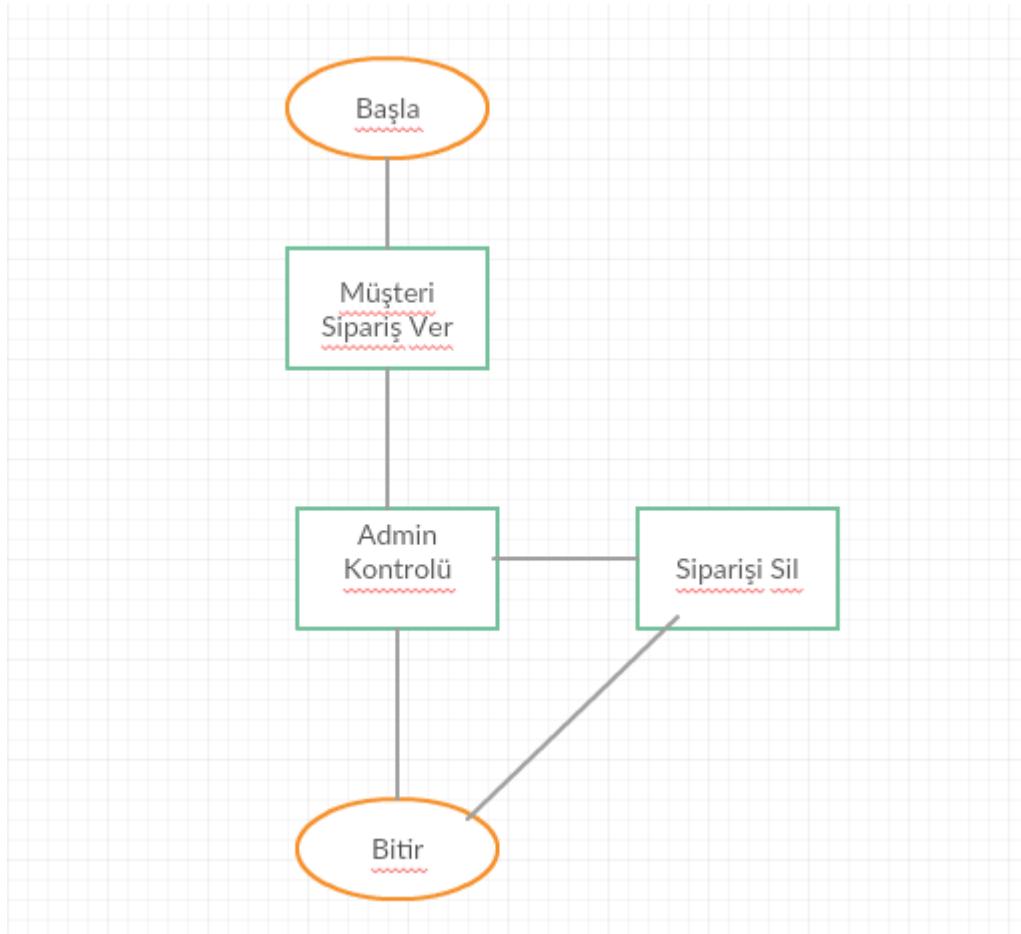
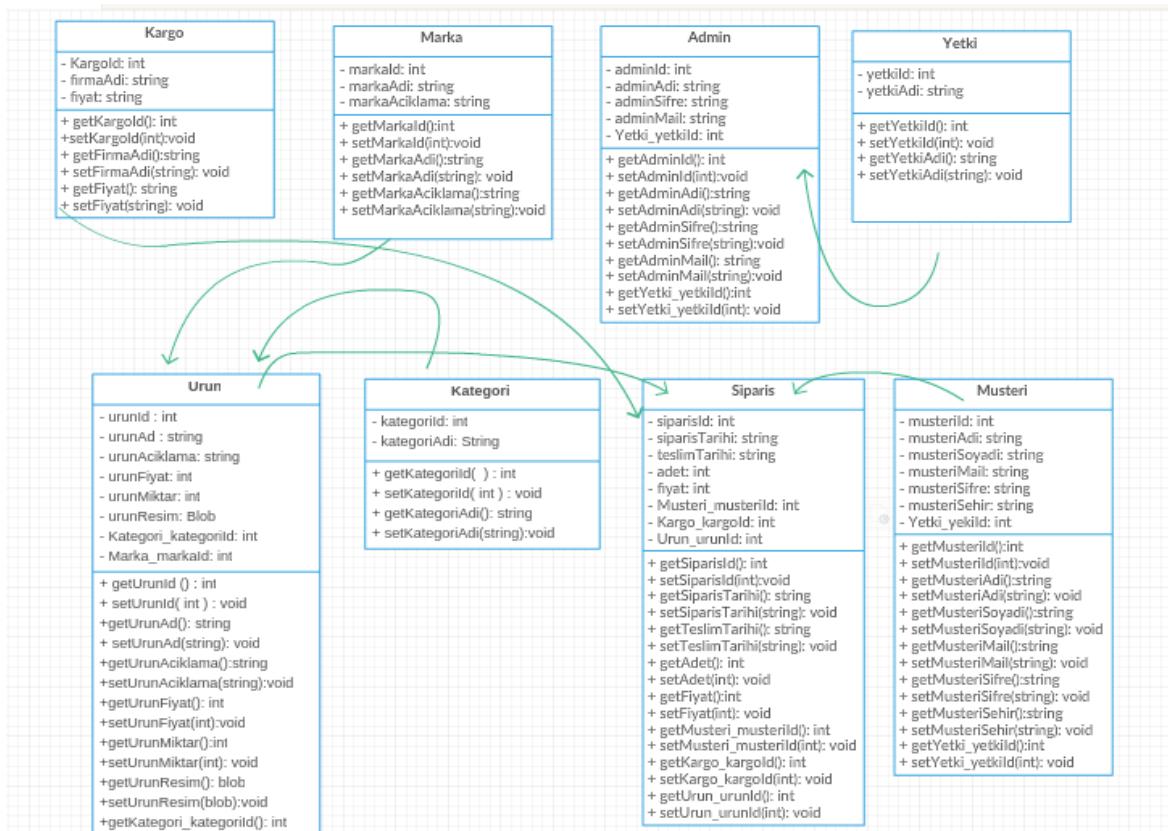


Murat DEMİR

ACTİVİTY DİYAGRAM



CLASS DİYAGRAMI



Admin

Admin	
- adminId: int	
- adminAdi: string	
- adminSifre: string	
- adminMail: string	
- Yetki_yetkild: int	
+ getAdminId(): int	
+ setAdminId(int):void	
+ getAdminAdi():string	
+ setAdminAdi(string): void	
+ getAdminSifre():string	
+ setAdminSifre(string):void	
+ getAdminMail(): string	
+ setAdminMail(string):void	
+ getYetki_yetkild():int	
+ setYetki_yetkild(int): void	

Kategori

Kategori	
- kategoriId: int	
- kategoriAdi: String	
+ getKategoriId() : int	
+ setKategoriId(int) : void	
+ getKategoriAdi(): string	
+ setKategoriAdi(string):void	

Kargo

Kargo	
- Kargold: int	
- firmaAdi: string	
- fiyat: string	
+ getKargold(): int	
+setKargold(int):void	
+ getFirmaAdi():string	
+ setFirmaAdi(string): void	
+ getFiyat(): string	
+ setFiyat(string): void	

Marka

Marka
- markaId: int - markaAdi: string - markaAciklama: string
+ getMarkaId():int + setMarkaId(int):void + getMarkaAdi():string + setMarkaAdi(string): void + getMarkaAciklama():string + setMarkaAciklama(string):void

Musteri

Musteri
- musterId: int - musterAdi: string - musterSoyadi: string - musterMail: string - musterSifre: string - musterSehir: string - Yetki_yekild: int
+ getMusterId():int + setMusterId(int):void + getMusterAdi():string + setMusterAdi(string): void + getMusterSoyadi():string + setMusterSoyadi(string): void + getMusterMail():string + setMusterMail(string): void + getMusterSifre():string + setMusterSifre(string): void + getMusterSehir():string + setMusterSehir(string): void + getYetki_yekild():int + setYetki_yekild(int): void

Siparis

Siparis	
- siparisId: int	
- siparisTarihi: string	
- teslimTarihi: string	
- adet: int	
- fiyat: int	
- Musteri_musterild: int	
- Kargo_kargold: int	
- Urun_urunId: int	
+ getSiparisId(): int	
+ setSiparisId(int):void	
+ getSiparisTarihi(): string	
+ setSiparisTarihi(string): void	
+ getTeslimTarihi(): string	
+ setTeslimTarihi(string): void	
+ getAdet(): int	
+ setAdet(int): void	
+ getFiyat():int	
+ setFiyat(int): void	
+ getMusteri_musterild(): int	
+ setMusteri_musterild(int): void	
+ getKargo_kargold(): int	
+ setKargo_kargold(int): void	
+ getUrun_urunId(): int	
+ setUrun_urunId(int): void	

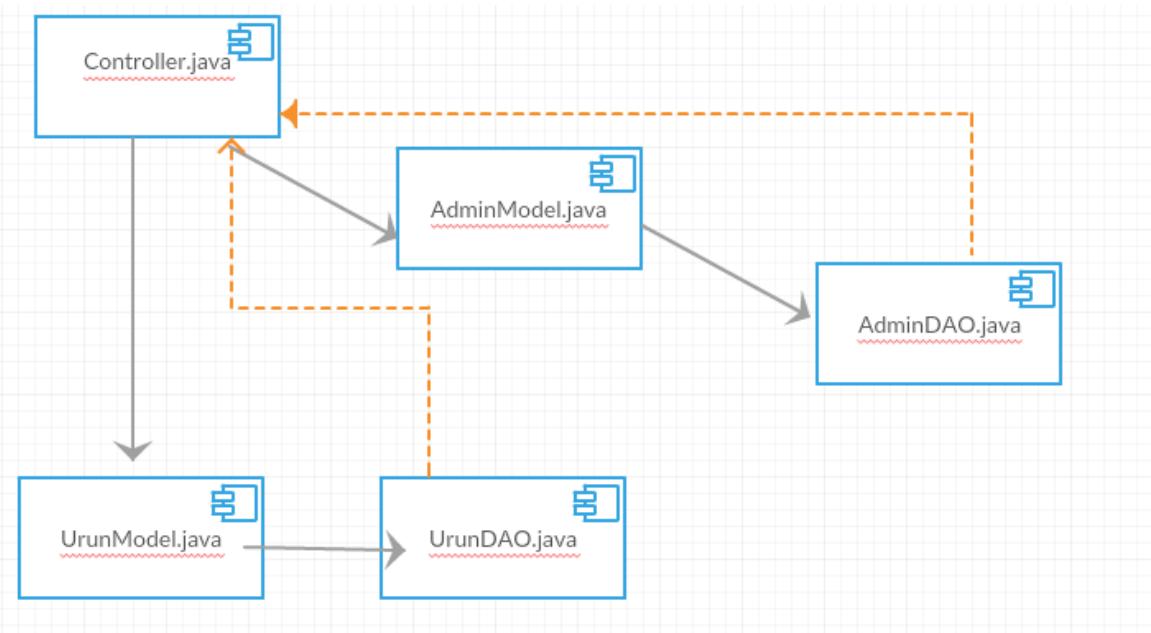
Yetki

Yetki	
- yetkild: int	
- yetkiAdi: string	
+ getYetkild(): int	
+ setYetkild(int): void	
+ getYetkiAdi(): string	
+ setYetkiAdi(string): void	

