

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

SİSTEM ANALİZİ VE TASARIMI
DÖNEM PROJESİ
YTU KARGO TAKİP SİSTEMİ

18011612 – AHMET UMUT ALKAN
17011025 – FURKAN AKÇAKAYA
15011040 – MUHAMMED KAĞAN ÜLKER
16011058 – OSMAN DOĞUKAN GÜNAYDIN
16011010 – FURKAN NECATİ ÜRKMEZ

DANIŞMAN ÖĞRETMEN
PROF. DR. OYA KALIPSIZ

PROJENİN AMACI

Müşterilerin iletilecek kargolarının teslim alımı, iletimi ve takibinin yapılmasıdır. Kargonun durumu ile ilgili müşterilere güncel takip imkanı sağlanmasıdır.

ÖN İNCELEME

1. Tüm şubelerin kargo hakkında bilgiye sahip olması sağlanır.
2. İletilecek kargo bilgileri alınır, sisteme girilir ve fiyatlandırması yapılır.
3. Tahmini teslim süresi hesaplanır.
4. Kargo durum takip edilir.
5. Teslimat sırasında herhangi bir durum oluşursa bu durumlar tespit edilir ve geri bildirimi sağlanır.

FİZİBİLİTE

1. Teknik Fizibilite

Sistem masaüstü uygulamalarda popüler olan ve platform bağımsızlığına sahip olduğu için Java ile geliştirilmiştir.

Sistemin geliştirilebilmesi için gerekli minimum donanımsal özellikler şunlardır:

RAM: 128 MB
Çekirdek Sayısı: 1
Disk Boyutu: 32 GB
CPU: 2.266 MHz

2. Zaman Fizibilitesi

Projenin tamamlanması 12 hafta sürer.

3. Yasal Fizibilite

Bu bölümde projenin gerekli olan yasalara uygunluğu ve yapımında kullanılan programların gerekli lisans izinleri incelenecektir.

Proje yapımında 2393 - Fikir ve Sanat Eserleri Kanunu'nda belirtilen hususlar dikkate alınmış olup sistemimize benzer özellikler taşıyan diğer sistemlerin çıkarları korunmuştur.

Projemiz kargo takip sistemi olup gerekli yönetmelik olan 27255 numaralı Resmi Gazete'de yayınlanan Karayolu Taşıma Yönetmeliği dikkate alınarak yapılmıştır.

4. Ekonomik Fizibilite

Sistemin geliştirilmesi için gerekli yazılımlar ücretsiz temin edilmiş olup, sistemin geliştirildiği bilgisayar 1500 TL, sistemi geliştiren çalışanlara verilen toplam ücret 1440 TL'dir. Sistem toplamda 2940 TL'ye mal olmuştur.

SİSTEM ANALİZİ VE MİMARİ TASARIMI

Bu projede; kargolar için müşterilerin güncel kargo bilgilerine erişmesi, kargo gönderirken ne kadar ücret ödemesi gerektiğini öğrenmesi amaçlanmaktadır. Sistemi oluştururken basit bir şekilde kargonun güncel konumunu, gönderilmek istenen kargoya dair bilgileri en hızlı şekilde kontrol edilebilmesini sağlamaktır. Bunları yapabilmek için hakim olunması gereken başlıca konu ise Java. Takip eden bölümlerde sistemle ilgili detaylar ve bağlantılar mevcuttur.

1. Sistemin Oluşturulması

Java ile yapılması planlanan bu projede kullanılacak programların kurulumları yapıldıktan sonra projeye başlangıç için yapılan zaman fizibilite raporuna göre sistem analizi ve tasarımlarına başlanmış olunacaktır.

1.1. Eclipse ile Proje Oluşturulması

Yapmaya karar verdiğimiz projede IDE olarak Eclipse kullanılması uygun görülmüştür.

