



# Дадлага Ажлын Тайлан

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
Мэдээлэл холбооны технологийн сургууль

**Хичээл:** Программчлалын дадлага (F.CSM360)  
**Хичээлийн жил:** 2024–2025 оны зун  
**Хичээл заасан багш:** Ч.Эрдэнэбат  
**Оюутан 1:** Б.Идэрбат (B242270075)  
**Оюутан 2:** Ө.Анар (B242270129)  
**Оюутан 3:** Д.Анхбаяр (B242270072)

Улаанбаатар 2025

# ERP Модуль Шинжилгээ

## Тээврийн Хэрэгсэл Түрээсийн Удирдлага

---

### Модулийн Ерөнхий Тойм

Odoo ERP-ийн “**Vehicle Rental Management**” модуль нь автомашин болон тээврийн хэрэгслийн түрээсийн процессуудыг автоматжуулах зорилготой. Энэ модуль нь гэрээний бүртгэл, төлбөр тооцоо, хэрэглэгчийн мэдээлэл, нэмэлт үйлчилгээ зэргийг цогцоор нь шийддэг.

### Зорилго

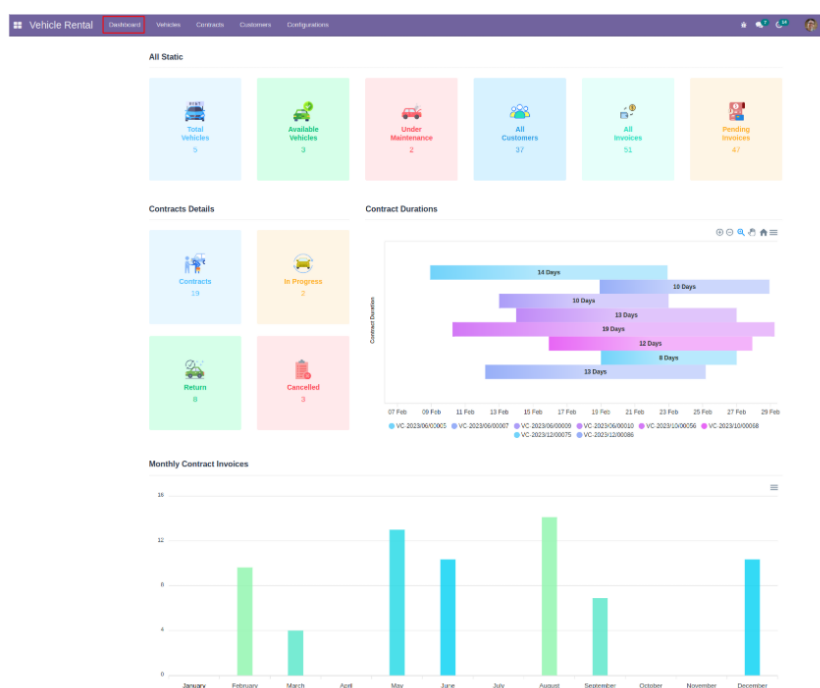
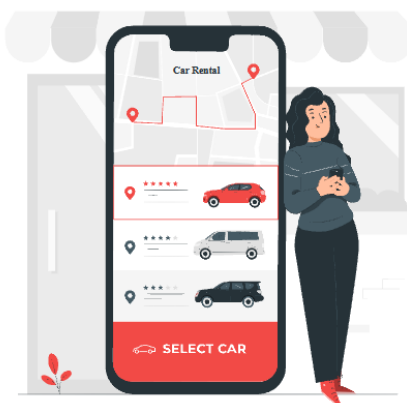
Байгууллагын тээврийн хэрэгсэл түрээсийн үйл ажиллагааг хялбар, хяналттай, ил тод болгох. Түрээсийн гэрээг байгуулах, хянах, төлбөрийн менежмент хийх, тайлан гаргах зэрэг бүх процессыг системчлэх.

