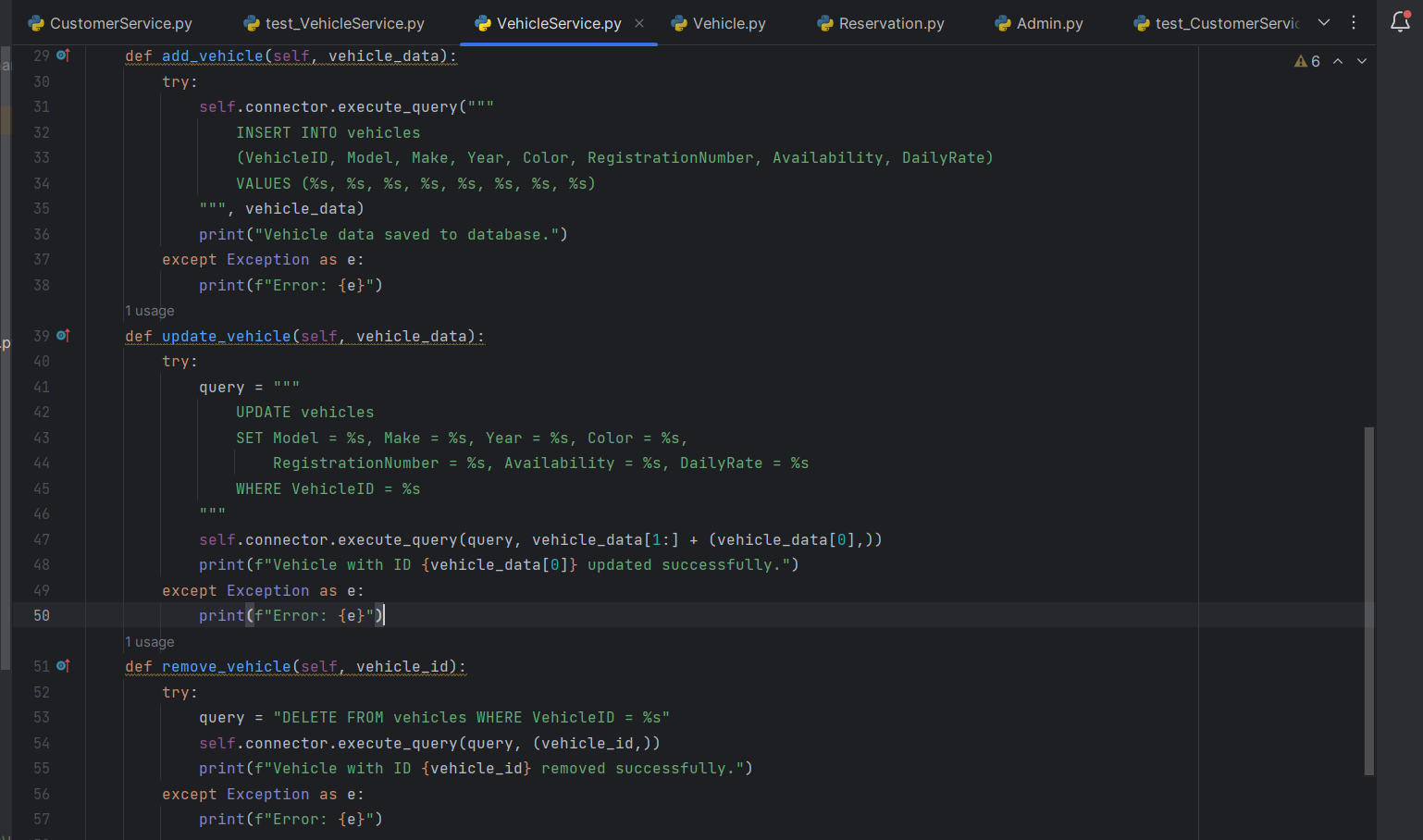
• Methods: GetCustomerById, GetCustomerByUsername, RegisterCustomer, UpdateCustomer, DeleteCustomer



**VehicleService (implements IVehicleService):**

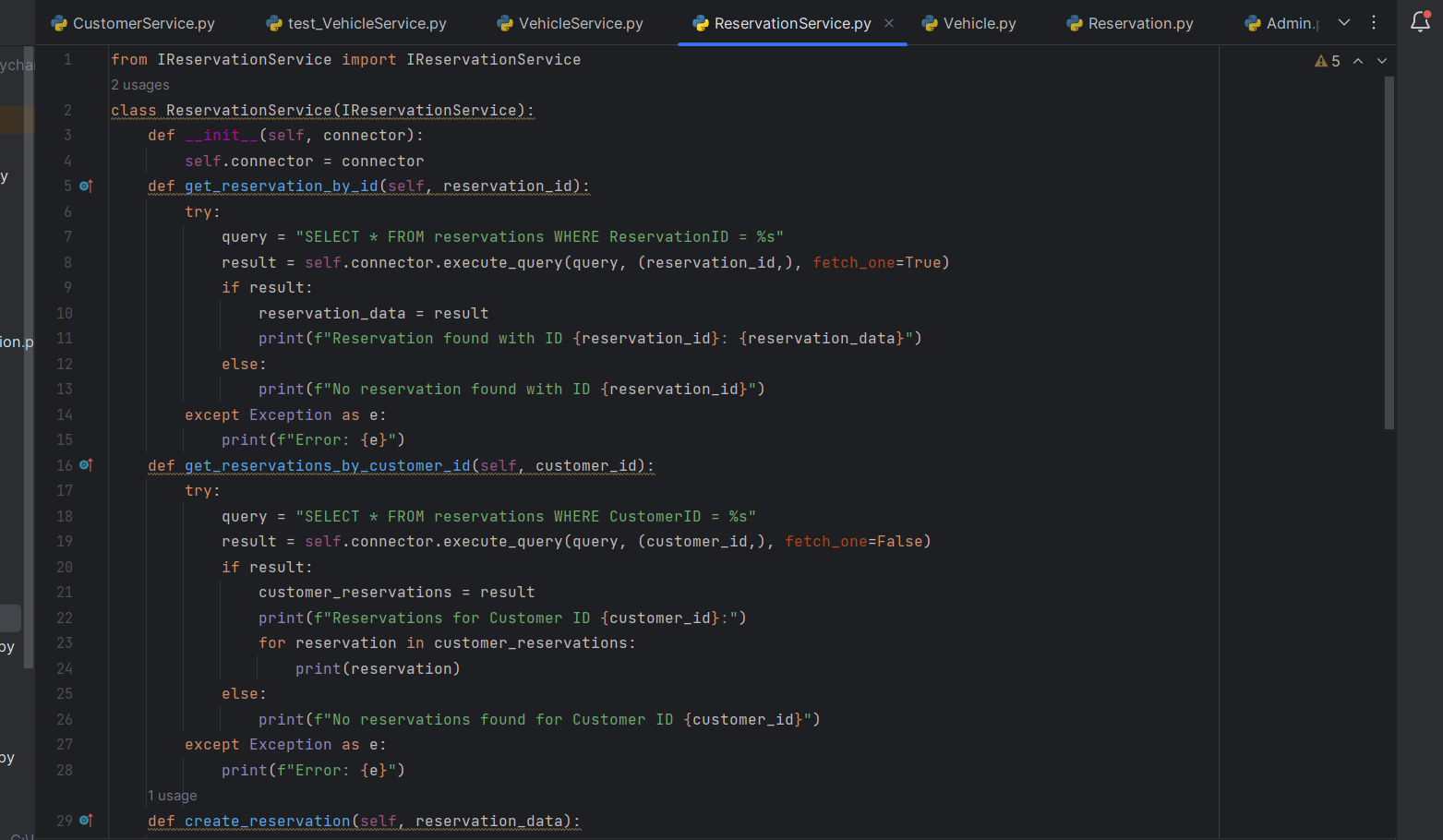
• Methods: GetVehicleById, GetAvailableVehicles, AddVehicle, UpdateVehicle, RemoveVehicle





**ReservationService (implements IReservationService):**

• Methods: GetReservationById, GetReservationsByCustomerId, CreateReservation, UpdateReservation, CancelReservation