2. Yazılım Tasarımı

Yazılım tasarımını yaparken iki tane konu başlığı bulunmakta olup bunlar;

1.Arayüz Tasarımı (Front End Development)

2. Fonksiyonların Yazılması (Back End Development)

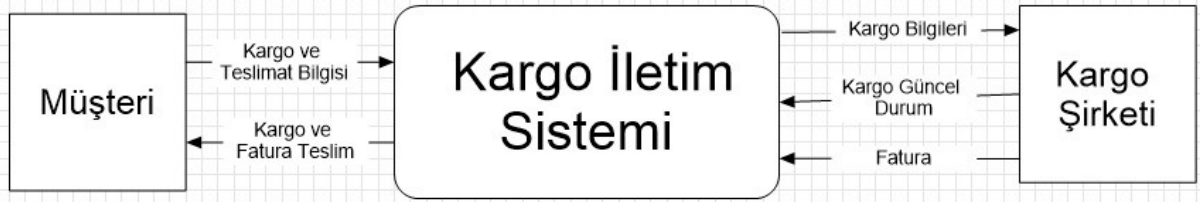
2.1. Arayüz Tasarımı

Arayüz tasarımı için önceliğimiz kullanım kolaylığı oldu.

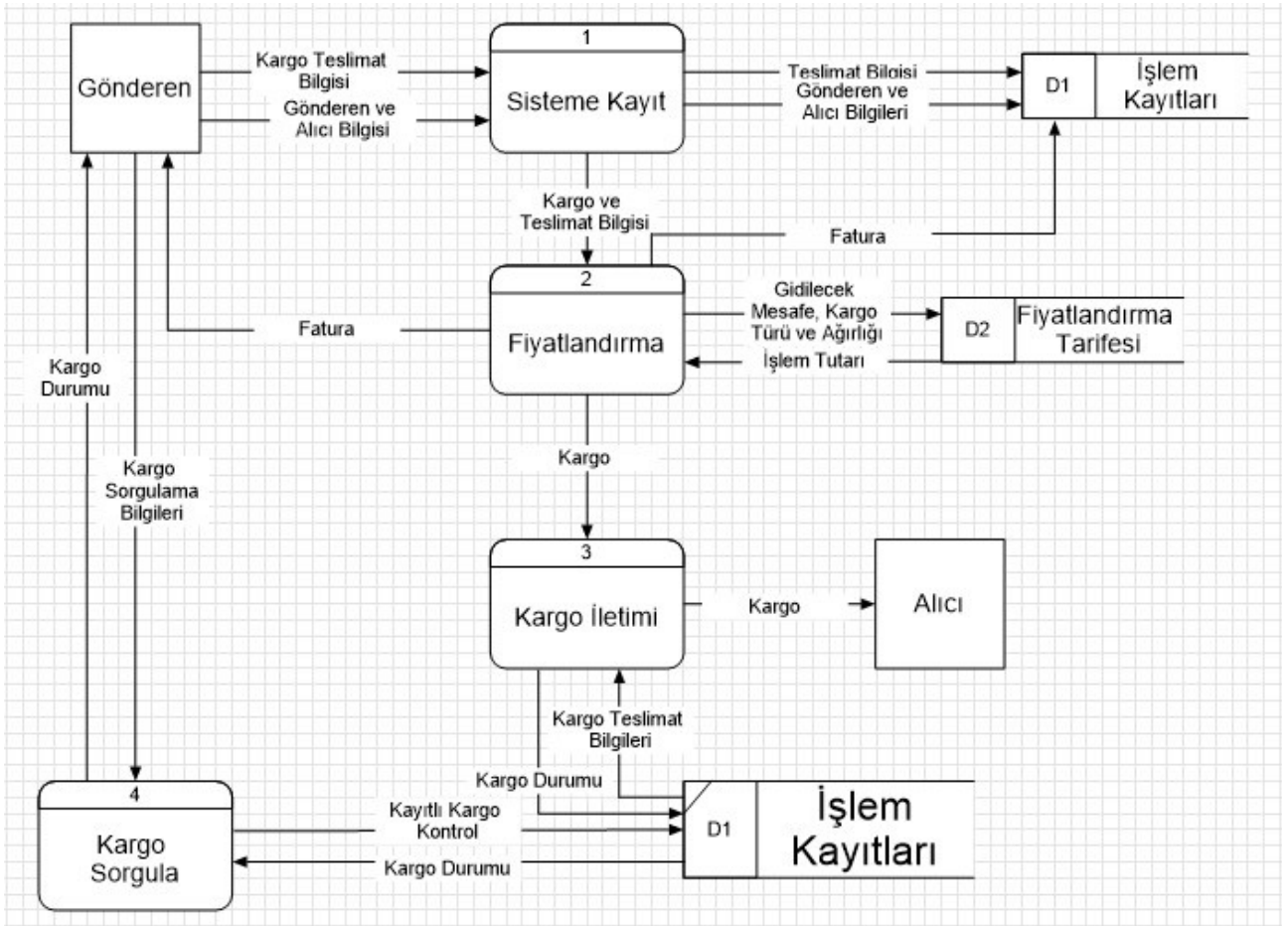
2.2. Fonksiyonların Yazılması