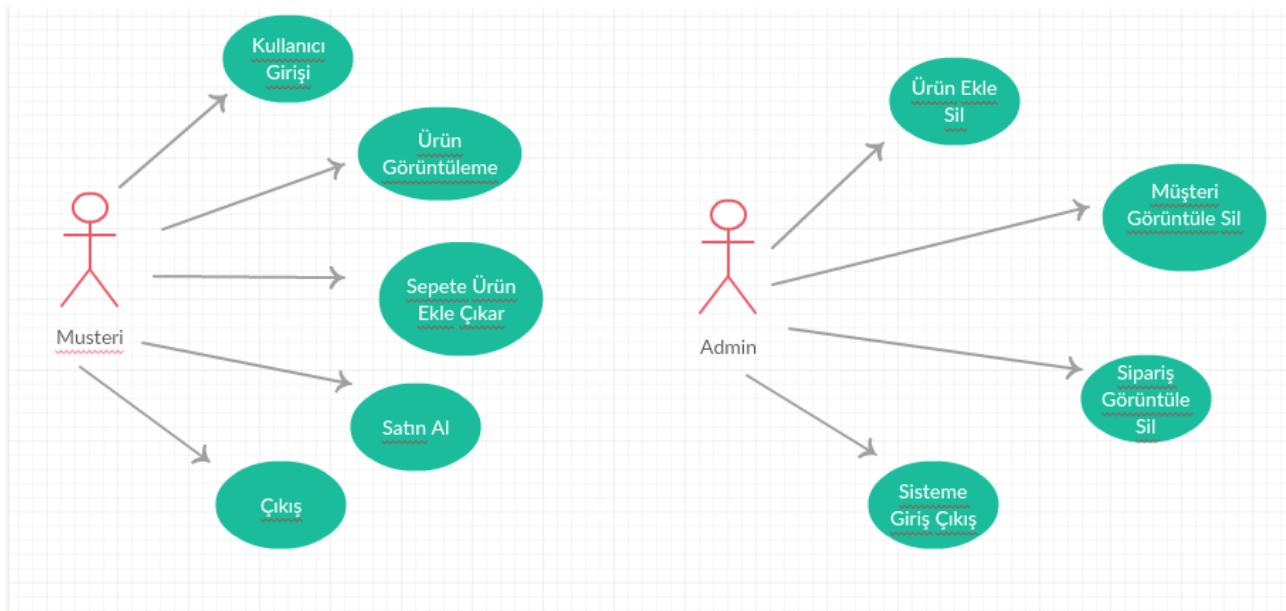
Urun

Urun	
- urunId : int	
- urunAd : string	
- urunAciklama: string	
- urunFiyat: int	
- urunMiktar: int	
- urunResim: Blob	
- Kategori_kategorild: int	
- Marka_markald: int	
+ getUrundai () : int	
+ setUrundai(int) : void	
+getUrundai(): string	
+ setUrundai(string): void	
+getUrundaiAciklama():string	
+setUrundaiAciklama(string):void	
+getUrundaiFiyat(): int	
+setUrundaiFiyat(int):void	
+getUrundaiMiktar():int	
+setUrundaiMiktar(int): void	
+getUrundaiResim(): blob	
+setUrundaiResim(blob):void	
+getKategori_kategorild(): int	
+setKategori_kategorild(int):void	
+getMarka_markald():int	
+setMarka_markald(int):void	