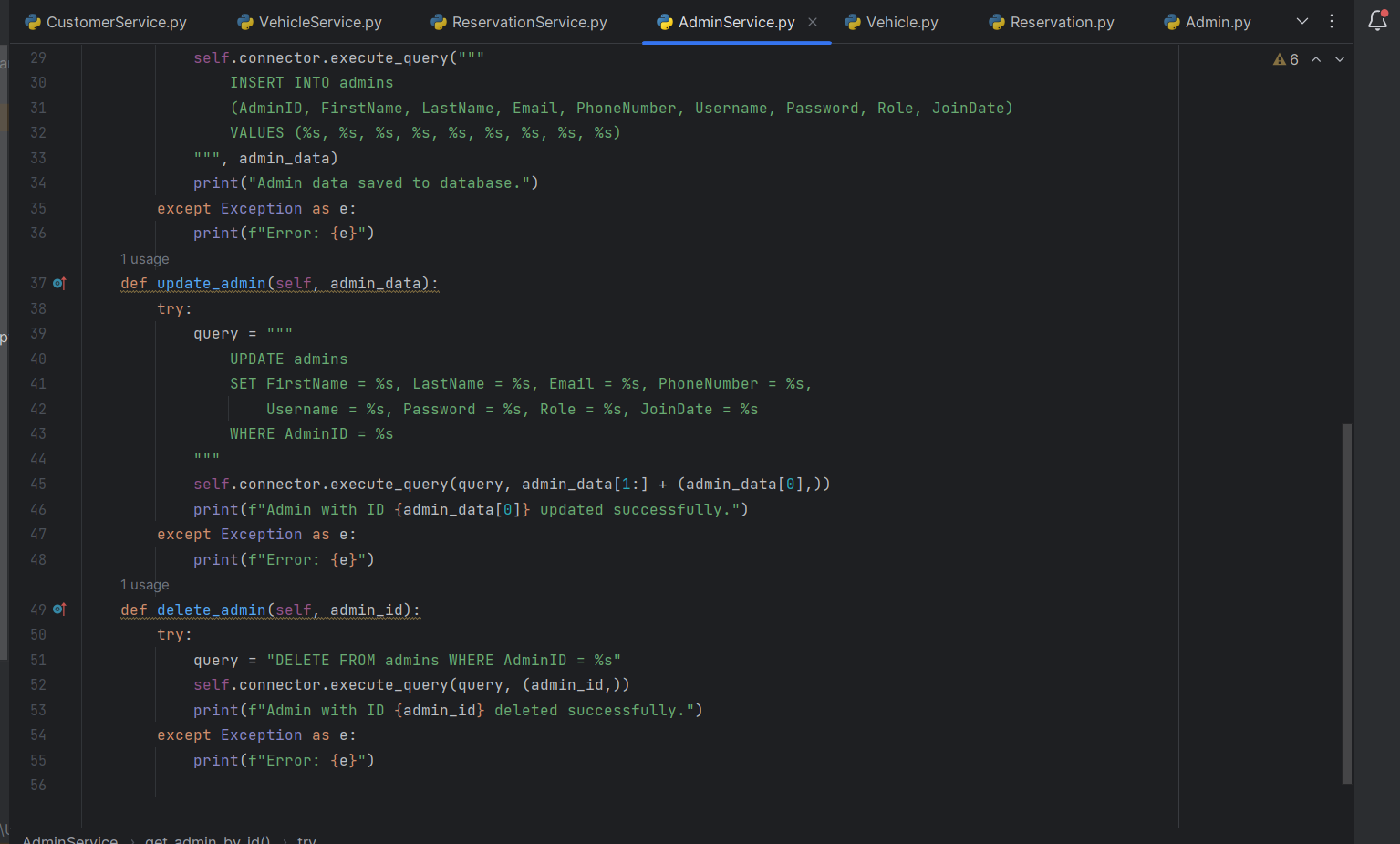




**AdminService (implements IAdminService):**

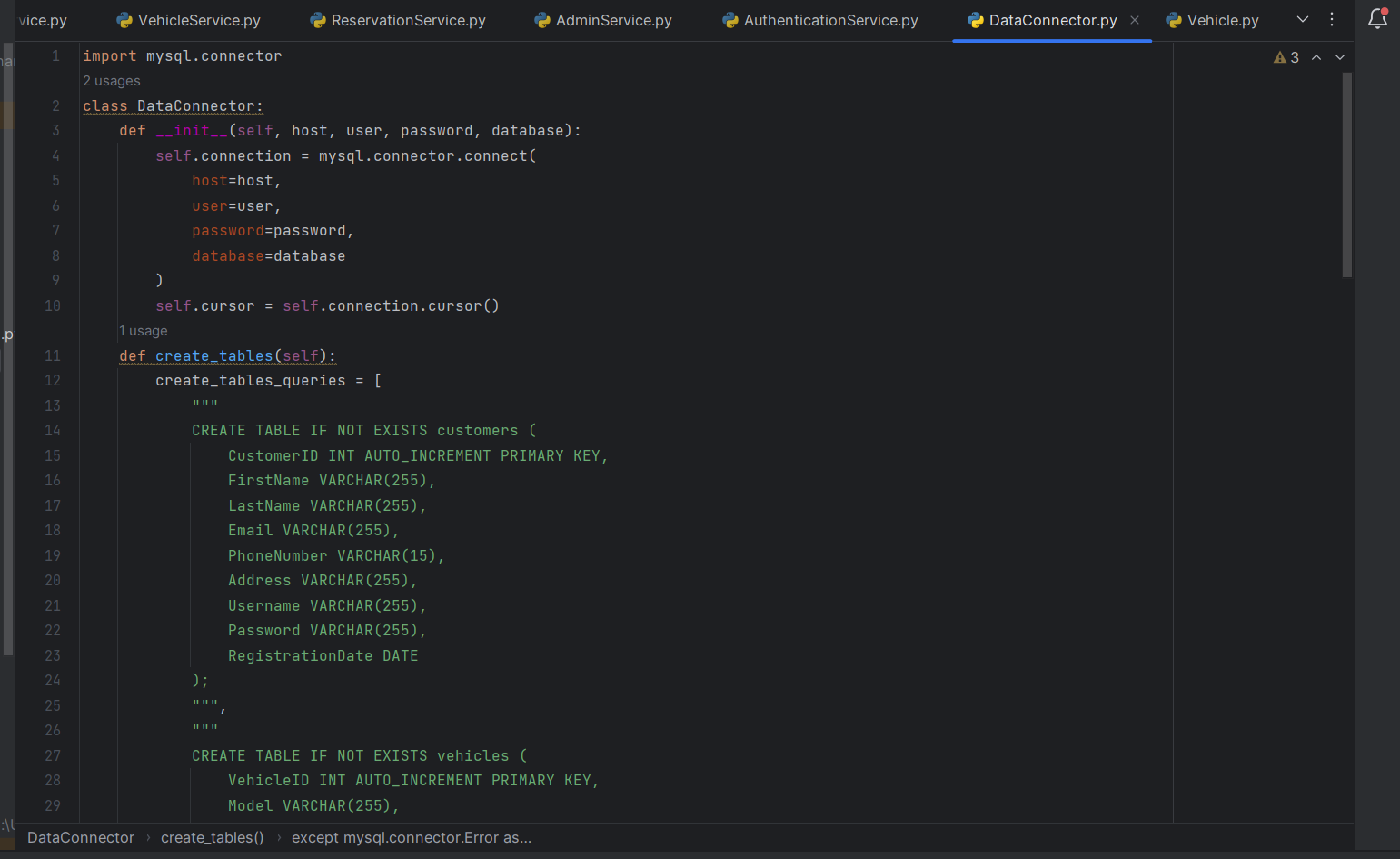
• Methods: GetAdminById, GetAdminByUsername, RegisterAdmin, UpdateAdmin, DeleteAdmin



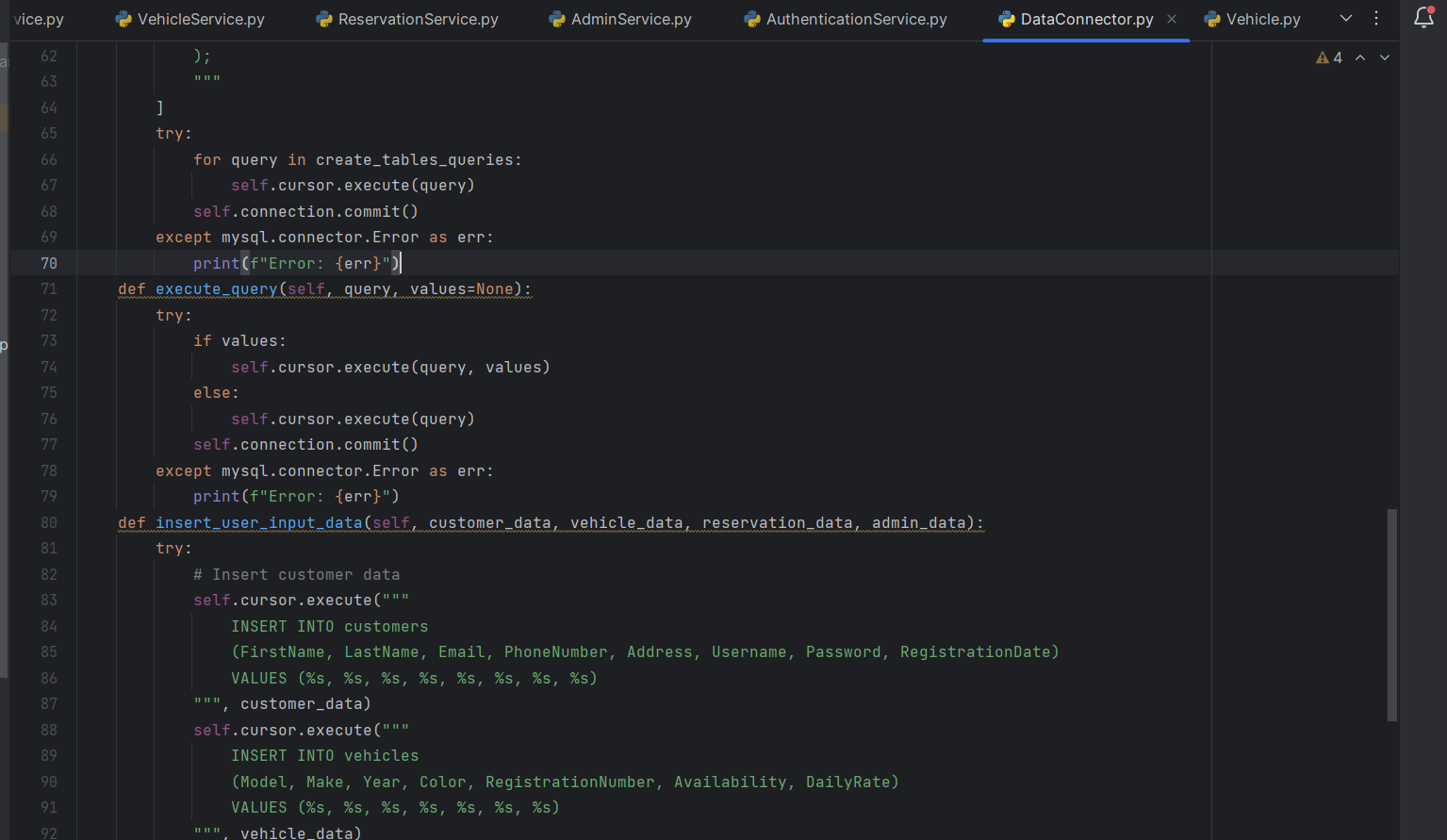


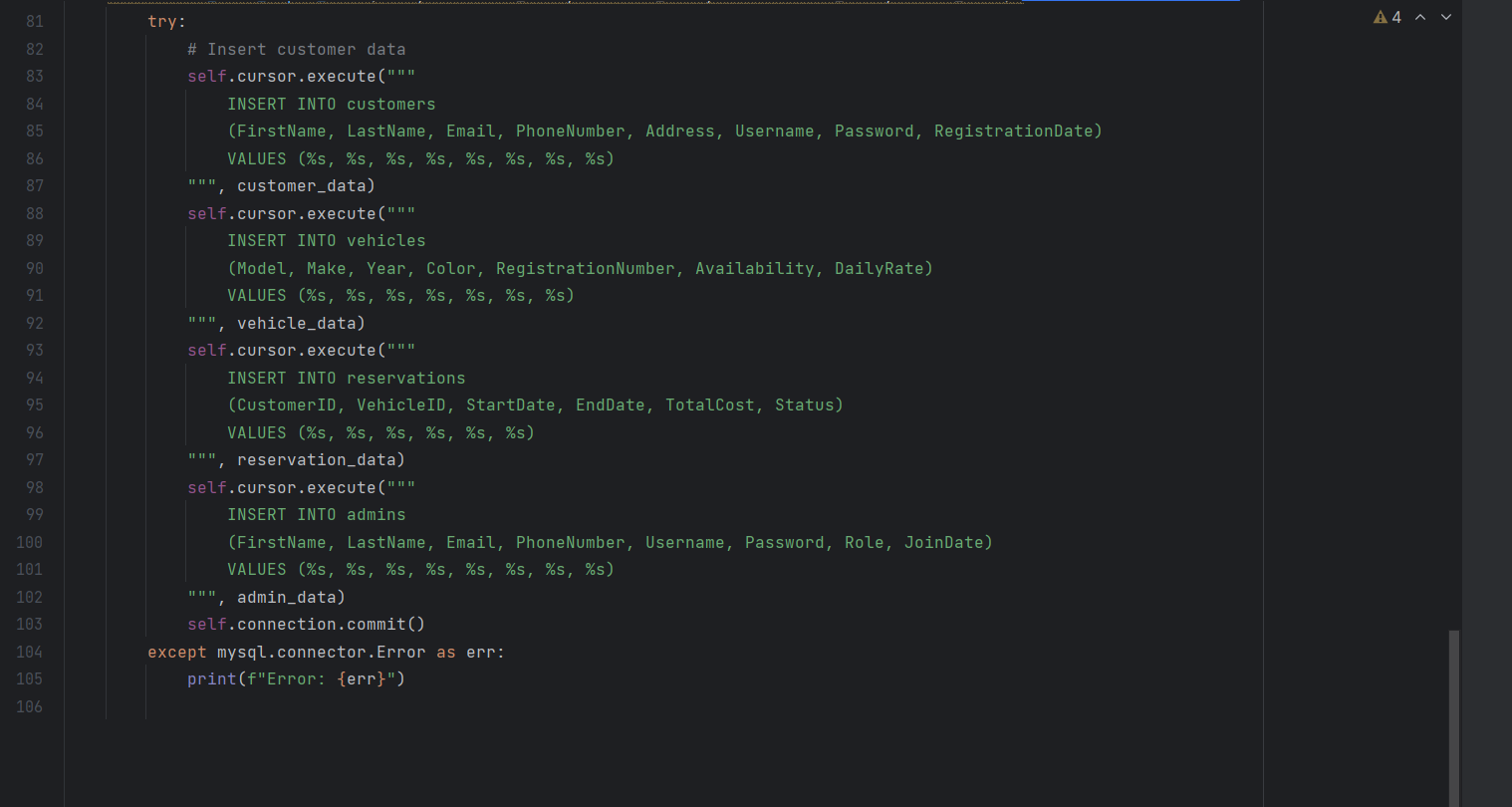
**DatabaseConnector:**

• A class responsible for handling database connections and interactions.