Fonksiyonlarımızı projenin fikir safhasında belirlemiş olduğumuz için direkt olarak UML-Kod uyumlu bir şekilde ilerledik. Tüm fonksiyonlara ve detaylarına ilerleyen bölümlerde yer verdik.

Taslak Data Flow



Data Flow (1. Düzey)



2.2.1. addCargo

Sisteme kargo eklemeye yarıyor.

2.2.2. findCargo

Sistemdeki kargoyu bulmaya yarıyor.

2.2.3. deleteCargo

Sistemden kargo silmeye yarıyor.

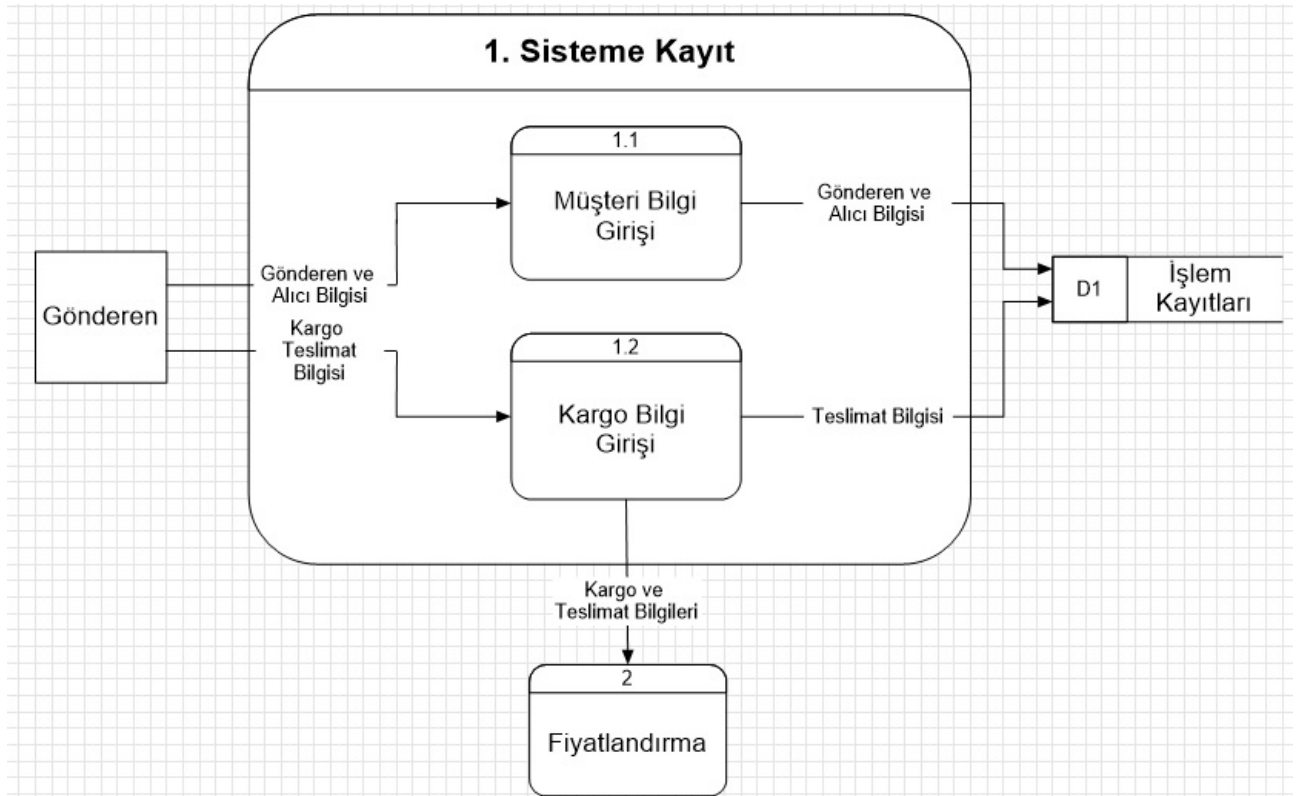
2.2.4. findTime

Sisteme eklenen bir kargonun tahmini varış süresini hesaplıyor.