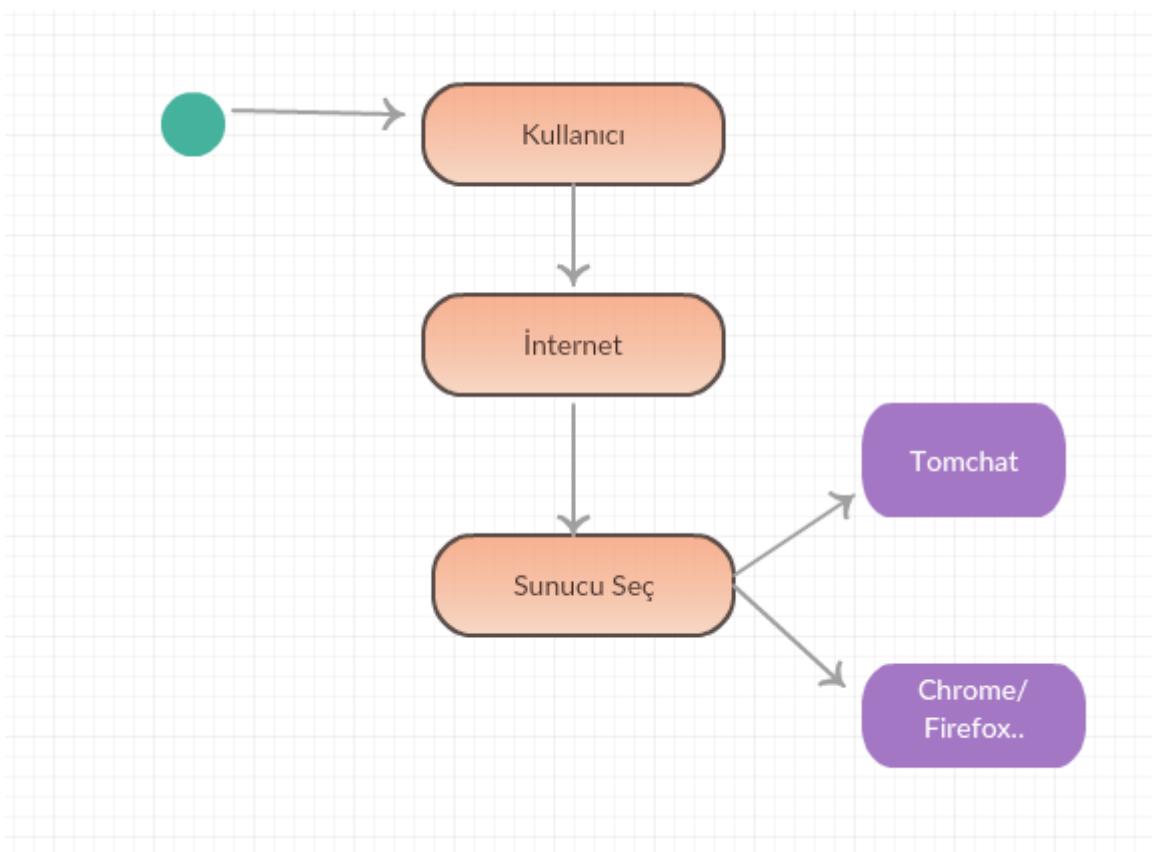
COMPONENT DİYAGRAM



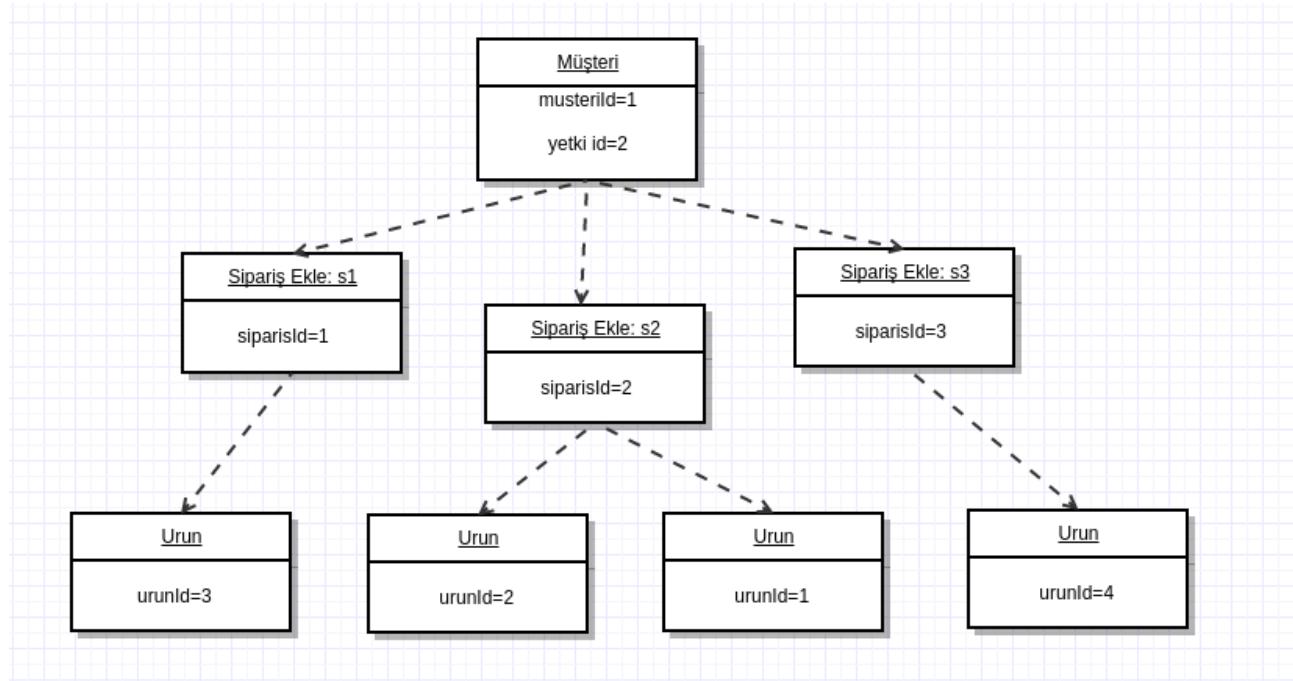
USE CASE DİYAGRAM



DEPLOYMENT DİYAGRAM

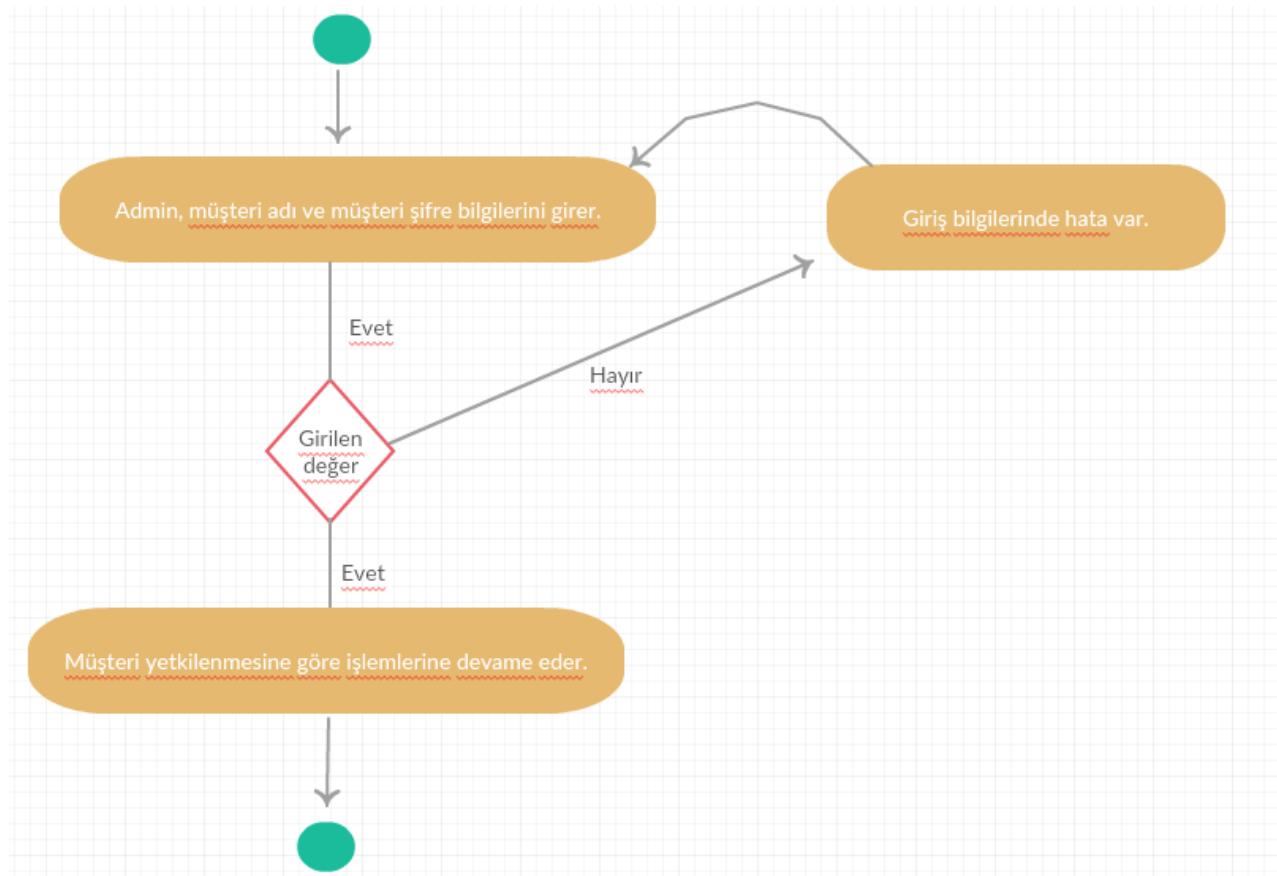


OBJECT DİYAGRAM



STATECHART DİYAGRAM

Müşteri Ekle/Güncelle



Müşteri Sil Görüntüle



Admin müşteri bölümüne girer.



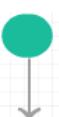
Müşteirler listelenir.



Silmek istenilen müşteri silinir.



Satın Alma



Müşteri seçtiği ürünü sepete ekler.



Sepet butonuna basıp sepetin içeriğini görür.



Sepette ürün var mı?

Hayır

Satin alma işlemi gerçekleşmez.

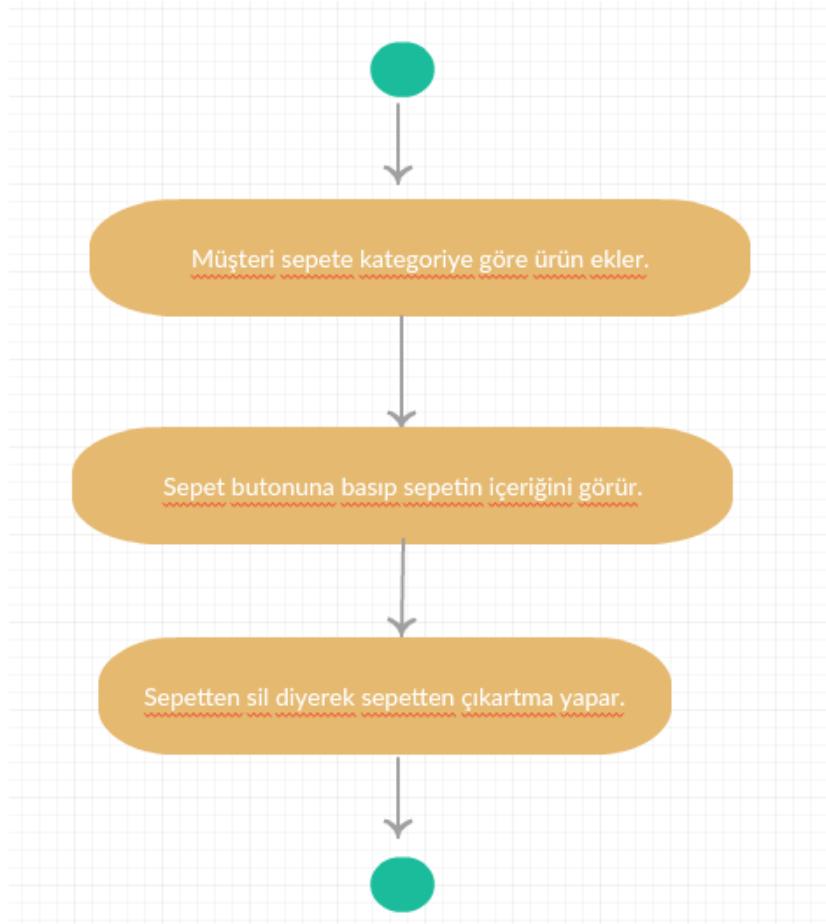


Evet

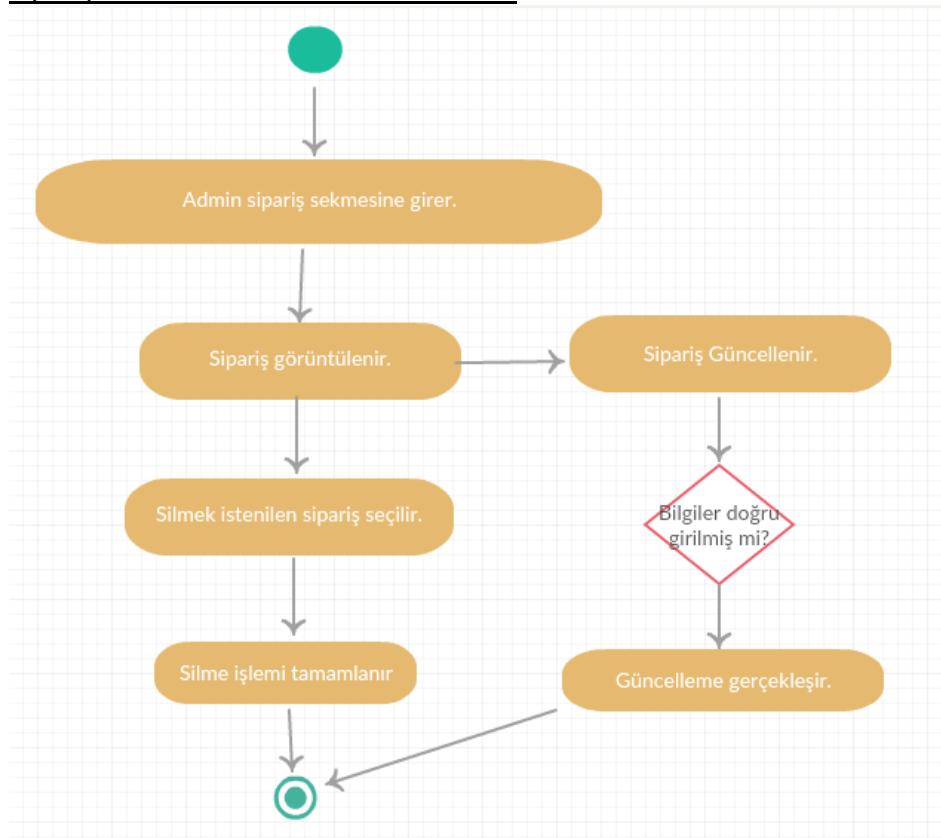
Satin alma işlemi gerçeklesir ve sepet boşalır.