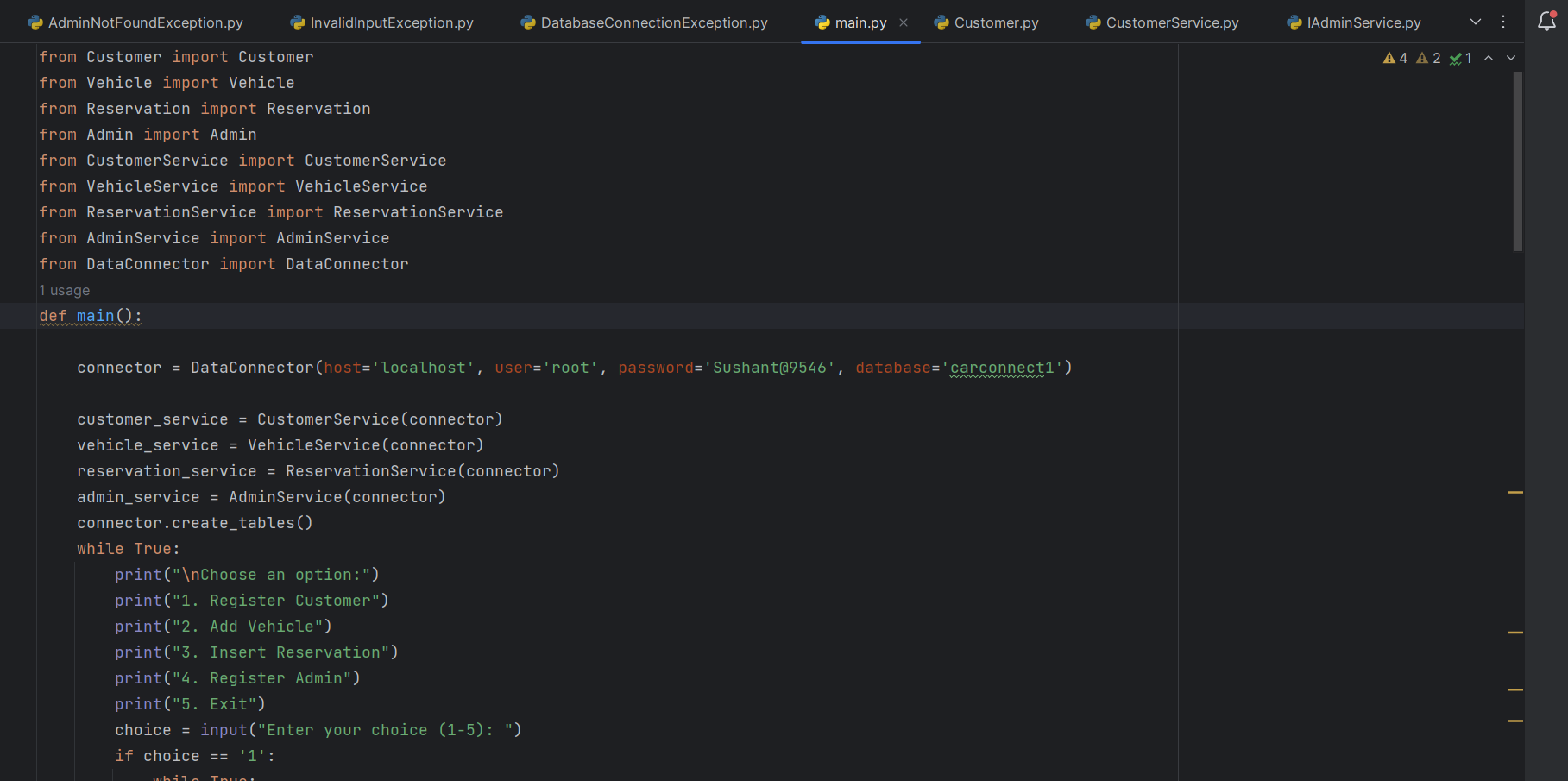


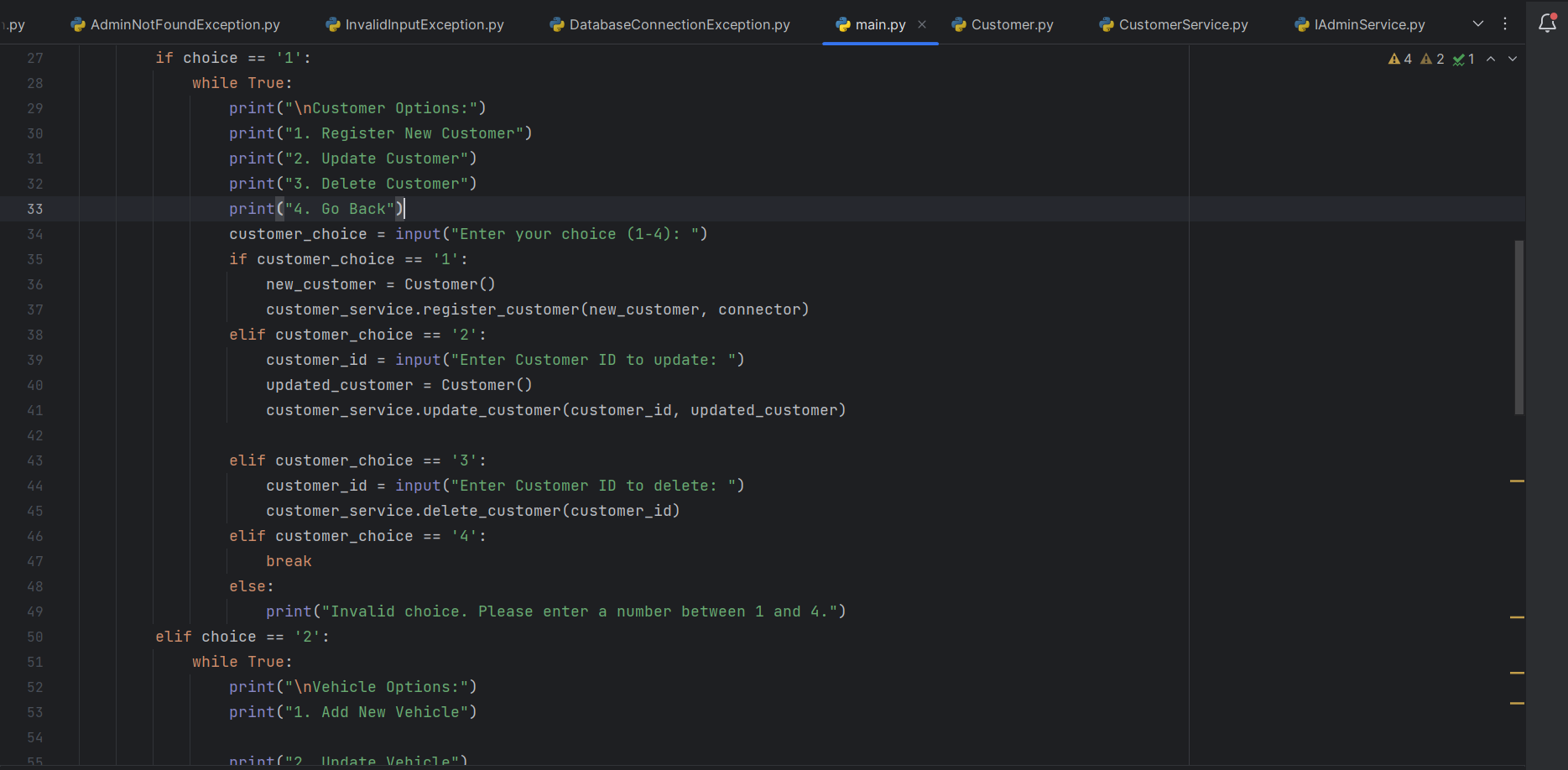


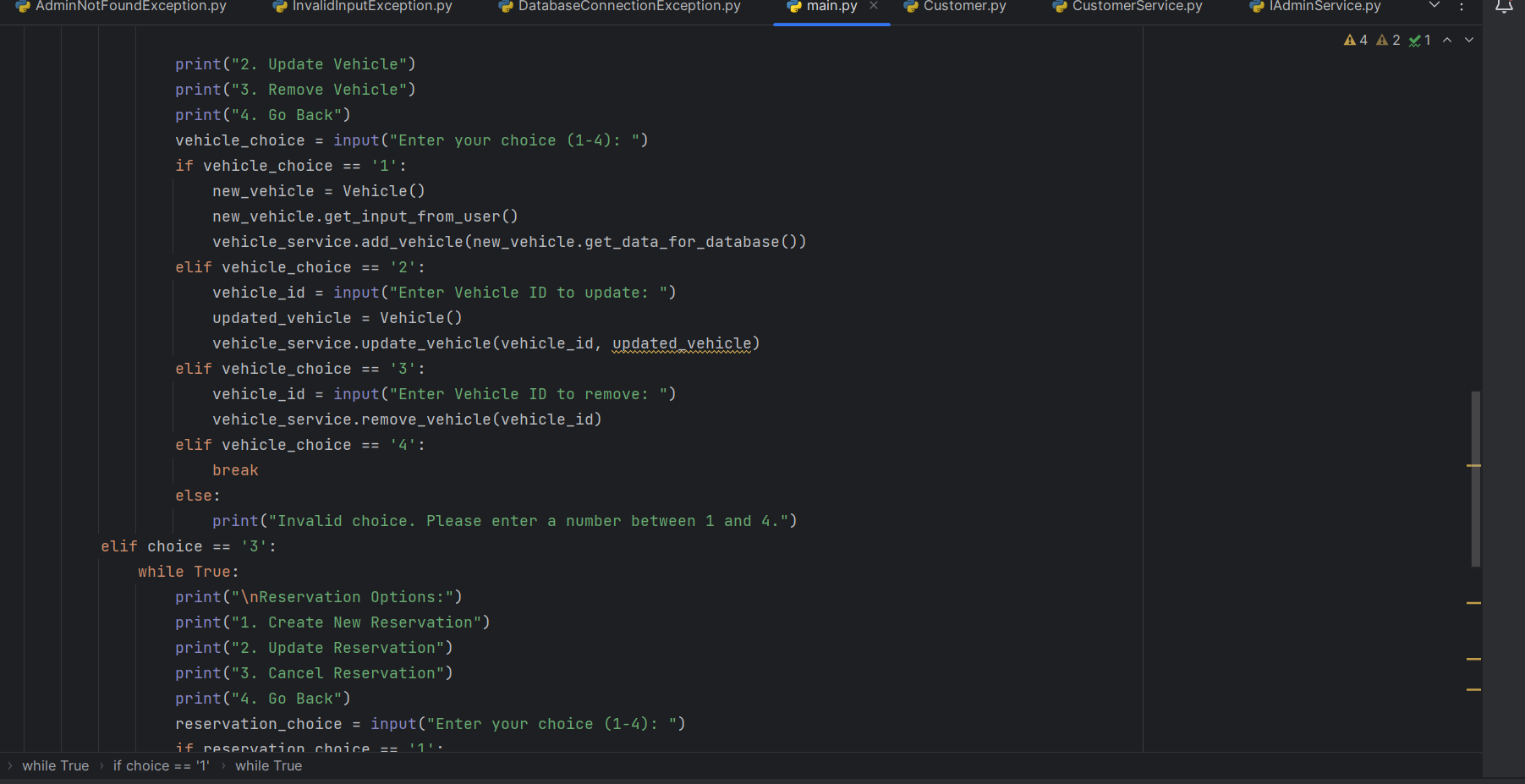


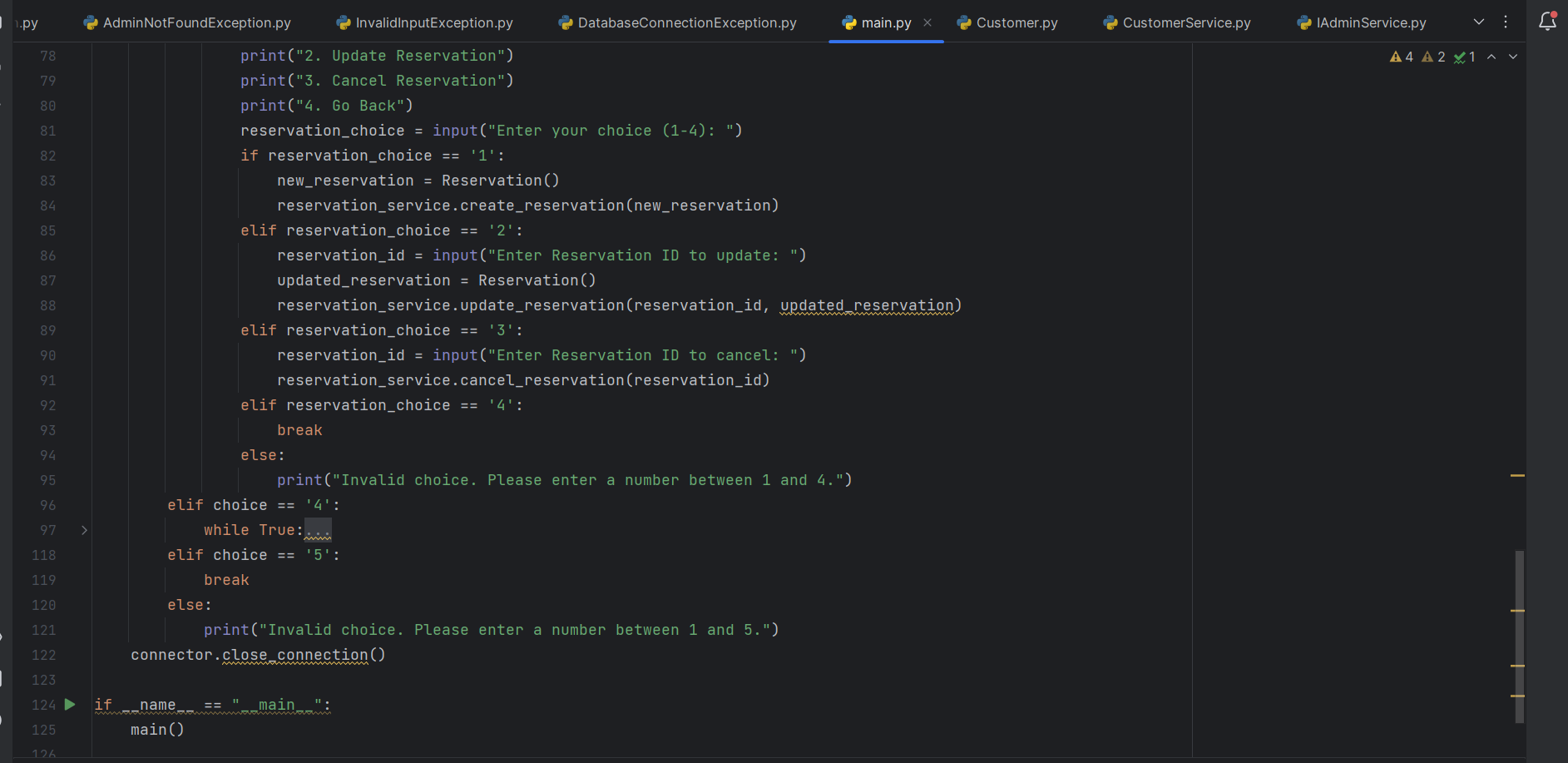


**Main.py Class**









**Interfaces:**

**ICustomerService:**

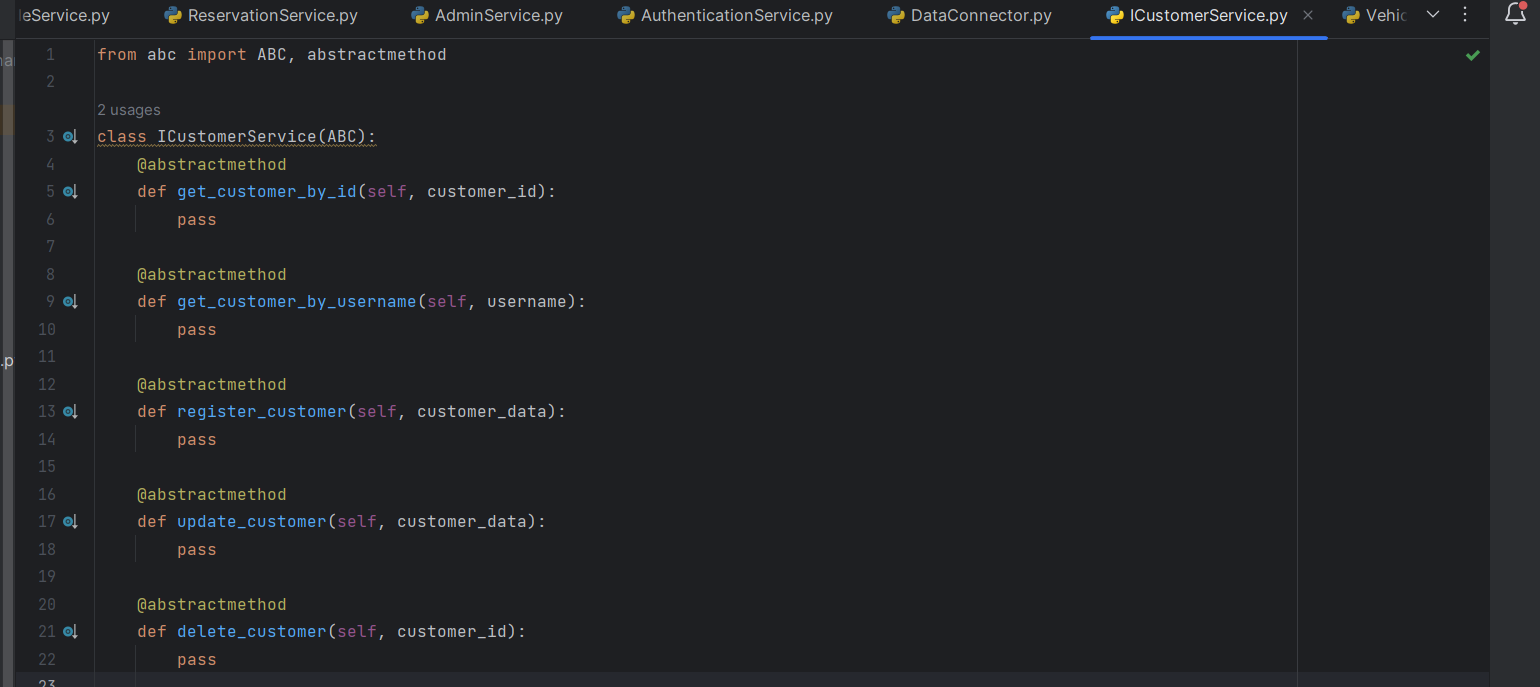
• GetCustomerById(customerId)

• GetCustomerByUsername(username)

• RegisterCustomer(customerData)

• UpdateCustomer(customerData)

• DeleteCustomer(customerId)



**IVehicleService:**

• GetVehicleById(vehicleId)

• GetAvailableVehicles()

• AddVehicle(vehicleData)

• UpdateVehicle(vehicleData)

• RemoveVehicle(vehicleId)