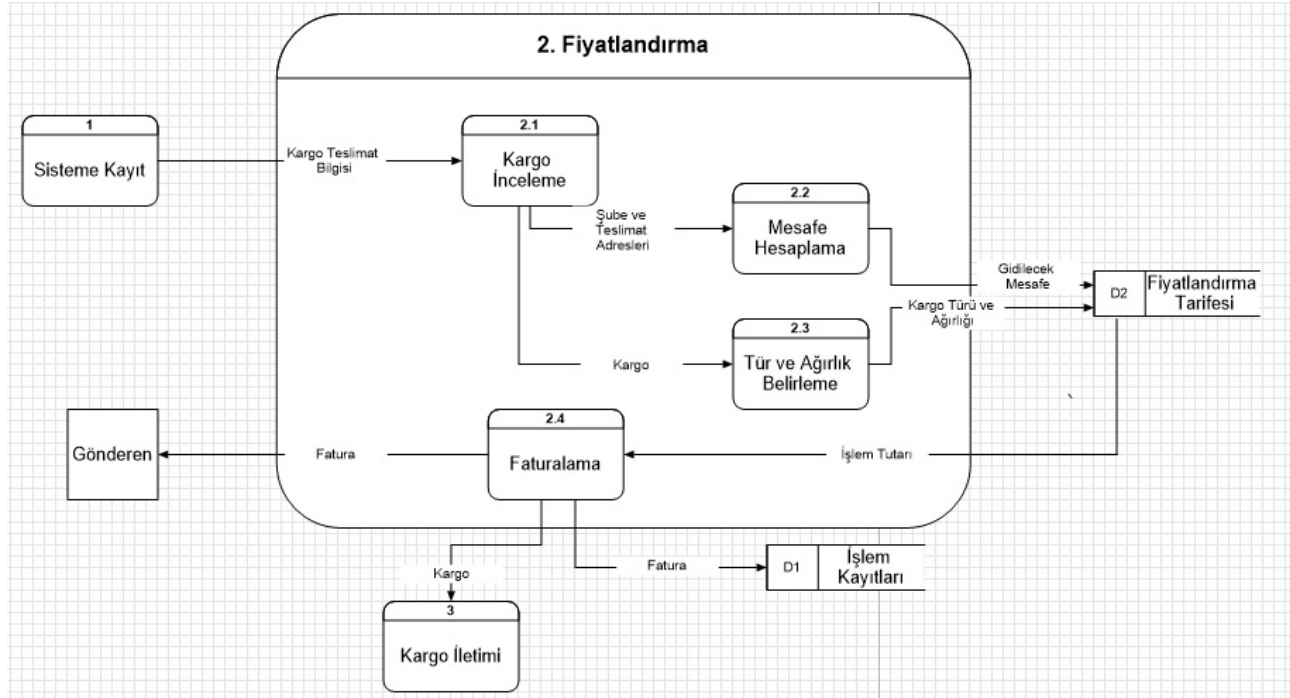
2.2.5. findPrice

Sisteme eklenen kargoların ücret bilgisini hesaplıyor.