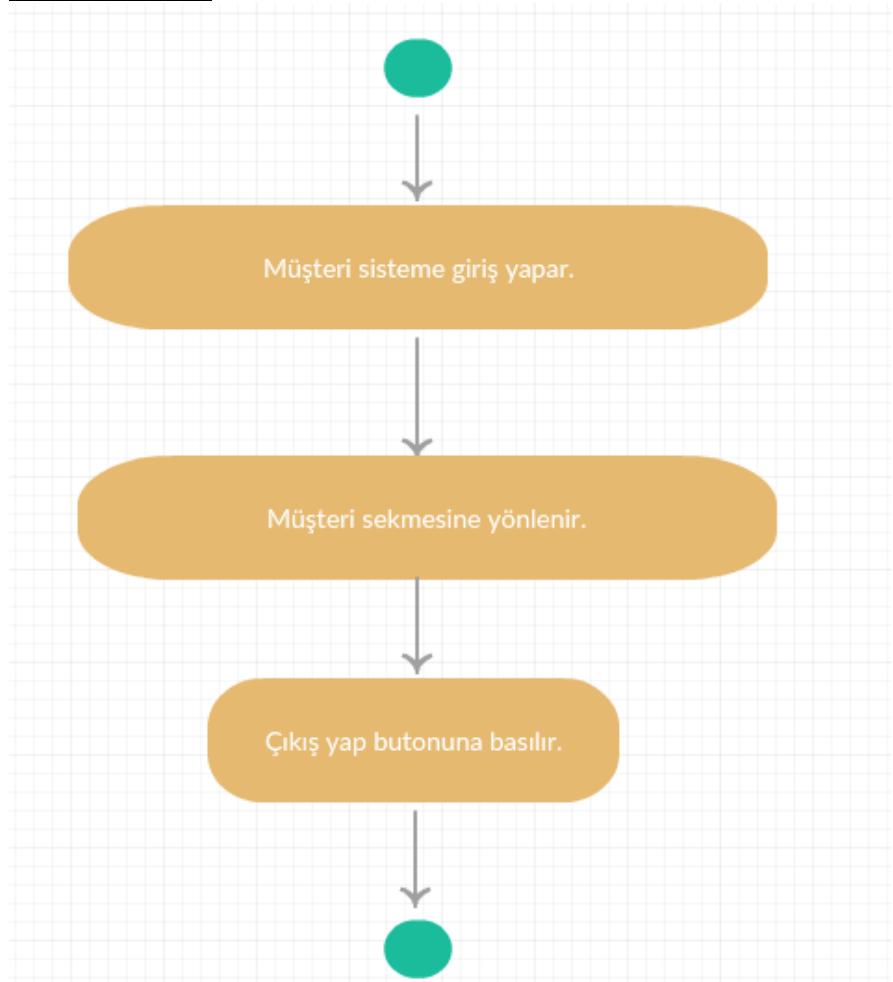
Sepete Ürün Ekleme Çıkarma



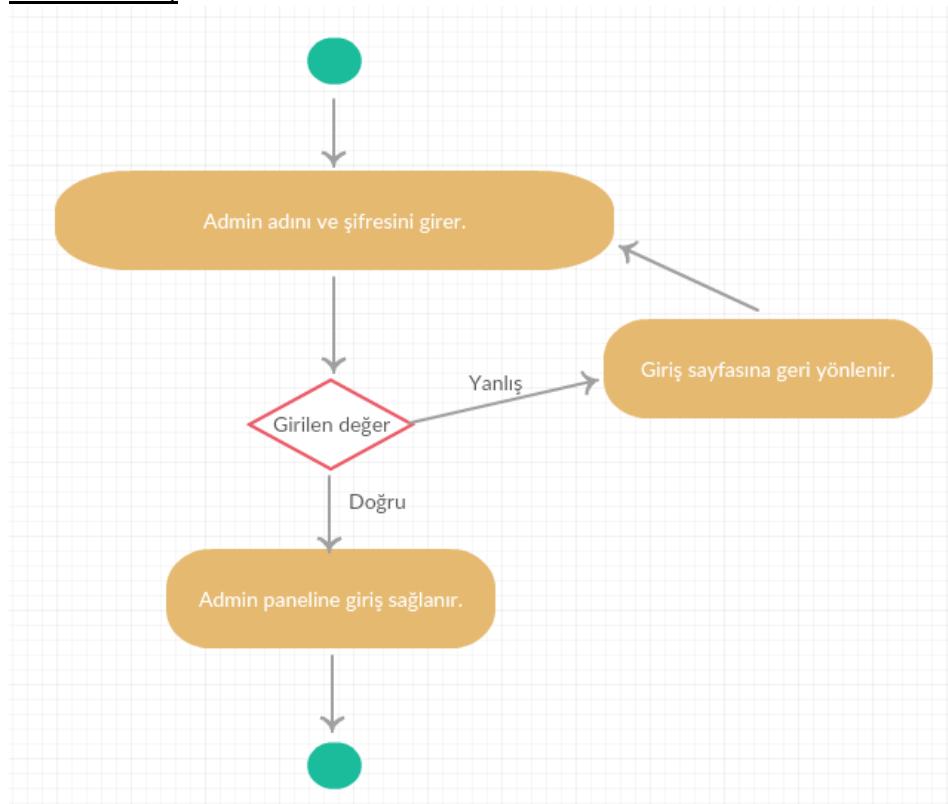
Sipariş Görüntüleme Güncelleme Silme



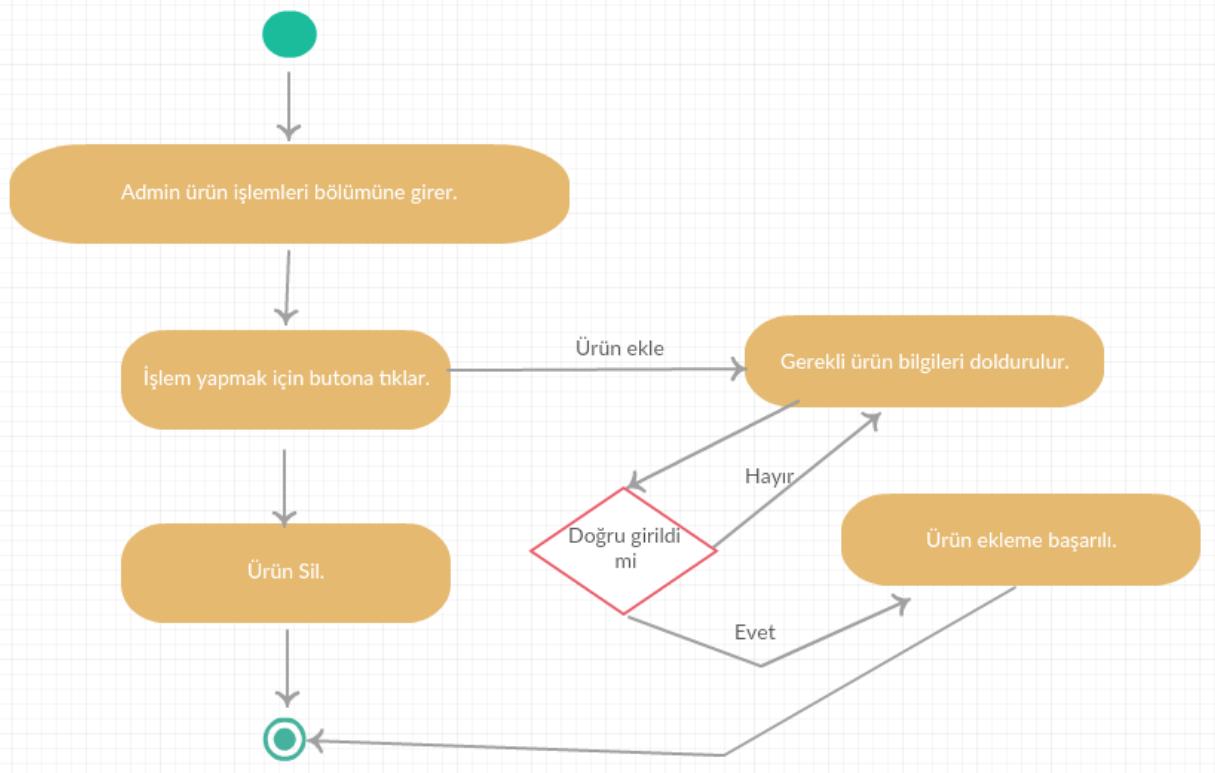
Sisteme Giriş



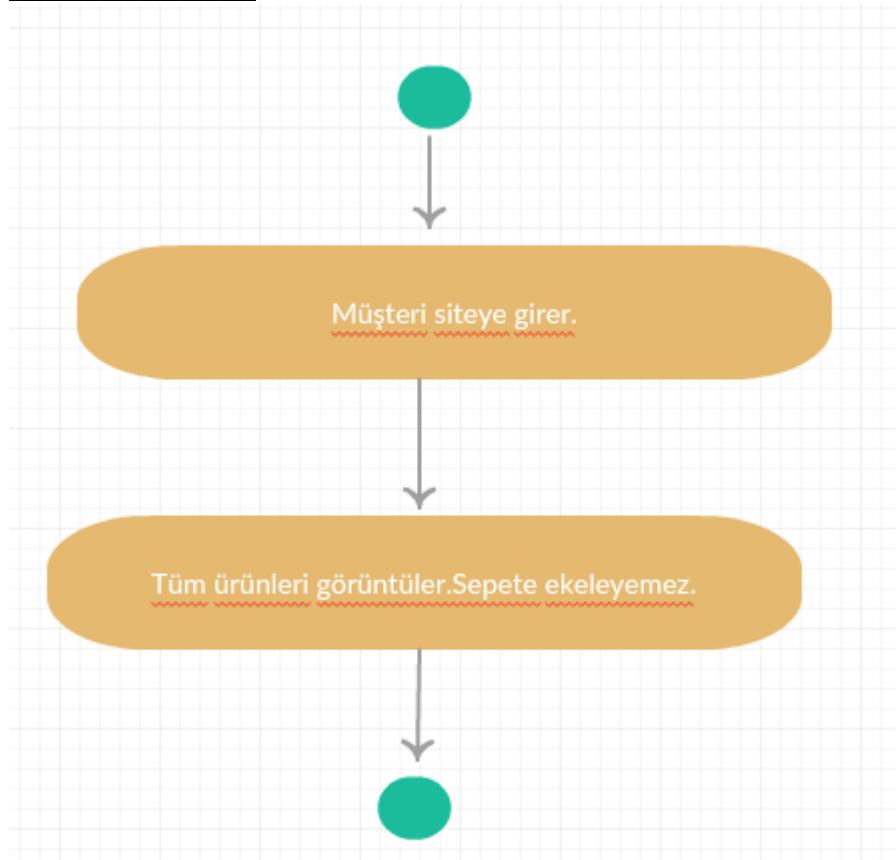
Sisteme Çıkış



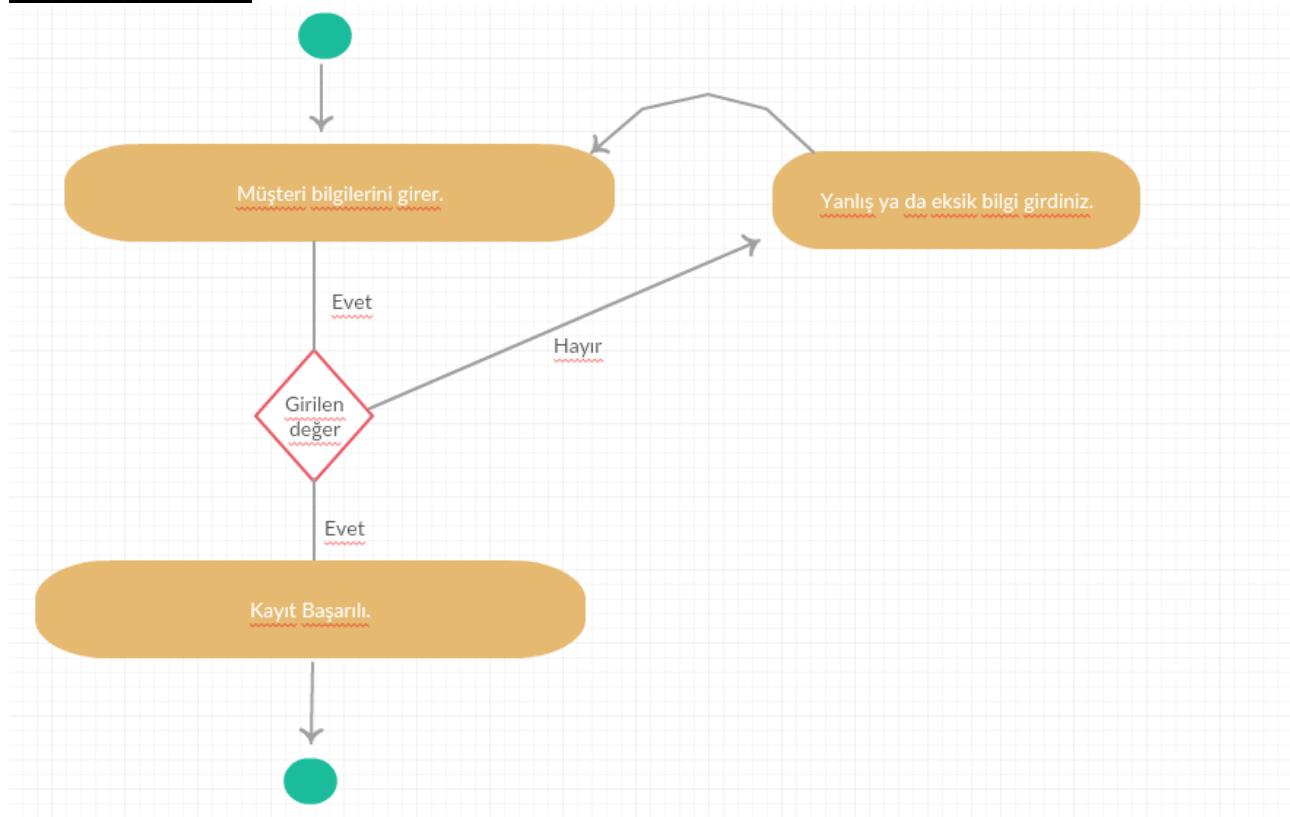
Ürün Ekleme Silme



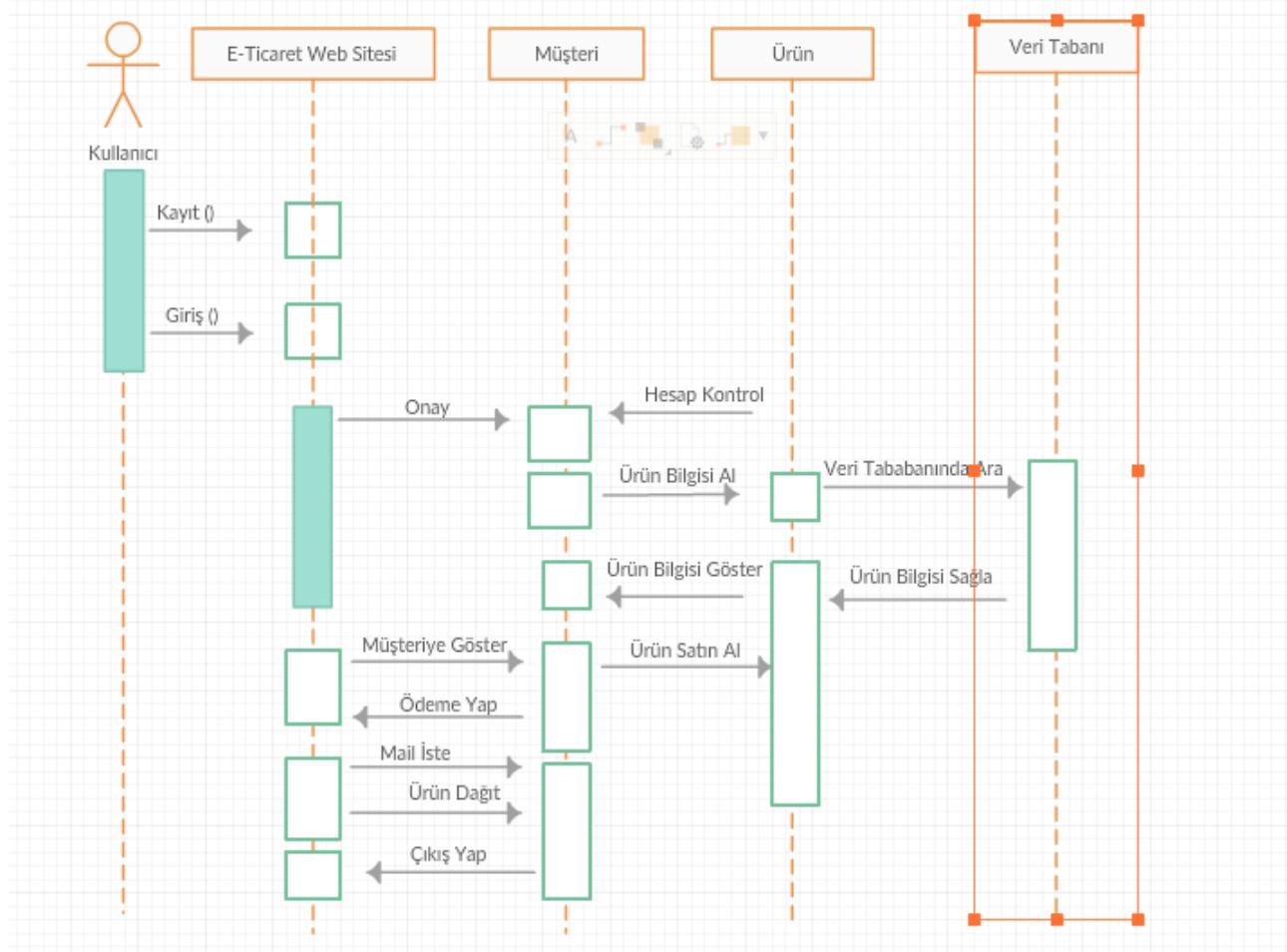