**IReservationService:**

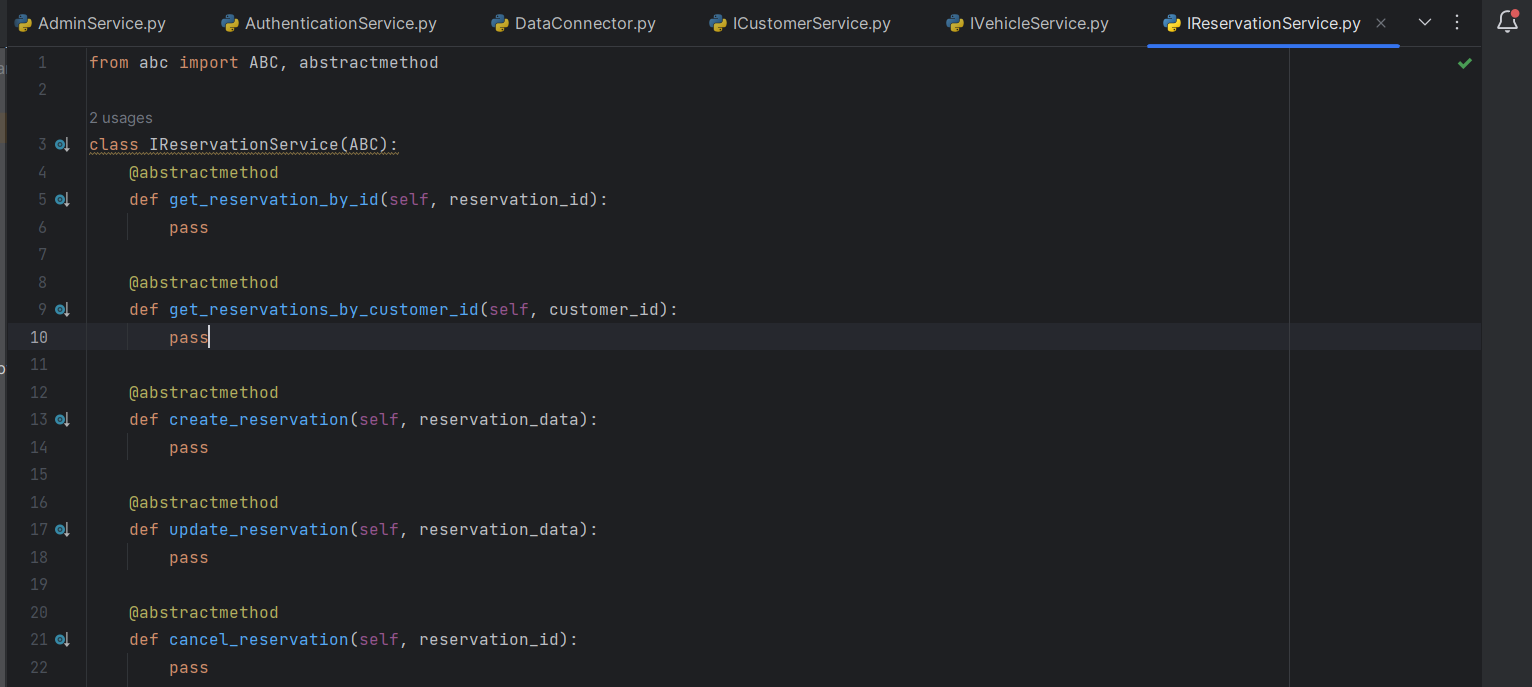
• GetReservationById(reservationId)

• GetReservationsByCustomerId(customerId)

• CreateReservation(reservationData)

• UpdateReservation(reservationData)

• CancelReservation(reservationId)



**IAdminService:**

• GetAdminById(adminId)

• GetAdminByUsername(username)

• RegisterAdmin(adminData)

• UpdateAdmin(adminData)

• DeleteAdmin(adminId)



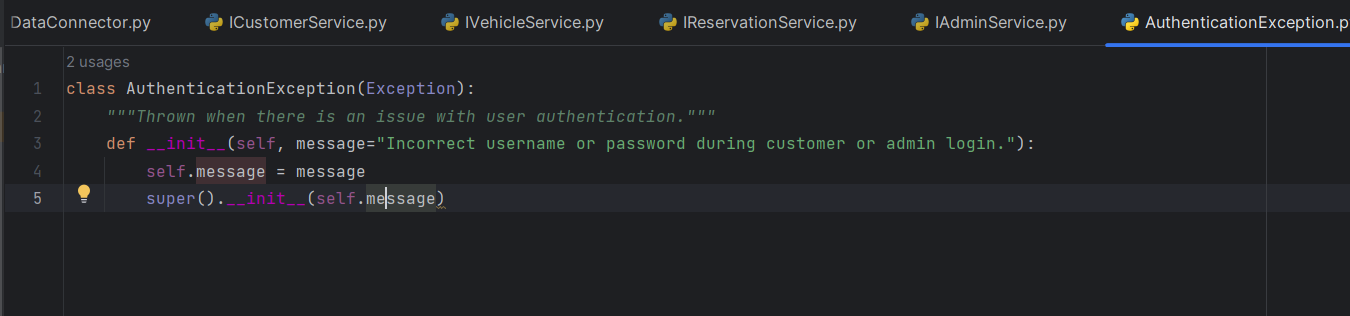
**Custom Exceptions:**

**Note: Each and every exceptions is connected to there respective modules in different classes.**

**AuthenticationException:**

• Thrown when there is an issue with user authentication.

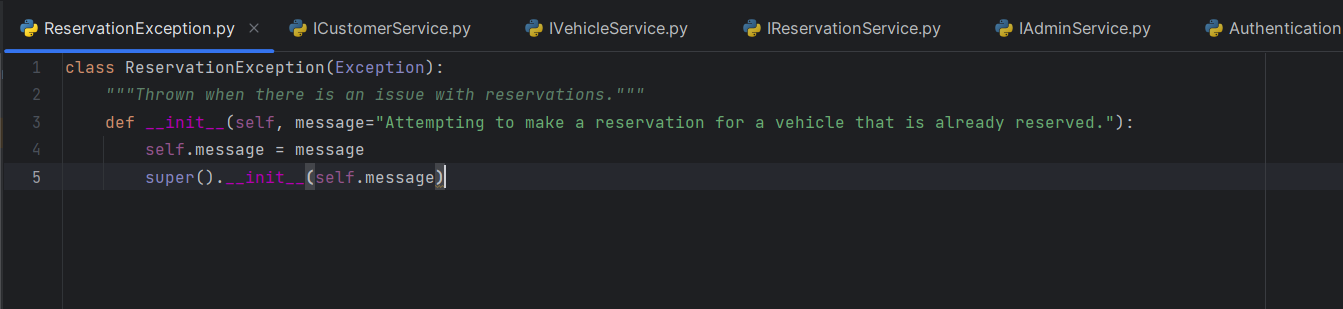
• Example Usage: Incorrect username or password during customer or admin login.

****

**ReservationException:**

• Thrown when there is an issue with reservations.

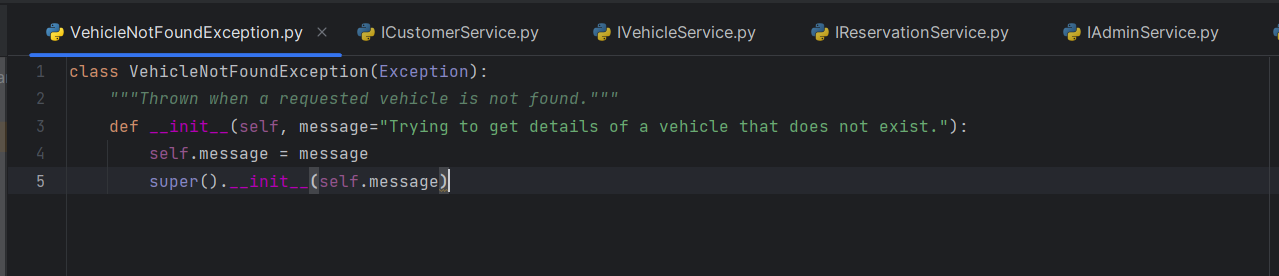
• Example Usage: Attempting to make a reservation for a vehicle that is already reserved.

****

**VehicleNotFoundException:**

• Thrown when a requested vehicle is not found.

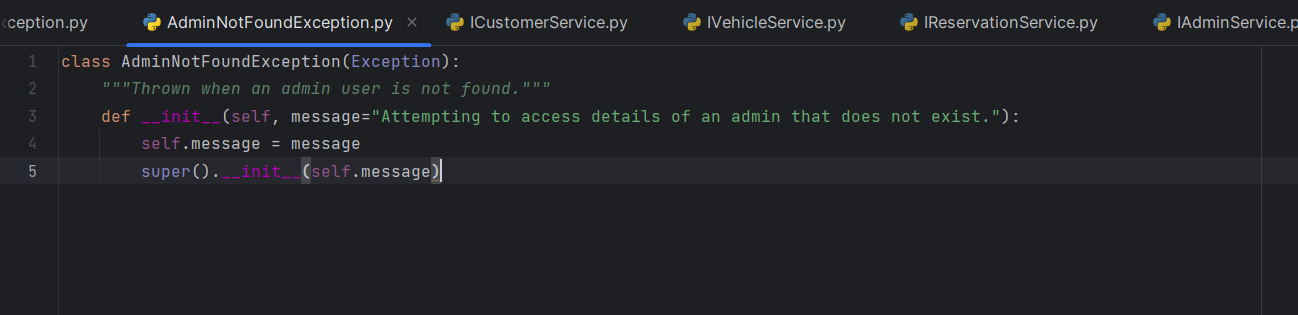
• Example Usage: Trying to get details of a vehicle that does not exist.

****

**AdminNotFoundException:**

• Thrown when an admin user is not found.

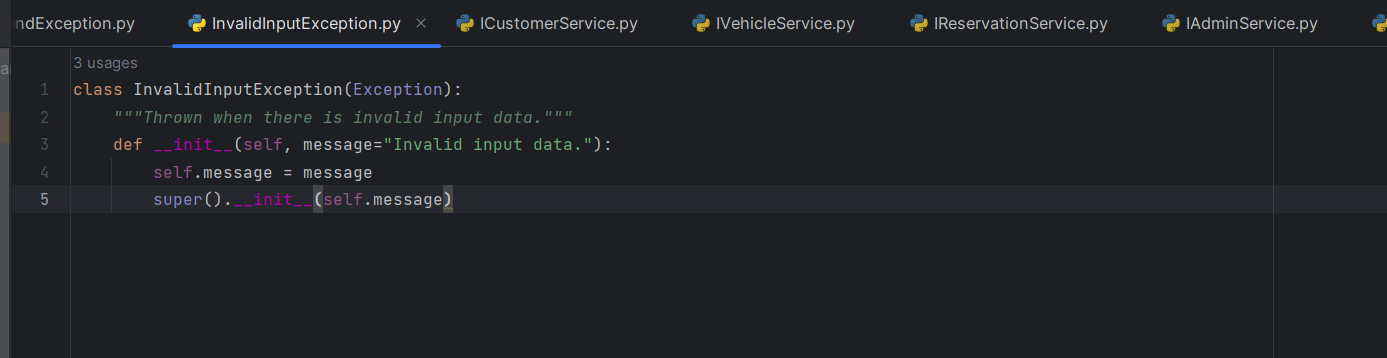
• Example Usage: Attempting to access details of an admin that does not exist.

****

**InvalidInputException:**

• Thrown when there is invalid input data.

• Example Usage: When a required field is missing or has an incorrect format.



DatabaseConnectionException:

• Thrown when there is an issue with the database connection.