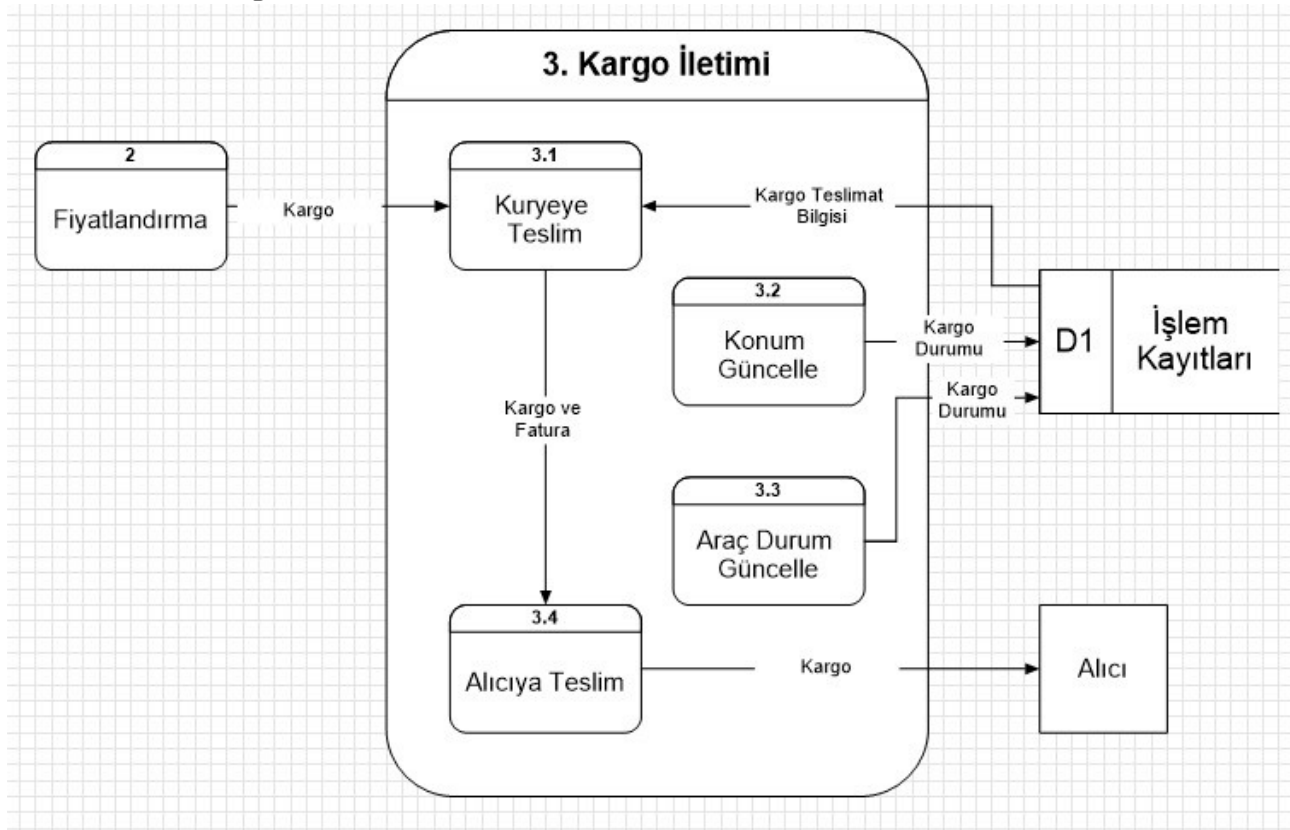
2.2.6. Sisteme Kayıt



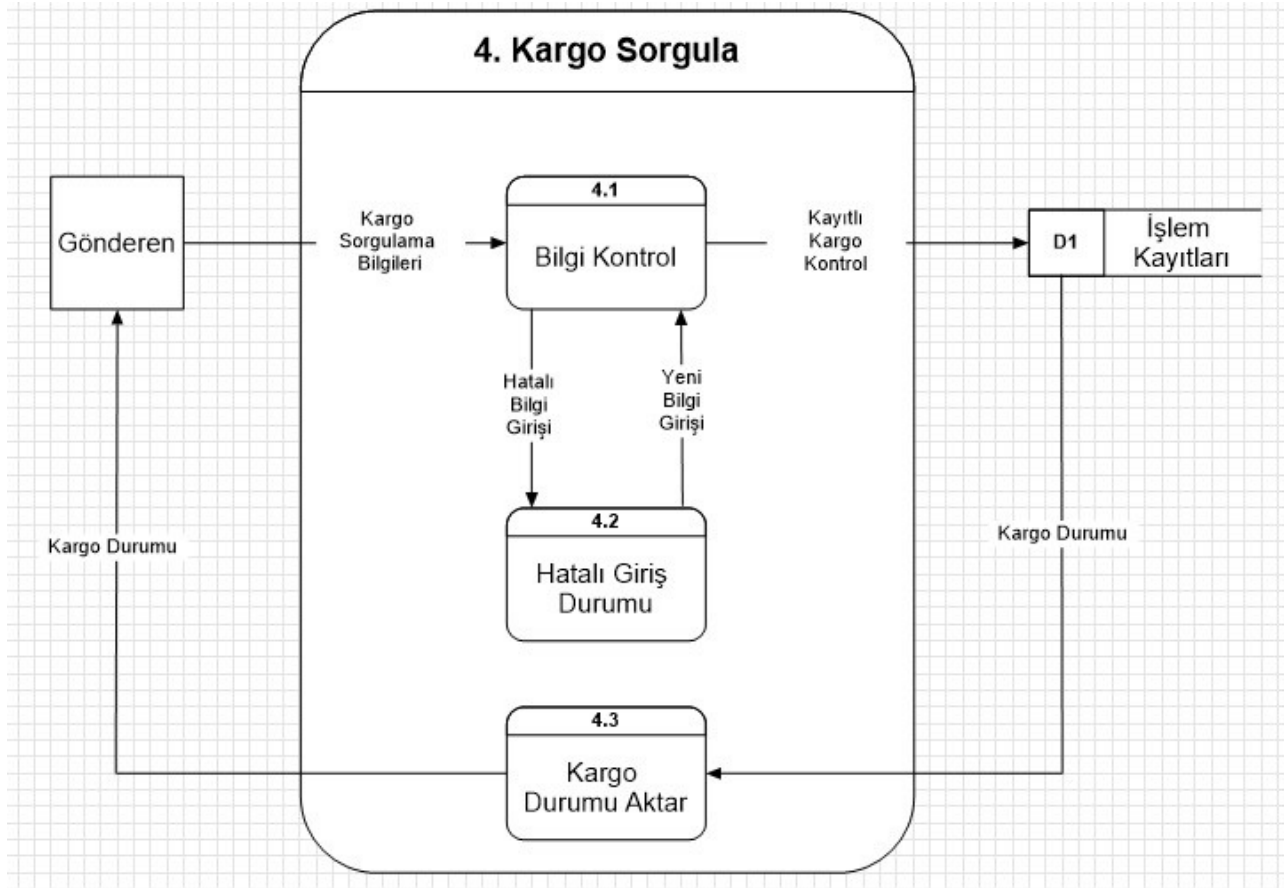
2.2.7. Fiyatlandırma



2.2.8. Kargo İletimi



2.2.9. Kargo Sorgula



UYGULAMALAR VE DENEYSEL SONUÇLAR

Yazılan kodlarımız bu aşamada test edilmiştir.

1. Class Branch

```
package project2;

import java.util.ArrayList;

public class Branch {
    private String branchName;
    int numberOfCargos = 0;
    private ArrayList<Cargo> cargos;

    public Branch() {
        super();
        this.branchName = branchName;
        cargos = new ArrayList<Cargo>();
    }
}
```

```

public void addCargo(Cargo c) {

    if(numberOfCargos<100) {
        cargos.add(c);
        numberOfCargos++;

    }
    else return;


}

public void cargoList() {

    for (int i=0; i<cargos.size(); i++) {
        System.out.println(cargos.get(i).toString());

    }
}

boolean deleteCargo(int cargoNo){
    for (int i = 0;i<cargos.size();i++) {
        if (cargos.get(i).getCargoNo()==cargoNo) {
            cargos.remove(i);
            return true;

        }
    }

    return false;
}

public String getBranchName() {
    return branchName;
}

    public void setBranchName(String branchName) {
        this.branchName = branchName;
    }

}