Ürün Görüntüleme



Yeni Müşteri Kaydı

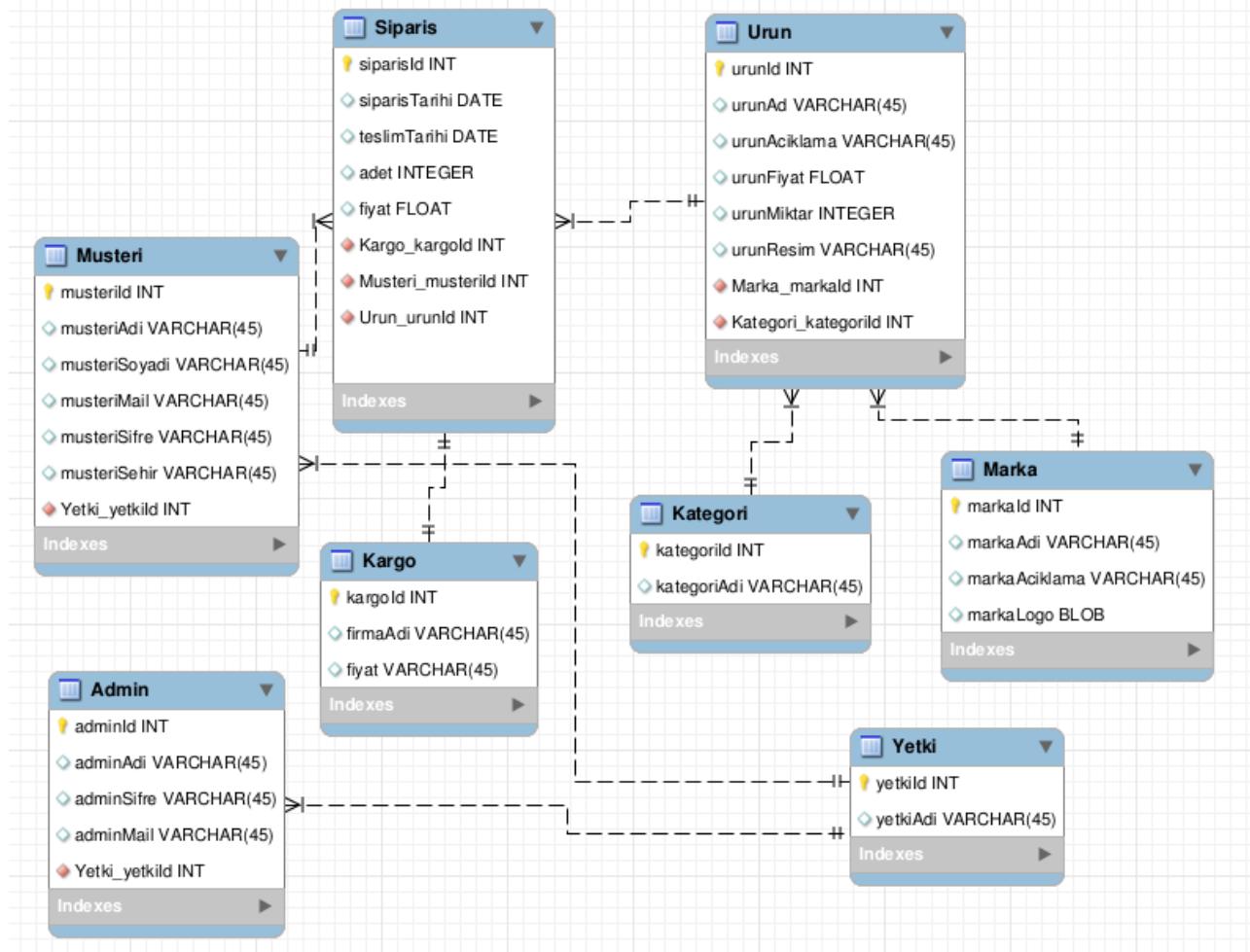


SEQUENCES DİYAGRAM



ER DİYAGRAMI

Tablolar arası 1-n ilişki olduğu için ara tablo yoktur. İlişkiler resimdeki gibidir.



E-TİCARET SİTESİNİN EKRAN GÖRÜNTÜSÜ ŞU ŞEKİLDEDİR:

Admin Girişi



Admin Giriş Yap

Mail :

murat.demir@bil.omu.edu.tr

Şifre :

...

