**1. شناسایی اجزای سیستم**

سیستم مدیریت خودرو ممکن است شامل اجزای زیر باشد:

* **کاربران**: مدیران، مشتریان، مکانیک‌ها، رانندگان.
* **عملیات**: رزرو خودرو، ثبت تعمیرات، پیگیری مکان خودرو، ثبت هزینه‌ها.
* **منابع اطلاعاتی**: اطلاعات خودرو، تاریخچه تعمیرات، برنامه‌های رزرو.

**2. دیاگرام مورد کاربرد (Use Case Diagram)**

این دیاگرام نقش‌ها و تعاملات بین بازیگران (Actors) و سیستم را نشان می‌دهد.

* **بازیگران (Actors)**:
  + مشتری: درخواست رزرو خودرو یا مشاهده اطلاعات خودرو.
  + مدیر: مدیریت خودروها و گزارش‌ها.
  + مکانیک: ثبت تعمیرات یا نگهداری.
  + سیستم: مدیریت داده‌ها.
* **موارد کاربرد (Use Cases)**:
  + رزرو خودرو.
  + ثبت تعمیرات.
  + نمایش گزارش‌ها.
  + ردیابی موقعیت.

**3. دیاگرام کلاس (Class Diagram)**

این دیاگرام ساختار داده‌های سیستم را تعریف می‌کند.

**کلاس‌های اصلی:**

* **Car**:
  + ویژگی‌ها: شماره پلاک، مدل، سال تولید، وضعیت (در حال رزرو، نیاز به تعمیر).
  + متدها: تغییر وضعیت، محاسبه هزینه‌ها.
* **Customer**:
  + ویژگی‌ها: شناسه، نام، اطلاعات تماس.
  + متدها: ثبت رزرو، مشاهده سوابق.
* **Reservation**:
  + ویژگی‌ها: شناسه رزرو، تاریخ شروع و پایان، مشتری مربوطه.
  + متدها: ایجاد رزرو، کنسل کردن.
* **Maintenance**:
  + ویژگی‌ها: تاریخ تعمیر، هزینه، مکانیک.
  + متدها: ثبت تعمیرات، مشاهده گزارش.

**4. دیاگرام توالی (Sequence Diagram)**

این دیاگرام ترتیب انجام عملیات را نشان می‌دهد. به عنوان مثال، **روند رزرو خودرو**:

1. مشتری درخواست رزرو را ارسال می‌کند.
2. سیستم موجود بودن خودرو را بررسی می‌کند.
3. اگر خودرو موجود باشد، رزرو ثبت می‌شود.
4. پیام تأیید به مشتری ارسال می‌شود.

**نمونه دیاگرام (توصیفی):**

1. **Use Case**:
   * "مشتری" -> "رزرو خودرو" -> "سیستم".
2. **Class**:
   * Customer {id, name, contact} ← Reservation {id, startDate, carID}.
3. **Sequence**:
   * Customer → System: CheckAvailability → Car: Available?

کلاس ها: public class Car

{

public string LicensePlate { get; set; }

public string Model { get; set; }

public int Year { get; set; }

public bool IsAvailable { get; set; }

public Car(string licensePlate, string model, int year)

{

LicensePlate = licensePlate;

Model = model;

Year = year;

IsAvailable = true;

}

public void UpdateAvailability(bool availability)

{

IsAvailable = availability;

}

}

public class Customer

{

public int Id { get; set; }

public string Name { get; set; }

public string ContactInfo { get; set; }

public Customer(int id, string name, string contactInfo)

{

Id = id;

Name = name;

ContactInfo = contactInfo;

}

}

public class Reservation

{

public int Id { get; set; }

public Car ReservedCar { get; set; }

public Customer Customer { get; set; }

public DateTime StartDate { get; set; }

public DateTime EndDate { get; set; }

public Reservation(int id, Car car, Customer customer, DateTime startDate, DateTime endDate)

{

Id = id;

ReservedCar = car;

Customer = customer;

StartDate = startDate;

EndDate = endDate;

car.UpdateAvailability(false); // تغییر وضعیت خودرو به "رزرو شده"

}

public void CancelReservation()

{

ReservedCar.UpdateAvailability(true); // بازگرداندن وضعیت خودرو به "در دسترس"

}

}

public class Maintenance

{

public int Id { get; set; }

public Car Car { get; set; }

public DateTime MaintenanceDate { get; set; }

public double Cost { get; set; }

public Maintenance(int id, Car car, DateTime date, double cost)

{

Id = id;

Car = car;

MaintenanceDate = date;

Cost = cost;

car.UpdateAvailability(false); // خودرو برای تعمیر رزرو شده است

}

}

سرویس ها: using System.Collections.Generic;

public class CarService

{

private List<Car> Cars = new List<Car>();

public void AddCar(Car car)

{

Cars.Add(car);

}

public List<Car> GetAvailableCars()

{

return Cars.FindAll(car => car.IsAvailable);

}

public Car FindCarByLicensePlate(string licensePlate)

{

return Cars.Find(car => car.LicensePlate == licensePlate);

}

}

using System.Collections.Generic;

public class ReservationService

{

private List<Reservation> Reservations = new List<Reservation>();

public void CreateReservation(int id, Car car, Customer customer, DateTime startDate, DateTime endDate)

{

if (car.IsAvailable)

{

var reservation = new Reservation(id, car, customer, startDate, endDate);

Reservations.Add(reservation);

}

else

{

Console.WriteLine("Car is not available for reservation.");

}

}

public List<Reservation> GetReservations()

{

return Reservations;

}

}

استفاده از کلاسها: using System;

class Program

{

static void Main(string[] args)

{

// ایجاد نمونه‌ها

CarService carService = new CarService();

ReservationService reservationService = new ReservationService();

// اضافه کردن خودروها

carService.AddCar(new Car("ABC-123", "Toyota Corolla", 2020));

carService.AddCar(new Car("XYZ-456", "Honda Civic", 2021));

// نمایش خودروهای موجود

Console.WriteLine("Available Cars:");

foreach (var car in carService.GetAvailableCars())

{

Console.WriteLine($"{car.LicensePlate} - {car.Model}");

}

// ایجاد مشتری

Customer customer = new Customer(1, "Ali Reza", "09121234567");

// رزرو خودرو

Car selectedCar = carService.FindCarByLicensePlate("ABC-123");

if (selectedCar != null)

{

reservationService.CreateReservation(1, selectedCar, customer, DateTime.Now, DateTime.Now.AddDays(3));

}

// نمایش رزروها

Console.WriteLine("Reservations:");

foreach (var reservation in reservationService.GetReservations())

{

Console.WriteLine($"{reservation.Customer.Name} reserved {reservation.ReservedCar.Model}");

}

}

}