### Хэрэглэгчийн Шаардлага

- Хялбар, ойлгомжтой захиалгын интерфэйстэй байх
- Түрээсийн гэрээг уян хатан нөхцлөөр байгуулах (өдөр, сар, жил г.м)
- Урьдчилгаа, хэсэгчлэн эсвэл бүрэн төлбөр хийх боломжтой
- Хохирлын бүртгэл, алдагдлын нөхөн төлбөрийн ил тод байдал
- Тайлан, гэрээг PDF-ээр татаж авах, хэвлэх боломж
- И-мэйл мэдэгдэлтэй байх (баталгаажуулалт, сануулга г.м)

### Гол Онцлогууд

- **Хяналтын Самбар:** Түрээсийн тоо, төлөв, орлого, графикуудыг харуулна
- **Түрээсийн төрөл:** Өдөр, сар, жил, улирал гэх мэт хугацааны сонголт
- **Төлбөрийн төрөл:** Хэсэгчлэн эсвэл бүхлээр нь төлөх
- **Автомат нэхэмжлэл:** Төлбөрийн хугацаанд үндэслэн үүсгэгддэг
- **Хохирлын бүртгэл:** Эвдрэл, гэмтлийн талаар тайлан гаргана
- **Урьдчилгаа төлбөр:** Захиалгыг баталгаажуулах барьцаа
- **Даатгал, цуцлалт:** Гэрээтэй холбогдсон хамгаалалт, уян хатан цуцлалт
- **Нэмэлт үйлчилгээ:** GPS, жолооч, нэмэлт төлбөр зэргийг сонгох
- **Fleet Integration:** Сул, ажиллаж буй болон засвартай машин хянах



Зураг 1: Тээврийн хэрэгсэл түрээсийн хяналтын самбар

## Дүгнэлт

“Vehicle Rental Management” модуль нь тээврийн хэрэгслийн түрээсийн системийг цахимжуулж, байгууллагын менежмент, хэрэглэгчийн үйлчилгээ, санхүүгийн хяналтыг сайжруулахад чиглэсэн хүчирхэг шийдэл юм. Бүх үйл явцыг нэг системд багтаан, хэрэглэгч төвтэй дижитал платформ болгон хөгжүүлжээ.

## Кодын тайлбар:

### Vehicle классын тайлбар

- **abstract class Vehicle** — Энэхүү класс нь **abstract** буюу абстракт төрөл бөгөөд шууд объект үүсгэх боломжгүй. Өөрөөр хэлбэл зөвхөн энэ классыг өргөтгөсөн (**extends**) дэд классууд ашиглаж, тодорхой функцуудыг хэрэгжүүлэх шаардлагатай.
- **String brand;** — Тээврийн хэрэгслийн брэнд (жишээлбэл: Toyota, BMW гэх мэт).
- **double price;** — Тээврийн хэрэгслийн үнэ.
- **Vehicle(String brand, double price)** — Классын конструктор бөгөөд объект үүсгэх үед брэнд болон үнийг онооно.
- **abstract double calculateRentalCost(int days, boolean insurance);** — Түрээсийн зардлыг тооцоолох абстракт метод. Энэ нь тухайн классыг өвлөж буй дэд классууд тус тусын онцлогт нийцүүлэн хэрэгжүүлэх ёстой функц юм.
- **public String getBrand()** — Тухайн унааны брэндийг буцаах зориулалттай **getter** функц.

### Зорилго

Энэхүү **Vehicle** класс нь бүх төрлийн тээврийн хэрэгслүүдэд нийтлэг байх шинж чанар (**brand**, **price**) болон үйлдлийг (**calculateRentalCost**) тодорхойлсон суурь загвар юм. Үүнийг **Car**, **Truck**, **Motorcycle** зэрэг классууд өргөтгөн, тус бүрийн онцлогт тохируулан хэрэгжүүлдэг.

```
1  abstract class Vehicle {
2  String brand;
3  double price;
4
5  public Vehicle(String brand, double price) {
6      this.brand = brand;
7      this.price = price;
8  }
9
10 abstract double calculateRentalCost(int days, boolean insurance
    );
11
12 public String getBrand() {
13     return brand;
14 }
15 }
```