Giriş Yap

Admin Kategori İşlemleri



Ana Sayfa Ayakkabı Erkek Bayan Aksesuar Teknoloji Çanta Sepet Oturumu Kapat Yönetim Paneli Ürün İşlemleri Kategori İşlemleri

Kategori Ekle

Kategori Adı :

Kategori Ekle

Admin Ürün İşlemleri



Ana Sayfa Ayakkabı Erkek Bayan Aksesuar Teknoloji Çanta Sepet Oturumu Kapat Yönetim Paneli Ürün İşlemleri Kategori İşlemleri

Ürün Ekle

Ürün Adı :

Ürün Açıklama :

Ürün Fiyat :

Ürün Miktar :

Ürün Kategori :

Ayakkabı

Resim Yükle :

Dosya seçilmedi

Ekle

Ürün Sil

Ürün	Ürün Adı	Fiyat	
	Kolye	100	Ürünü Sil
	Çanta	150	Ürünü Sil
	Gözlük	500	Ürünü Sil
	Kombin	250	Ürünü Sil

Admin Sipariş İşlemleri



Ana Sayfa Ayakkabı Erkek Bayan Aksesuar Teknoloji Çanta Sepet Oturumu Kapat Ürün İşlemleri

Adı- Soyadı	Ürün Adı	Adet	Fiyat
Murat Demir	kolye	1	₺ 100
		Ürünü Sil	
Murat Demir	Saat	1	₺ 1000
		Ürünü Sil	
Murat Demir	Saat	1	₺ 1000
		Ürünü Sil	
localhost:8080/e-ticaret/siparis.jsp	Kolye	1	₺ 100

Müşteri Giriş – Kayıt Ekranı



Ana Sayfa Elektronik Değerli Eşya Aksesuar Giyim Bahçe Kirtasiye Giriş & Kayıt

Kayıt Ol

Adı :

murat

Soyadı :

demir

Mail :

murat.demir@bil.omu.edu.tr

Şifre :

Şehir :

samsun

Kayıt Ol

Giriş Yap

Mail :

murat.demir@bil.omu.edu.tr

Şifre :

Giriş Yap

Ana Sayfa



Ana Sayfa Ayakkabı Erkek Bayan Aksesuar Teknoloji Çanta Giriş & Kayıt



Kolye



Çanta



Gözlük



Kombin



Kombin



Kemer

Giriş Yapıldıktan Sonra

 Ana Sayfa Ayakkabı Erkek Bayan Aksesuar Teknoloji Çanta Sepet Oturumu Kapat



Kolye



150 TL
Çanta



Gözlük



Kombin



Kombin



localhost:8080/e-ticaret/urun.jsp?urunId=14

Kategoriye Göre Getir (Erkek Tıklandı)

 Ana Sayfa Ayakkabı Erkek Bayan Aksesuar Teknoloji Çanta Sepet Oturumu Kapat



Kombin



Kombin



Kombin

Sepet

 Ana Sayfa Ayakkabı Erkek Bayan Aksesuar Teknoloji Çanta Sepet Oturumu Kapat

Sepetim

Ürün	Ürün Adı	Fiyat	
	Gözlük	500	<button>Sepetten Çıkar</button>
	Kombin	500	<button>Sepetten Çıkar</button>

Satin Al

Ürüne Git



Ara Sayfa Ayakkabı Erkek Bayan Aksesuar Teknoloji Çanta Sepet Oturumu Kapat



Çanta

150

Deri

Sepete At

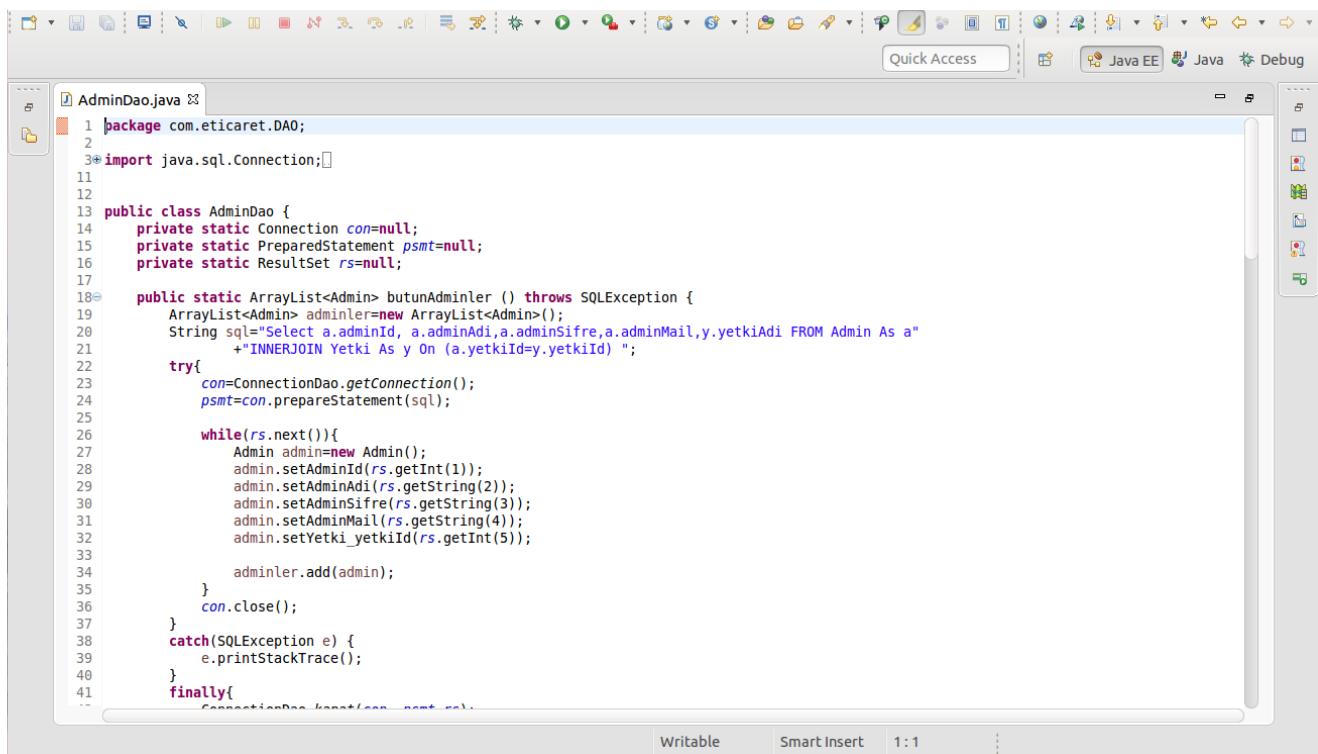
MODEL – VIEW – CONTROLLER YAPISINA GÖRE PROJE KODLARIM :

Dao Sınıfları:

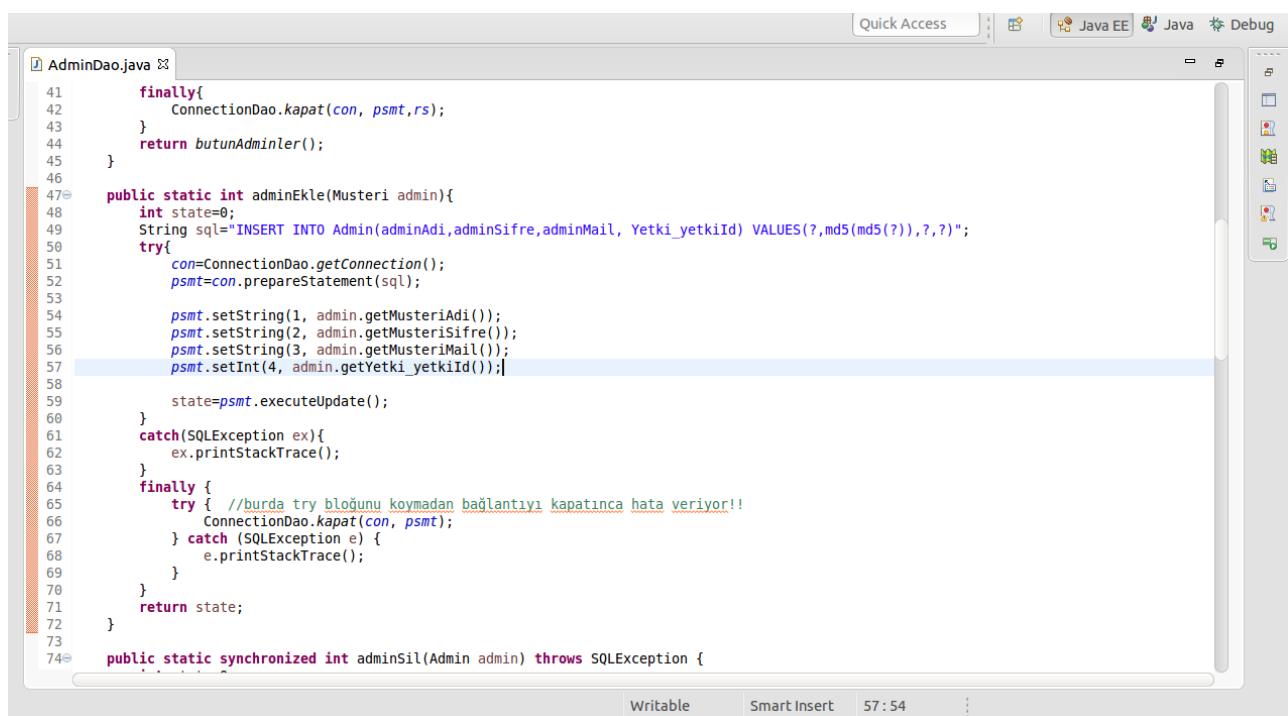
▼ # com.eticaret.DAO

- AdminDao.java
- ConnectionDao.java
- KargoDao.java
- KategoriDao.java
- MarkaDao.java
- MusteriDao.java
- SiparisDao.java
- UrunDao.java
- YetkiDao.java