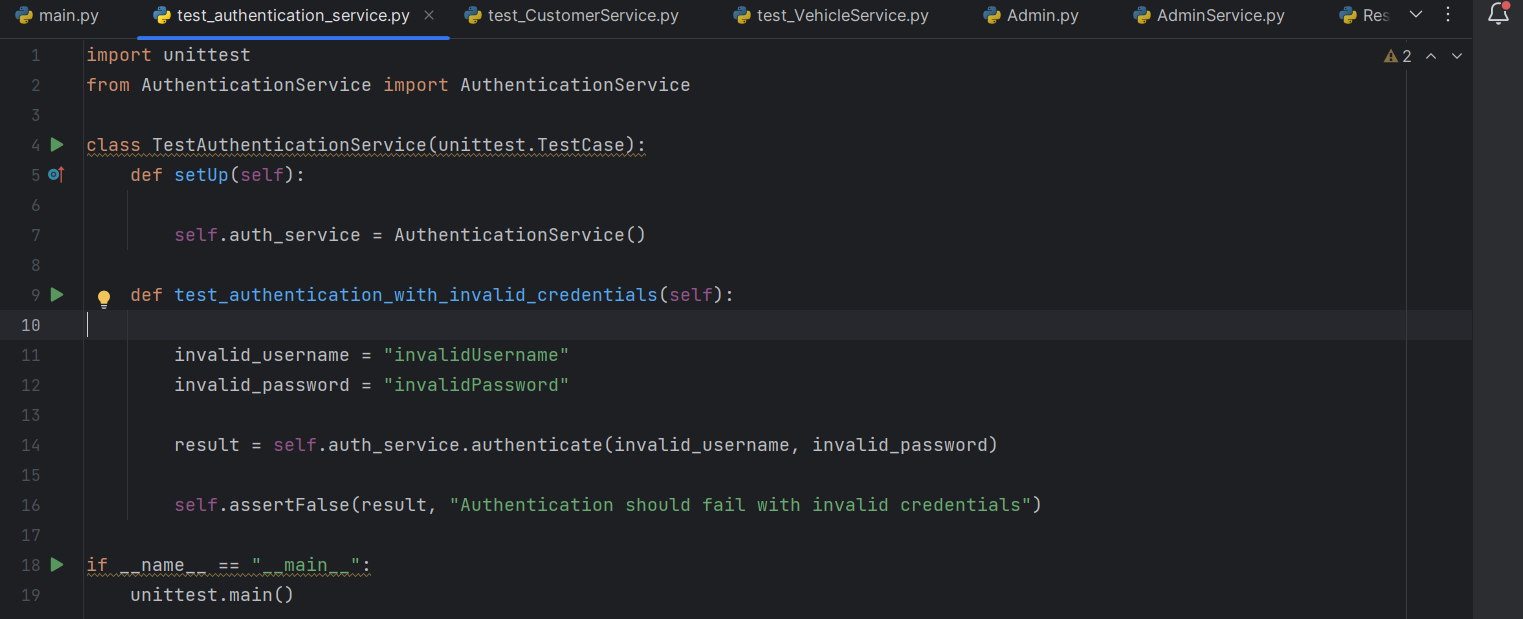
• Example Usage: Unable to establish a connection to the database.



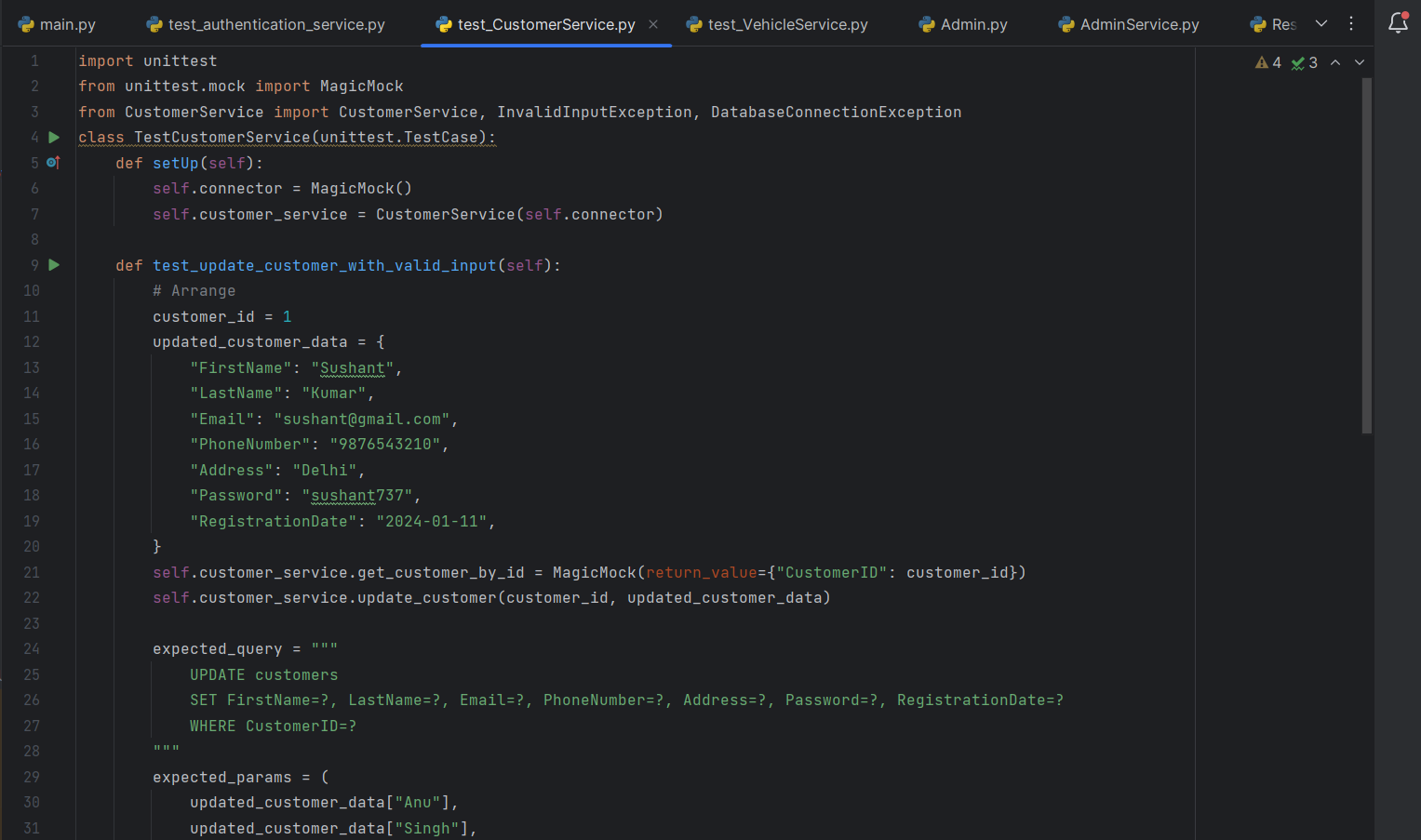
**Unit Testing:**

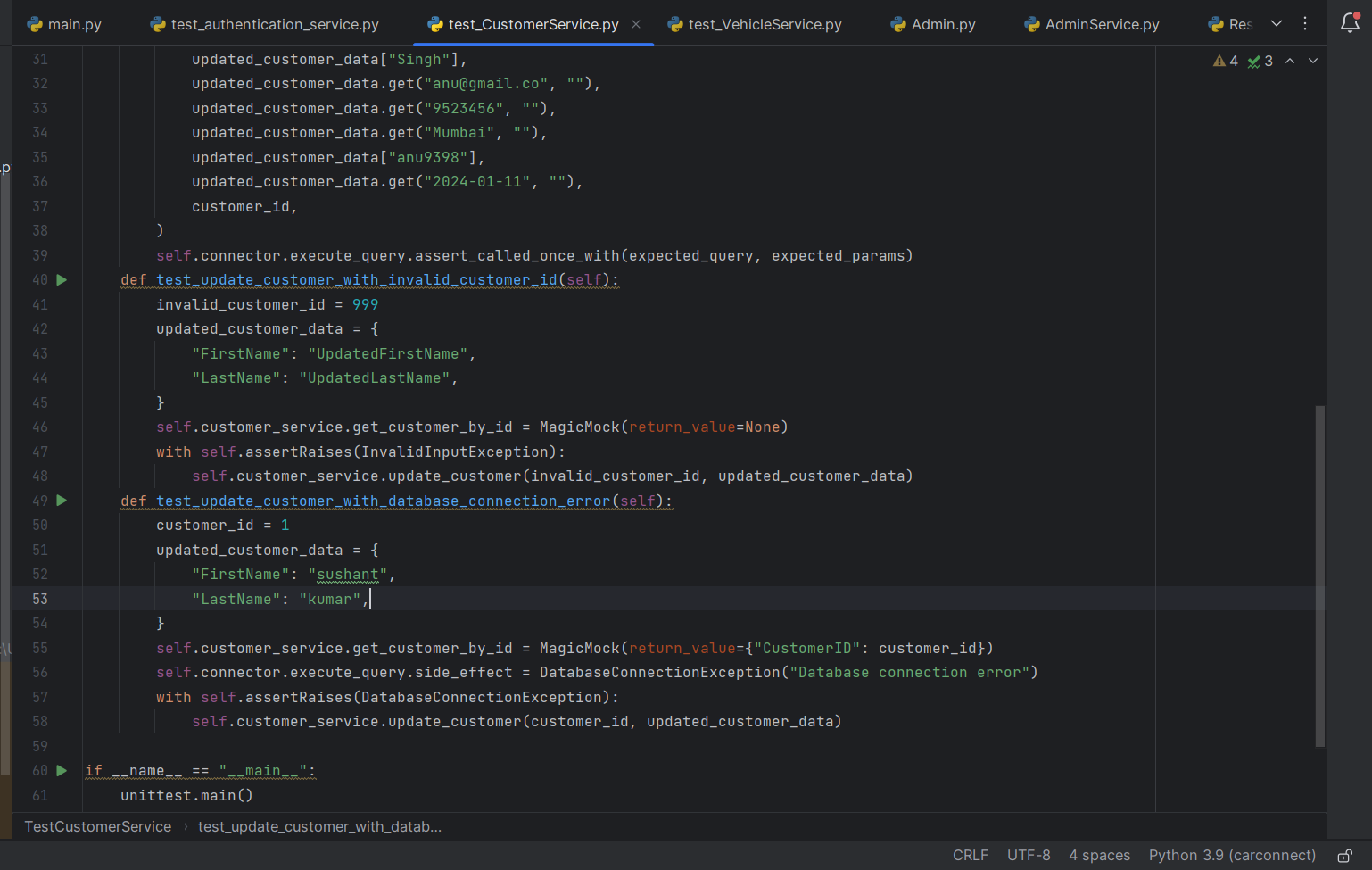
Create NUnit test cases for car rental System are essential to ensure the correctness and reliability of your system. Below are some example questions to guide the creation of NUnit test cases for various components of the system:

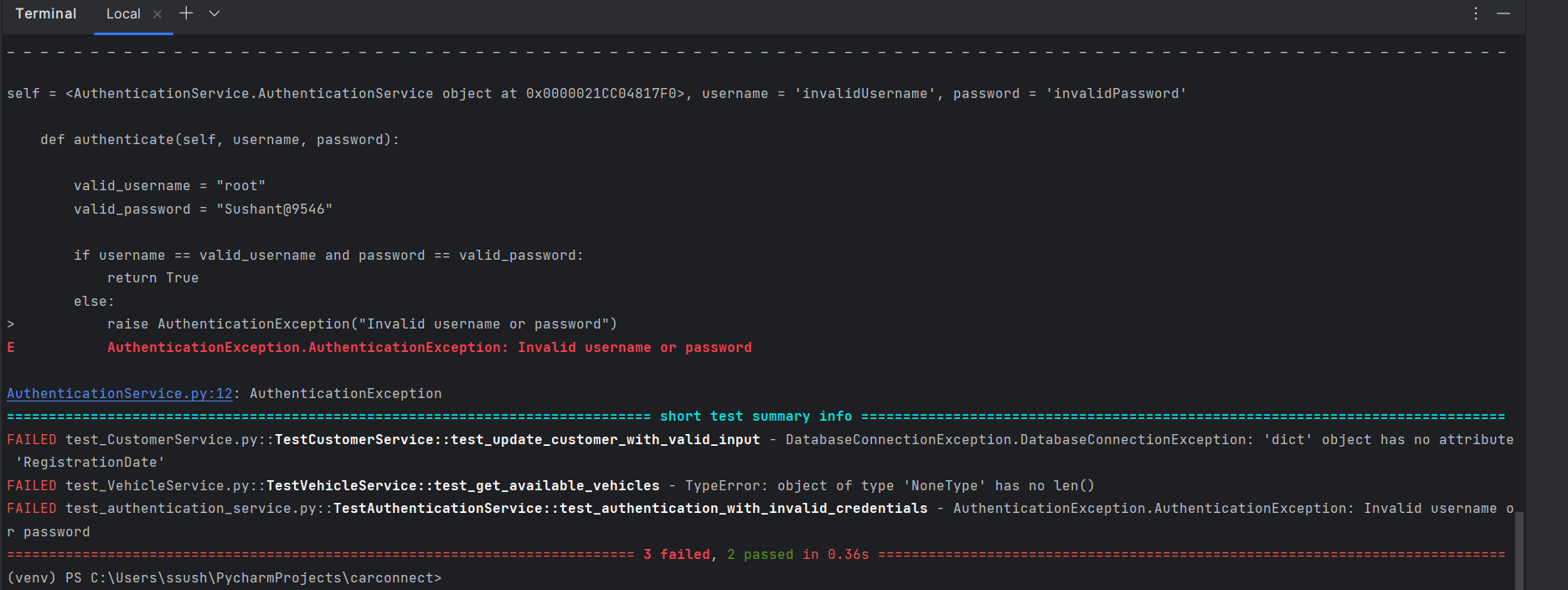
1. **Test customer authentication with invalid credentials.**



**2. Test updating customer information.**

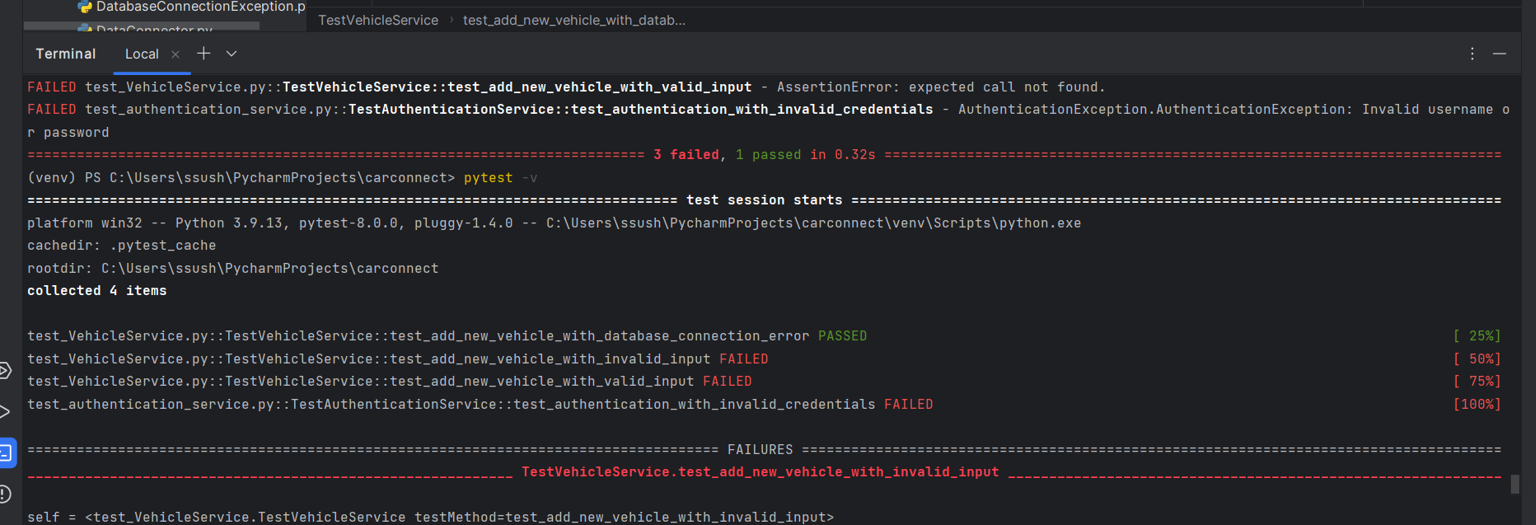




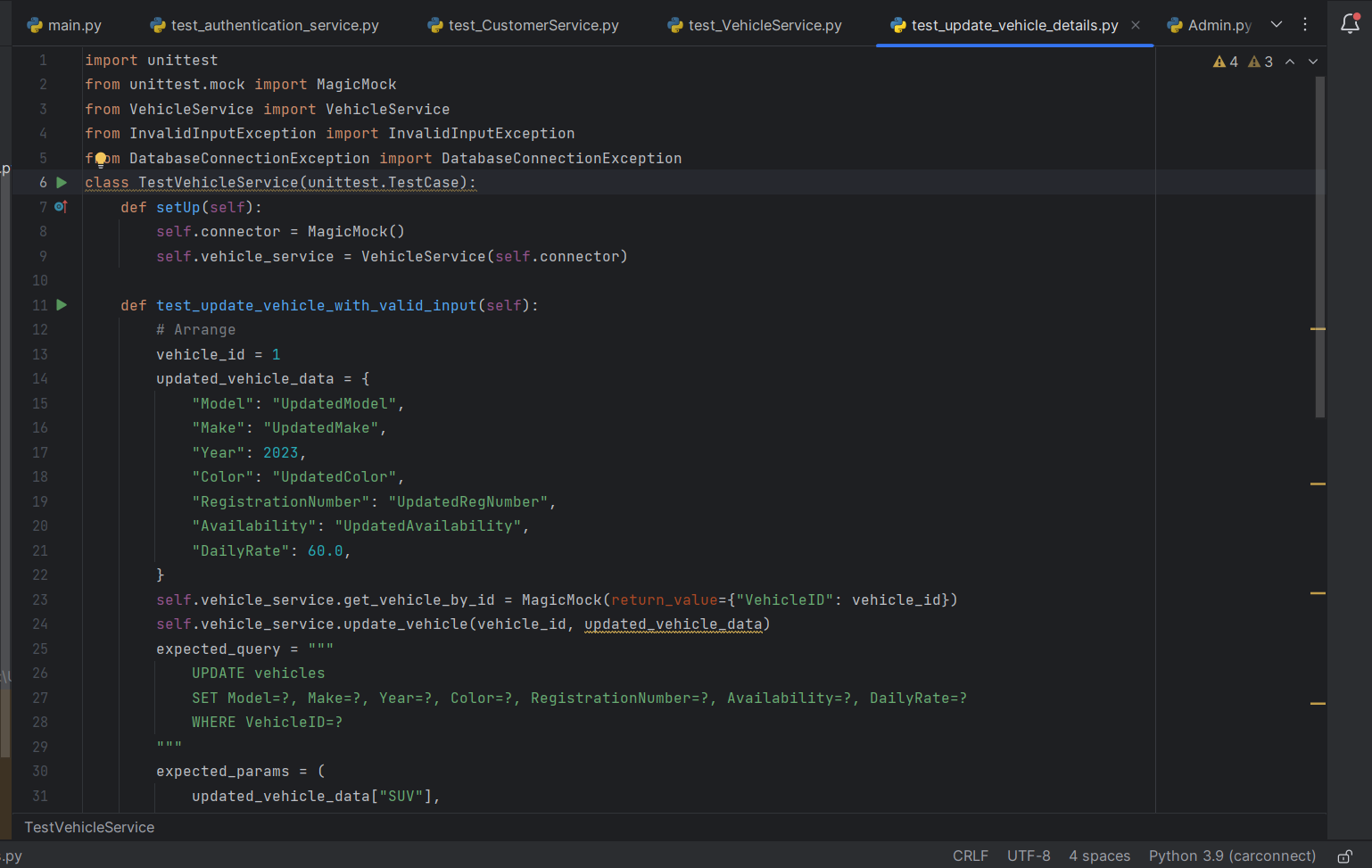


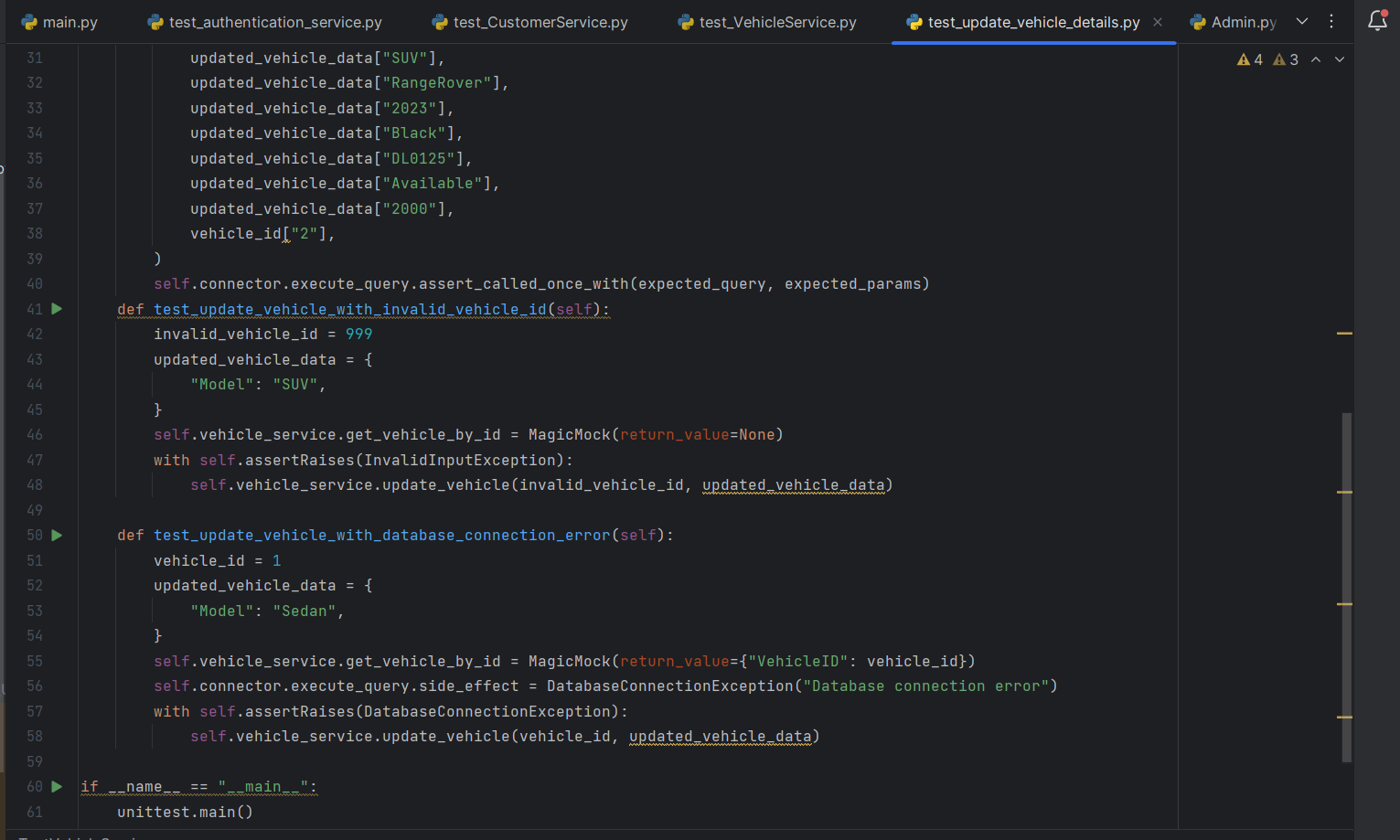
**3. Test adding a new vehicle.**





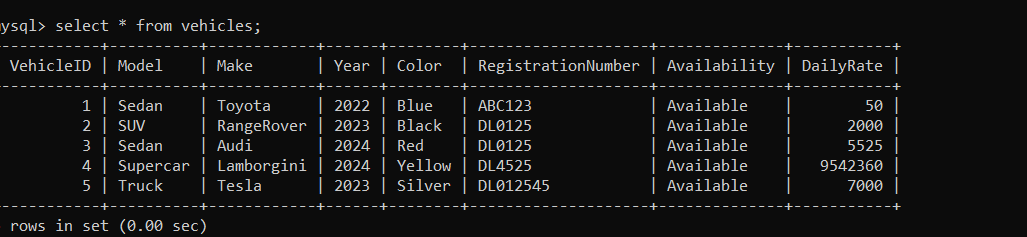
**4. Test updating vehicle details. 5. Test getting a list of available vehicles. 6. Test getting a list of all vehicles.**