## Car Классын (Vehicle классыг өргөтгөсөн) тайлбар

- **class Car extends Vehicle** — Car класс нь Vehicle классыг өргөтгөж байгаа бөгөөд тээврийн хэрэгслийн нэг төрөл болох автомашиныг төлөөлнө.
- **public Car(String brand, double price)** — Конструктор функц нь эцэг Vehicle классын конструкторыг **super** түлхүүр үгээр дамжуулан дуудаж байна. Ингэснээр автомашины брэнд ба үнэ оноогдоно.
- **double calculateRentalCost(int days, boolean insurance)** — Түрээсийн нийт зардлыг тооцоолох функц. Түрээсийн зардал нь дараах байдлаар тооцогдоно:
  - Энгийн зардал: өдөр тутмын үнэ нь тухайн автомашины **price** хувьсагчаар үржигдэнэ.
  - Хэрэв даатгал авбал өдөрт 2500 төгрөгөөр нэмэгдэнэ.
- Энэ функц нь Vehicle класст тодорхойлогдсон **abstract calculateRentalCost()** функцийг автомашинд тохируулан хэрэгжүүлж байгаа хэрэг юм.

## Жишээ хэрэглээ

Жишээлбэл:

```
1 Car car = new Car("Toyota", 30000);
2 double cost = car.calculateRentalCost(3, true); // 3
```

Дээрх жишээнд 'Toyota' брэндийн автомашиныг 3 хоногоор даатгалтайгаар түрээслэхэд:

$$\text{Нийт зардал} = 30000 \times 3 + 2500 \times 3 = 97500 \text{ төгрөг}$$

```
1 class Car extends Vehicle {
2   public Car(String brand, double price) {
3     super(brand, price);
4   }
5
6   double calculateRentalCost(int days, boolean insurance) {
7     double cost = price * days;
8     if (insurance) cost += 2500 * days;
9     return cost;
10  }
11 }
```

## Motorcycle Классын (Vehicle классыг өргөтгөсөн) тайлбар

- **class Motorcycle extends Vehicle** — Motorcycle класс нь Vehicle классыг өргөтгөсөн бөгөөд энэ нь мотоциклын төрлийн тээврийн хэрэгслийг төлөөлнө.
- **public Motorcycle(String brand, double price)** — Конструктор функц нь брэнд болон үнийг эцэг Vehicle класст дамжуулан оноож байна. Энэ нь **super** түлхүүр үгээр гүйцэтгэгддэг.

- `double calculateRentalCost(int days, boolean insurance)` — Түрээсийн зардлыг дараах байдлаар тооцно:
  - Энгийн зардал: өдрийн үнэ `price` хувьсагчаар үржигдэнэ.
  - Хэрэв даатгал авбал өдөр бүр 1750 төгрөгөөр нэмэгдэнэ.
- Энэ функц нь `Vehicle` классын `abstract calculateRentalCost()` функцийг мотоциклд зориулан хэрэгжүүлж байна.

## Жишээ хэрэглээ

Жишээ:

```
1 Motorcycle moto = new Motorcycle("Yamaha", 18000);
2 double cost = moto.calculateRentalCost(2, true); // 2 ,
```

Энэ тохиолдолд `Yamaha` брэндийн мотоциклийг 2 хоногийн хугацаанд даатгалтайгаар түрээслэхэд:

$$\text{Нийт зардал} = 18000 \times 2 + 1750 \times 2 = 39500 \text{ төгрөг}$$

```
1 class Motorcycle extends Vehicle {
2     public Motorcycle(String brand, double price) {
3         super(brand, price);
4     }
5
6     double calculateRentalCost(int days, boolean insurance) {
7         double cost = price * days;
8         if (insurance) cost += 1750 * days;
9         return cost;
10    }
11 }
```

## Тээврийн Хэрэгслийн Түрээсийн Системийн Үндсэн Классын тайлбар (VehicleRentalSystem)

- Энэ класс нь хэрэглэгчтэй харилцаж, тээврийн хэрэгслийн сонголт, түрээсийн хоног болон даатгалын мэдээлэлд үндэслэн нийт зардлыг тооцоолох үндсэн системийг агуулдаг.
- `Scanner scanner = new Scanner(System.in);` — Хэрэглэгчийн оролтыг авахын тулд `Scanner` ашиглаж байна.
- `List<Car>` болон `List<Motorcycle>` — Машин болон мотоциклийн жагсаалтуудыг тус тус үүсгэж, хэд хэдэн жишээ өгөгдөл нэмж байна.
- `if (typeChoice == 1)` — Хэрэглэгч 1-ийг сонговол автомашины жагсаалтыг харуулна. 2-ыг сонговол мотоциклын жагсаалтыг харуулна.