AdminDao.java



```
1 package com.eticaret.DAO;
2
3 import java.sql.Connection;
4
5 public class AdminDao {
6     private static Connection con=null;
7     private static PreparedStatement psmt=null;
8     private static ResultSet rs=null;
9
10    public static ArrayList<Admin> butunAdminler () throws SQLException {
11        ArrayList<Admin> adminler=new ArrayList<Admin>();
12        String sql="Select a.adminId, a.adminAdi,a.adminSifre,a.adminMail,y.yetkiAdi FROM Admin As a"
13                +"INNERJOIN Yetki As y On (a.yetkiId=y.yetkiId) ";
14        try{
15            con=ConnectionDao.getConnection();
16            psmt=con.prepareStatement(sql);
17            while(rs.next()){
18                Admin admin=new Admin();
19                admin.setAdminId(rs.getInt(1));
20                admin.setAdminAdi(rs.getString(2));
21                admin.setAdminSifre(rs.getString(3));
22                admin.setAdminMail(rs.getString(4));
23                admin.setYetki_yetkiId(rs.getInt(5));
24
25                adminler.add(admin);
26            }
27            con.close();
28        } catch(SQLException e) {
29            e.printStackTrace();
30        }
31        finally{
32            ConnectionDao.kapat(con, psmt, rs);
33        }
34    }
35
36    public static int adminEkle(Musteri admin){
37        int state=0;
38        String sql="INSERT INTO Admin(adminAdi,adminSifre,adminMail, Yetki_yetkiId) VALUES(?,md5(?),?,?)";
39        try{
40            con=ConnectionDao.getConnection();
41            psmt=con.prepareStatement(sql);
42
43            psmt.setString(1, admin.getMusteriAdi());
44            psmt.setString(2, admin.getMusteriSifre());
45            psmt.setString(3, admin.getMusteriMail());
46            psmt.setInt(4, admin.getYetki_yetkiId());
47
48            state=psmt.executeUpdate();
49        } catch(SQLException ex){
50            ex.printStackTrace();
51        }
52        finally {
53            try { //burda try bloğunu koymadan bağlantıyı kapatınca hata veriyor!!
54                ConnectionDao.kapat(con, psmt);
55            } catch (SQLException e) {
56                e.printStackTrace();
57            }
58        }
59        return state;
60    }
61
62    public static synchronized int adminSil(Admin admin) throws SQLException {
63
64
65
66
67
68
69
70
71
72
73
74}
```



```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
```

The screenshot shows a Java IDE interface with a code editor window titled "AdminDao.java". The code implements two static methods: `adminSil` and `adminSifreGuncelle`. Both methods use JDBC to execute SQL statements. The `adminSil` method deletes an Admin record by ID. The `adminSifreGuncelle` method updates an Admin record's password (adminSifre) using MD5 hashing. Error handling includes catching `SQLException`s and printing stack traces. Finally, it closes the database connection.

```
public static synchronized int adminSil(Admin admin) throws SQLException {
    int state=0;
    String sql="Delete from Admin where adminId=?;";
    try{
        con= ConnectionDao.getConnection();
        psmt= con.prepareStatement(sql);
        psmt.setInt(1, admin.getAdminId());
        state=psmt.executeUpdate();
    }
    catch (SQLException ex){
        ex.printStackTrace();
    }
    finally{
        ConnectionDao.kapat(con, psmt);
    }
    return state;
}

public static synchronized int adminSifreGuncelle(Admin admin) throws SQLException {
    int state=0;
    String sql="Update from Admin set adminSifre=md5(?) where (adminId=?);";
    try{
        con=ConnectionDao.getConnection();
        psmt=con.prepareStatement(sql);
        psmt.setString(1, admin.getAdminSifre());
        psmt.setInt(2, admin.getAdminId());
        state=psmt.executeUpdate();
        con.close();
    }
}
```

The screenshot shows a Java IDE interface with a code editor window titled "AdminDao.java". The code continues from the previous snippet, showing the implementation of the `getAdminGiris` method. This method performs a SELECT query to find an Admin record matching the provided email and password. It uses JDBC to connect to the database, prepare a statement, and execute a query. If a result is found, it creates a new `Admin` object and sets its properties from the result set. Finally, it closes the connection and handles any potential `SQLException`.

```
try{
    con=ConnectionDao.getConnection();
    psmt=con.prepareStatement(sql);
    psmt.setString(1, Mail);
    psmt.setString(2, Sifre);
    rs=psmt.executeQuery();
    if(rs.next()){
        admin=new Admin();
        admin.setAdminId(rs.getInt(1));
        admin.setAdminAdi(rs.getString(2));
        admin.setAdminSifre(rs.getString(3));
        admin.setAdminMail(rs.getString(4));
    }
    con.close();
}
catch (SQLException ex) {
    ex.printStackTrace();
} finally {
    try {
```

The screenshot shows a Java code editor window with the file `AdminDao.java` open. The code implements a static method `getAdminGiris` that performs a database query to retrieve an `Admin` object based on provided email and password. It uses try-with-resources to handle the database connection and statement. Error handling includes printing the stack trace for any SQL exceptions.

```
116     public static Admin getAdminGiris(String Mail, String Sifre) {
117         Admin admin=null;
118         String sql="Select * from Admin where adminMail=? and adminSifre=?;";
119
120         try{
121             con=ConnectionDao.getConnection();
122             psmt=con.prepareStatement(sql);
123
124             psmt.setString(1, Mail);
125             psmt.setString(2, Sifre);
126             rs=psmt.executeQuery();
127
128             if(rs.next()){
129                 admin=new Admin();
130                 admin.setAdminId(rs.getInt(1));
131                 admin.setAdminAdi(rs.getString(2));
132                 admin.setAdminSifre(rs.getString(3));
133                 admin.setAdminMail(rs.getString(4));
134             }
135             con.close();
136         } catch (SQLException ex) {
137             ex.printStackTrace();
138         } finally {
139             try { }
140             catch (SQLException ex) {
141                 ConnectionDao.kapat(con, psmt);
142             } catch (SQLException e) {
143                 e.printStackTrace();
144             }
145         }
146         return admin;
147     }
148 }
```

ConnectionDao.java

The screenshot shows a Java code editor window with the file `ConnectionDao.java` open. The code defines a static method `getConnection` that establishes a database connection using the MySQL JDBC driver. It also contains two static methods, `kapat`, which close database resources (Connection, PreparedStatement, and ResultSet) in a try-with-resources manner.

```
1 package com.eticaret.DAO;
2
3 import java.sql.Connection;
4
5 public class ConnectionDao {
6
7     public static Connection getConnection(){
8         Connection connectStatus =null;
9
10    try{
11        Class.forName("com.mysql.jdbc.Driver");
12        connectStatus=DriverManager.getConnection("jdbc:mysql://localhost:3306/eticaret?useUnicode=true&characterEncoding=UTF-8&useUnicode=true&characterEncoding=UTF-8");
13    } catch (SQLException | ClassNotFoundException e) {
14        e.printStackTrace();
15    }
16    return connectStatus;
17 }
18
19 public static void kapat(Connection con,PreparedStatement psmt, ResultSet rs) throws SQLException{
20     if (con != null)
21         con.close();
22     if (psmt != null)
23         psmt.close();
24     if (rs != null)
25         rs.close();
26 }
27
28 public static void kapat(Connection con,PreparedStatement psmt) throws SQLException{
29     if (con != null)
30         con.close();
31     if (psmt != null)
32         psmt.close();
33 }
34 }
```

KargoDao.java

The screenshot shows a Java code editor window with the file 'KargoDao.java' open. The code defines a class 'KargoDao' with two static methods: 'kargoListele()' and 'kargoSil(Kargo kargo)'. Both methods handle database connections and statements using try-catch-finally blocks. The code uses JDBC to execute SQL queries and update statements.

```
1 package com.eticaret.DAO;
2
3 import java.sql.Connection;
4
5 public class KargoDao {
6
7     private static Connection con = null;
8     private static PreparedStatement psmt = null;
9     private static ResultSet rs = null;
10
11    public static ArrayList<Kargo> kargoListele() throws SQLException {
12        ArrayList<Kargo> kargolar = new ArrayList<Kargo>();
13        String sql = "SELECT * FROM Kargo order by KargoId asc";
14        try {
15            con = ConnectionDao.getConnection();
16            psmt = con.prepareStatement(sql);
17            rs = psmt.executeQuery();
18            while (rs.next()) {
19                kargolar.add(new Kargo(rs.getInt(1), rs.getString(2), rs.getString(3)));
20            }
21            con.close();
22        } catch (SQLException ex) {
23            ex.printStackTrace();
24        } finally {
25            ConnectionDao.kapat(con, psmt, rs);
26        }
27        return kargolar;
28    }
29
30    public static int kargoSil(Kargo kargo) throws SQLException {
31        int state = 0;
32        String sql = "DELETE FROM Kargo WHERE kargoId = ?;";
33        try {
34            con = ConnectionDao.getConnection();
35            psmt = con.prepareStatement(sql);
36            psmt.setInt(1, kargo.getKargoId());
37            state = psmt.executeUpdate();
38            con.close();
39        } catch (SQLException ex) {
40            ex.printStackTrace();
41        } finally {
42            ConnectionDao.kapat(con, psmt);
43        }
44        return state;
45    }
46}
```

The screenshot shows the same Java code editor window with the file 'KargoDao.java' open. It now includes two additional static methods: 'kargoEkle(Kargo kargo)' and 'kargoSil(Kargo kargo)'. These methods also use JDBC to interact with the database, specifically for inserting new records and deleting existing ones.

```
36    public static int kargoSil(Kargo kargo) throws SQLException {
37        int state = 0;
38        String sql = "DELETE FROM Kargo WHERE kargoId = ?;";
39        try {
40            con = ConnectionDao.getConnection();
41            psmt = con.prepareStatement(sql);
42            psmt.setInt(1, kargo.getKargoId());
43            state = psmt.executeUpdate();
44            con.close();
45        } catch (SQLException ex) {
46            ex.printStackTrace();
47        } finally {
48            ConnectionDao.kapat(con, psmt);
49        }
50        return state;
51    }
52
53    public static int kargoEkle(Kargo kargo) throws SQLException {
54        int state = 0;
55        String sql = "INSERT INTO Kargo(firmaAdi, fiyat) VALUES (?, ?);";
56        try {
57            con = ConnectionDao.getConnection();
58            psmt = con.prepareStatement(sql);
59            psmt.setString(1, kargo.getFirmaAdi());
60            psmt.setString(2, kargo.getFiyat());
61            state = psmt.executeUpdate();
62            con.close();
63        } catch (SQLException ex) {
64            ex.printStackTrace();
65        } finally {
66            ConnectionDao.kapat(con, psmt);
67        }
68        return state;
69    }
70}
```

```
57     ...
58     con = ConnectionDao.getConnection();
59     psmt = con.prepareStatement(sql);
60     psmt.setString(1, kargo.getFirmaAdi());
61     psmt.setString(2, kargo.getFiyat());
62     state = psmt.executeUpdate();
63     con.close();
64 }catch(SQLException ex){
65     ex.printStackTrace();
66 }finally {
67     ConnectionDao.kapat(con, psmt);
68 }
69 return state;
70 }
71 public static int kargoGuncelle(Kargo kargo) throws SQLException {
72     int state = 0;
73     String sql = "UPDATE Kargo Set firmaAdi = ?, fiyat = ? WHERE kargoId = ?";
74     try{
75         con = ConnectionDao.getConnection();
76         psmt = con.prepareStatement(sql);
77         psmt.setString(1, kargo.getFirmaAdi());
78         psmt.setString(2, kargo.getFiyat());
79         psmt.setInt(3, kargo.getKargoId());
80         state = psmt.executeUpdate();
81         con.close();
82     }catch(SQLException ex){
83         ex.printStackTrace();
84     }finally {
85         ConnectionDao.kapat(con, psmt);
86     }
87     return state;
88 }
89 }
```