```


2. Class Cargo

```
package project2;

public abstract class Cargo {
    private int cargoNo;
    private String fromWhere;
    private String senderName;
    private String senderPhone;
    private String toWhere;
    private String destinationName;
    private String destinationPhone;
    private double weight;
    private int range;
    public Cargo(int cargoNo, String fromWhere, String senderName,
String senderPhone, String toWhere,
                String destinationName, String destinationPhone,
double weight,int range) {

        this.cargoNo = cargoNo;
        this.fromWhere = fromWhere;
        this.senderName = senderName;
        this.senderPhone = senderPhone;
        this.toWhere = toWhere;
        this.destinationName = destinationName;
        this.destinationPhone = destinationPhone;
        this.weight = weight;
        this.range=range;
    }
    public int getRange() {
        return range;
    }
    public void setRange(int range) {
        this.range = range;
    }
    public String toString() {
        return "Cargo [cargoNo=" + cargoNo + ", fromWhere=" +
fromWhere + ", senderName=" + senderName
            + ", senderPhone=" + senderPhone + ", toWhere="
+ toWhere + ", destinationName=" + destinationName
            + ", destinationPhone=" + destinationPhone + ",
weight=" + weight + "];";
    }
    public int getCargoNo() {
        return cargoNo;
    }
    public void setCargoNo(int cargoNo) {
        this.cargoNo = cargoNo;
    }
    public String getFromWhere() {
```

```

        return fromWhere;
    }
    public void setFromWhere(String fromWhere) {
        this.fromWhere = fromWhere;
    }
    public String getSenderName() {
        return senderName;
    }
    public void setSenderName(String senderName) {
        this.senderName = senderName;
    }
    public String getSenderPhone() {
        return senderPhone;
    }
    public void setSenderPhone(String senderPhone) {
        this.senderPhone = senderPhone;
    }
    public String getToWhere() {
        return toWhere;
    }
    public void setToWhere(String toWhere) {
        this.toWhere = toWhere;
    }
    public String getDestinationName() {
        return destinationName;
    }
    public void setDestinationName(String destinationName) {
        this.destinationName = destinationName;
    }
    public String getDestinationPhone() {
        return destinationPhone;
    }
    public void setDestinationPhone(String destinationPhone) {
        this.destinationPhone = destinationPhone;
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(double weight) {
        this.weight = weight;
    }

    public double findTime() {
        return range/90;
    }

    public abstract double findPrice();

```

```

}

```

3. Class sCargo

```
package project2;

public class sCargo extends Cargo {

    public sCargo(int cargoNo, String fromWhere, String
senderName, String senderPhone, String toWhere,
                String destinationName, String destinationPhone,
double weight,int range) {
        super(cargoNo, fromWhere, senderName, senderPhone,
toWhere, destinationName, destinationPhone, weight, range);
        // TODO Auto-generated constructor stub
    }

    @Override
    public double findPrice() {
        // TODO Auto-generated method stub
        return getWeight()*0.40*(getRange()*0.01);
    }

}
```

4. Class mCargo

```
package project2;

public class mCargo extends Cargo {

    public mCargo(int cargoNo, String fromWhere, String
senderName, String senderPhone, String toWhere,
                String destinationName, String destinationPhone,
double weight,int range) {
        super(cargoNo, fromWhere, senderName, senderPhone,
toWhere, destinationName, destinationPhone, weight,range);
        // TODO Auto-generated constructor stub
    }

    @Override
    public double findPrice() {

        return getWeight()*0.60*(getRange()*0.01);
    }

}
```

5. Class lCargo

```
package project2;

public class lCargo extends Cargo {

    public lCargo(int cargoNo, String fromWhere, String
senderName, String senderPhone, String toWhere,
                String destinationName, String destinationPhone,
double weight,int range) {
        super(cargoNo, fromWhere, senderName, senderPhone,
toWhere, destinationName, destinationPhone, weight,range);
        // TODO Auto-generated constructor stub
    }

    @Override
    public double findPrice() {

        return getWeight()*0.90*(getRange()*0.01);
    }

}
```

6. Ekran Görüntüsü

```
run:
Cargo [cargoNo=1, fromWhere=batman, senderName=ali, senderPhon
e=123, toWhere=ankara, destinationName=veli,
destinationPhone=234, weight=7.0]
Cargo [cargoNo=1, fromWhere=batman, senderName=ali, senderPhon
e=123, toWhere=ankara, destinationName=veli,
destinationPhone=234, weight=7.0]
Cargo [cargoNo=1, fromWhere=batman, senderName=ali, senderPhon
e=123, toWhere=ankara, destinationName=veli,
destinationPhone=234, weight=7.0]
10.0
25.2000000000000003
10.0
25.2000000000000003
10.0
56.699999999999996
BUILD SUCCESSFUL (total time: 0 seconds)
```

REFERANSLAR

1. Karayolu Taşıma Yönetmeliği:

<http://www.tesk.org.tr/tr/mevzuat/im12.php>

2. Donanım gereksinimleri:

<https://www.java.com/tr/download/help/sysreq.xml>