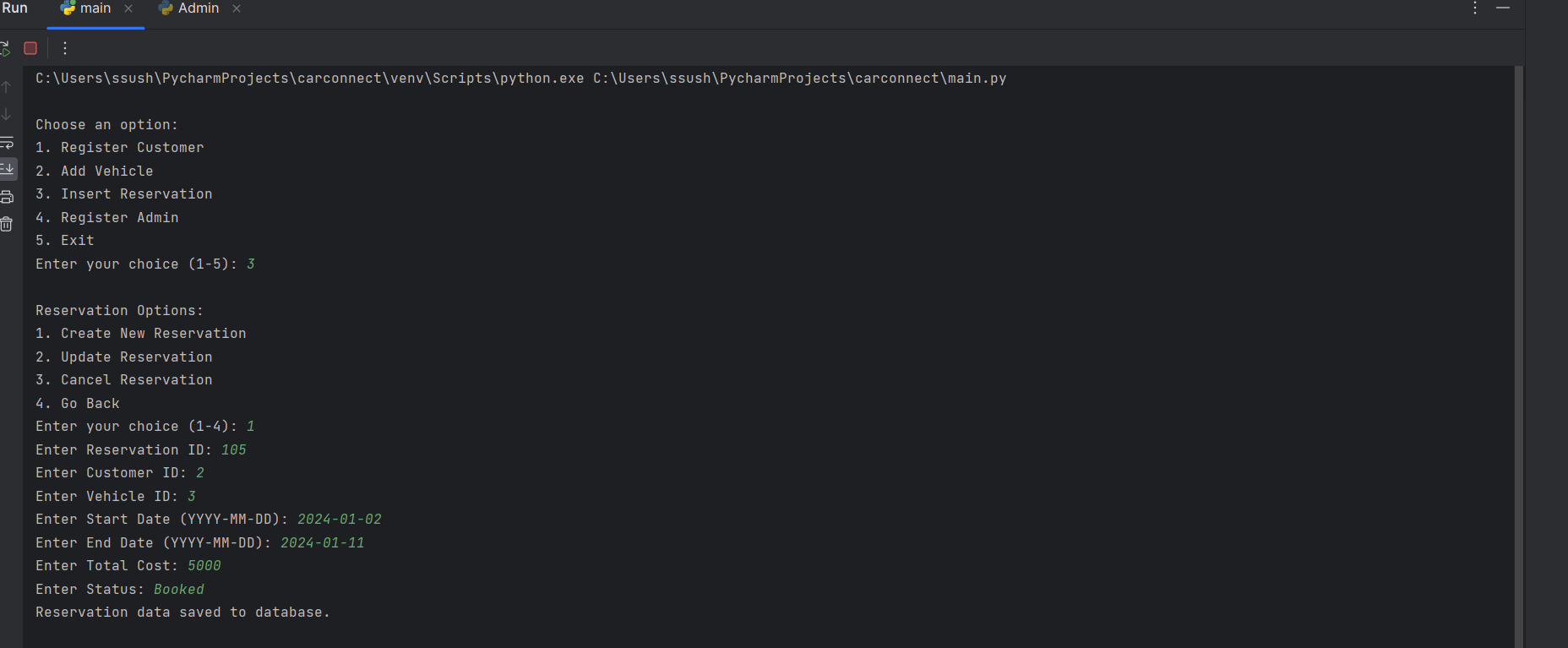


**OUTPUTS**

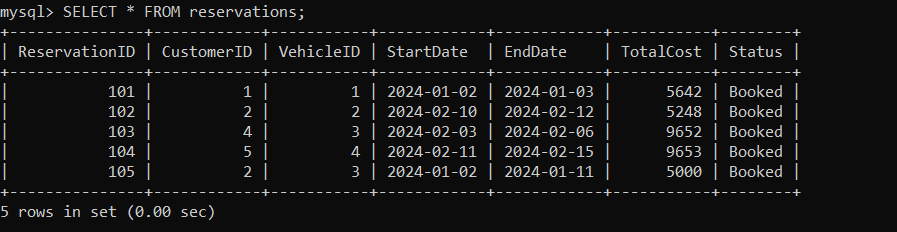


DATA SUCCESSFULLY SAVED INTO OUR DATABASE

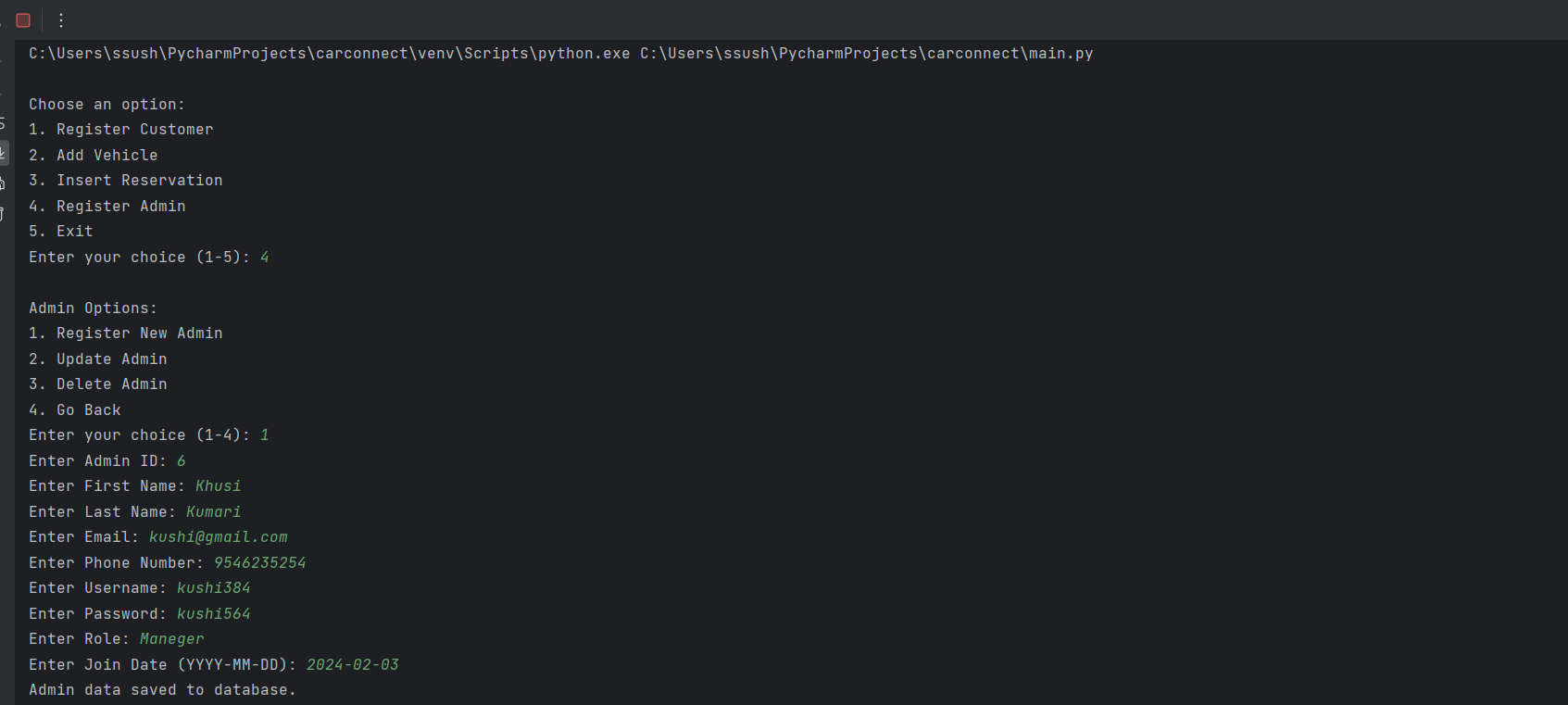
2) RESERVATION MENU



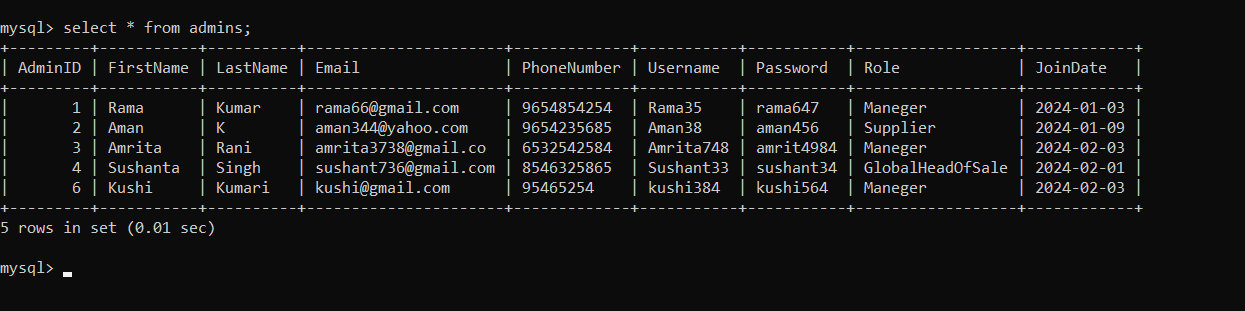
DATA SUCCESSFULLY SAVED INTO OUR RESERVATION TABLE AND DATABASE



3) ADDING AND UPDATING NEW VALUE INTO ADMIN TABLE



DATA SUCCESSFULLY SAVED INTO OUR ADMIN TABLE AND DATABASE



**\*\*\*\*\*\*\*\*\*\*ThankYou\*\*\*\*\*\*\*\*\*\***