- Хэрэглэгч сонголтоо хийсний дараа тухайн сонгосон объектыг `selectedVehicle` хувьсагчид онооно.
- Түрээсийн өдрүүд болон даатгал байгаа эсэхийг асууж, нийт түрээсийн зардлыг дараах байдлаар тооцно:

`selectedVehicle.calculateRentalCost(days, insurance);`

- Эцэст нь тухайн унааны нэр, өдрийн тоо, даатгалын төлөв, нийт зардлыг харуулна.

## Жишээ Гүйлгээ

Хэрэв хэрэглэгч:

- Тээврийн хэрэгсэл: Toyota Prius
- Түрээсийн хоног: 2
- Даатгал: yes

Тэгвэл нийт зардал дараах байдлаар тооцогдоно:

$$\text{Нийт зардал} = 150000 \times 2 + 2500 \times 2 = 305000$$

```

1 import java.util.ArrayList;
2 import java.util.List;
3 import java.util.Scanner;
4
5 public class VehicleRentalSystem {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         List<Car> cars = new ArrayList<>();
10        cars.add(new Car("Toyota Aqua", 100000));
11        cars.add(new Car("Toyota Prius", 150000));
12        cars.add(new Car("Toyota Alphard", 250000));
13
14        List<Motorcycle> motorcycles = new ArrayList<>();
15        motorcycles.add(new Motorcycle("Honda CB500", 150000));
16        motorcycles.add(new Motorcycle("Yamaha MT-07", 200000));
17        motorcycles.add(new Motorcycle("KTM Duke 390", 300000));
18
19        System.out.println("*****Vehicle rental system*****");
20        System.out.println("Select vehicle type: 1. Car 2. Moto");
21        int typeChoice = scanner.nextInt();
22
23        Vehicle selectedVehicle = null;
24
25        if (typeChoice == 1) {
26            System.out.println("Cars:");
27            for (int i = 0; i < cars.size(); i++) {

```

```

28         System.out.println((i + 1) + ". " + cars.get(i).
           getBrand() + " " + cars.get(i).price + "$" + "/day")
           ;
29     }
30     System.out.print("Select (1-" + cars.size() + "): ");
31     int carChoice = scanner.nextInt();
32     selectedVehicle = cars.get(carChoice - 1);
33
34 } else if (typeChoice == 2) {
35     System.out.println("Motos:");
36     for (int i = 0; i < motorcycles.size(); i++) {
37         System.out.println((i + 1) + ". " + motorcycles.get
           (i).getBrand() + " " + motorcycles.get(i).price
           + "$" + "/day");
38     }
39     System.out.print("Select (1-" + motorcycles.size() + "
       : ");
40     int motoChoice = scanner.nextInt();
41     selectedVehicle = motorcycles.get(motoChoice - 1);
42
43 } else {
44     System.out.println("Invalid. Try again");
45     return;
46 }
47
48 System.out.print("Enter number of rental days?");
49 int days = scanner.nextInt();
50 scanner.nextLine();
51
52 System.out.print("Insurance? (yes/no): ");
53 String insuranceInput = scanner.nextLine();
54 boolean insurance = insuranceInput.equalsIgnoreCase("yes");
55
56 double totalCost = selectedVehicle.calculateRentalCost(days
   , insurance);
57
58 System.out.println("\n*****Paymenmt*****");
59 System.out.println("Vehicle: " + selectedVehicle.getBrand()
   );
60 System.out.println("Day: " + days);
61 System.out.println("Insurance: " + (insurance ? "yes" : "no
   "));
62 System.out.println("Total cost: " + totalCost + "$");
63 }
64 }

```

## Жишээ Оролт ба Гаралт

### Оролт



```
1 *****Vehicle rental system*****
2 Select vehicle type: 1. Car  2. Moto
3 1
4 Cars:
5 1.  Toyota Aqua 100000.0$/day
6 2.  Toyota Prius 150000.0$/day
7 3.  Toyota Alphard 250000.0$/day
8 Select (1-3): 2
9 Enter number of rental days? 3
10 Insurance? (yes/no): yes
```

Listing 1: Жишээ оролт

## Гаралт

```
1 *****Paymentt*****
2 Vehicle: Toyota Prius
3 Day: 3
4 Insurance: yes
5 Total cost: 472500.0$
```

Listing 2: Жишээ гаралт