KategoriDao.java

```
1 package com.eticaret.DAO;
2
3 import java.sql.Connection;
4
5 public class KategoriDao {
6
7     private static Connection con = null;
8     private static PreparedStatement psmt = null;
9     private static ResultSet rs = null;
10
11     public static ArrayList<Kategori> butunKategoriler () throws SQLException {
12         ArrayList<Kategori> butunkategoriler = new ArrayList<Kategori>();
13         String sql = "SELECT * FROM Kategori;";
14         try{
15             con = ConnectionDao.getConnection();
16             psmt=con.prepareStatement(sql);
17             rs = psmt.executeQuery();
18             while(rs.next()){
19                 Kategori kategori=new Kategori();
20                 kategori.setKategoriId(rs.getInt(1));
21                 kategori.setKategoriAdi(rs.getString(2));
22                 butunkategoriler.add(kategori);
23             }
24         }catch(SQLException ex){
25             ex.printStackTrace();
26         }finally {
27             con.close();
28             psmt.close();
29             rs.close();
30         }
31     }
32     return butunkategoriler;
33 }
```

```
42
43@ public static int kategoriEkle(Kategori kategori) throws SQLException{
44     int state = 0;
45     String sql = "INSERT INTO Kategori(kategoriAdi) VALUES (?)";
46     try{
47         con = ConnectionDao.getConnection();
48         psmt = con.prepareStatement(sql);
49         psmt.setString(1, kategori.getKategoriAdi());
50         state = psmt.executeUpdate();
51         con.close();
52     }catch(SQLException ex){
53         ex.printStackTrace();
54     }finally {
55         ConnectionDao.kapat(con, psmt);
56     }
57     return state;
58 }
59
60@ public static int kategoriSil(Kategori kategori) throws SQLException{
61     int state = 0;
62     String sql = "DELETE FROM Kategori WHERE (kategoriId = ?);";
63     try{
64         con = ConnectionDao.getConnection();
65         psmt = con.prepareStatement(sql);
66         psmt.setInt(1, kategori.getKategoriId());
67         state = psmt.executeUpdate();
68         con.close();
69     }catch(SQLException ex){
70         ex.printStackTrace();
71     }finally {
72         ConnectionDao.kapat(con, psmt);
73     }
74     return state;
75 }
76 }
```

Writable Smart Insert 1:1

```
52     String sql = "DELETE FROM Kategori WHERE (kategoriId = ?);";
53     try{
54         con = ConnectionDao.getConnection();
55         psmt = con.prepareStatement(sql);
56         psmt.setInt(1, kategori.getKategoriId());
57         state = psmt.executeUpdate();
58         con.close();
59     }catch(SQLException ex){
60         ex.printStackTrace();
61     }finally {
62         ConnectionDao.kapat(con, psmt);
63     }
64     return state;
65 }
66
67@ public static int kategoriGuncelle(Kategori kategori) throws SQLException{
68     int state = 0;
69     String sql = "UPDATE Kategori SET kategoriAdi = ? WHERE (kategoriId = ?);";
70     try{
71         con = ConnectionDao.getConnection();
72         psmt = con.prepareStatement(sql);
73         psmt.setString(1, kategori.getKategoriAdi());
74         psmt.setInt(2, kategori.getKategoriId());
75         state = psmt.executeUpdate();
76         con.close();
77     }catch(SQLException ex){
78         ex.printStackTrace();
79     }finally {
80         ConnectionDao.kapat(con, psmt);
81     }
82     return state;
83 }
84
85 }
```

Writable Smart Insert 1:1

MarkaDao.java

```
1 package com.eticaret.DAO;
2
3 import java.sql.Connection;
4 //import com.eticaret.model.Urun;
5
6 public class MarkaDao {
7     private static Connection con = null;
8     private static PreparedStatement psmt = null;
9     private static ResultSet rs = null;
10
11
12     public static ArrayList<Marka> markalarGetir() throws SQLException {
13         ArrayList<Marka> markalar = new ArrayList<Marka>();
14         String sql = "Select * from Marka order by markaId asc;";
15         try {
16             con = ConnectionDao.getConnection();
17             psmt = con.prepareStatement(sql);
18             rs = psmt.executeQuery();
19             while (rs.next()) {
20                 Marka marka = new Marka();
21                 marka.setMarkaId(rs.getInt(1));
22                 marka.setMarkaAdi(rs.getString(2));
23                 marka.setMarkaAciklama(rs.getString(3));
24                 markalar.add(marka);
25             }
26             con.close();
27         } catch (SQLException ex) {
28             ex.printStackTrace();
29         } finally {
30             try {
31                 con.close();
32                 psmt.close();
33             } catch (SQLException e) {
34             }
35         }
36     }
37 }
```

The screenshot shows the first part of the `MarkaDao.java` file. It contains the implementation of the `markalarGetir()` method. The code uses JDBC to execute a SELECT query on the `Marka` table, ordering the results by `markaId` in ascending order. It retrieves each row into a `Marka` object and adds it to a list. Finally, it closes the connection and statement.

```
48     public static int markaEkle(Marka marka) throws SQLException {
49         int state = 0;
50         String sql = "INSERT INTO Marka(markaAdi, markaAciklama, markaLogo) VALUES(?, ?, ?);";
51         try{
52             con = ConnectionDao.getConnection();
53             psmt = con.prepareStatement(sql);
54             psmt.setString(1, marka.getMarkaAdi());
55             psmt.setString(2, marka.getMarkaAciklama());
56             psmt.setBlob(3, marka.getStreamImg());
57             state = psmt.executeUpdate();
58             con.close();
59         }catch(SQLException ex){
60             ex.printStackTrace();
61         }finally {
62             try {
63                 con.close();
64                 psmt.close();
65             } catch (SQLException e) {
66
67                 e.printStackTrace();
68             }
69         }
70         return state;
71     }
72
73     public static int markaSil(Marka marka) throws SQLException {
74         int state = 0;
75         String sql = "DELETE FROM Marka WHERE markaId = ?";
76         try{
77             con = ConnectionDao.getConnection();
78             psmt = con.prepareStatement(sql);
79             psmt.setInt(1, marka.getMarkaId());
80             state = psmt.executeUpdate();
81             con.close();
82         }catch(SQLException ex){
83
84         }
85     }
86 }
```

The screenshot shows the second part of the `MarkaDao.java` file. It contains the implementations of the `markaEkle()` and `markaSil()` methods. The `markaEkle()` method inserts a new record into the `Marka` table using the provided `Marka` object. The `markaSil()` method deletes a record from the `Marka` table where the `markaId` matches the one provided in the `Marka` object.

```
MarkaDao.java
85     try {
86         con.close();
87         psmt.close();
88     } catch (SQLException e) {
89         e.printStackTrace();
90     }
91     return state;
92 }
93 public static int markaLogoGuncelle(Marka marka) throws SQLException {
94     int state = 0;
95     String sql = "UPDATE Marka set markaAdi = ?, markaAciklama = ?, markaLogo = ? WHERE markaId = ?";
96     try{
97         con = ConnectionDao.getConnection();
98         psmt = con.prepareStatement(sql);
99         psmt.setString(1, marka.getMarkaAdi());
100        psmt.setString(2, marka.getMarkaAciklama());
101        psmt.setBlob(3, marka.getStreamImg());
102        psmt.setInt(4, marka.getMarkaId());
103        state = psmt.executeUpdate();
104        con.close();
105    }catch(SQLException ex){
106        ex.printStackTrace();
107    }finally {
108        try {
109            con.close();
110            psmt.close();
111        } catch (SQLException e) {
112            e.printStackTrace();
113        }
114    }
115 }
116 return state;
117 }
118 }
```

Writable Smart Insert 1:1

```
MarkaDao.java
109     ex.printStackTrace();
110 }finally {
111     try {
112         con.close();
113         psmt.close();
114     } catch (SQLException e) {
115         e.printStackTrace();
116     }
117 }
118 return state ;
119 }
120 }

122 public static byte[] getImageByMarkaId(int markaID){
123     String query = "select markaLogo from Marka where markaId=?";
124     Connection con = ConnectionDao.getConnection();
125     PreparedStatement psmt;
126     byte[] bytesImage = null;
127     try {
128         psmt = con.prepareStatement(query);
129         psmt.setInt(1, markaID);
130         ResultSet rs = psmt.executeQuery();
131         if(rs.next()){
132             bytesImage = rs.getBytes("MarkaLogo");
133         }
134         con.close();
135     } catch (SQLException e) {
136         e.printStackTrace();
137     }
138     return bytesImage;
139 }
140 }
141 }
142 }
```

Writable Smart Insert 1:1

MüşteriDao.java

```
1 package com.eticaret.DAO;
2
3 import java.sql.Connection;
4
5 public class MusteriDao {
6     private static Connection con=null;
7     private static PreparedStatement psmt=null;
8     private static ResultSet rs=null;
9
10    public static ArrayList<Musteri> butunMusteriler() throws SQLException {
11        ArrayList <Musteri> butunMusteriler= new ArrayList <Musteri>();
12        String sql= "SELECT m.musteriId,m.musteriAdi,m.musteriSoyadi,m.musteriMail,m.musteriSifre,m.musteriSehir,y.yetkiAdi"
13        + "FROM Musteri As m INNER JOIN Yetki As y ON (m.yetkiId=y.yetkiId) ";
14
15        try{
16            con=ConnectionDao.getConnection();
17            psmt=con.prepareStatement(sql);
18            rs=psmt.executeQuery();
19
20            while (rs.next()){
21                Musteri musteri=new Musteri();
22                musteri.setMusteriId(rs.getInt(1));
23                musteri.setMusteriAdi(rs.getString(2));
24                musteri.setMusteriSoyadi(rs.getString(3));
25                musteri.setMusteriMail(rs.getString(4));
26                musteri.setMusteriSifre(rs.getString(5));
27                musteri.setMusteriSehir(rs.getString(6));
28                musteri.setYetki_id(rs.getInt(7));
29
30                butunMusteriler.add(musteri);
31            }
32
33        } finally {
34            con.close();
35        }
36    }
37}
```

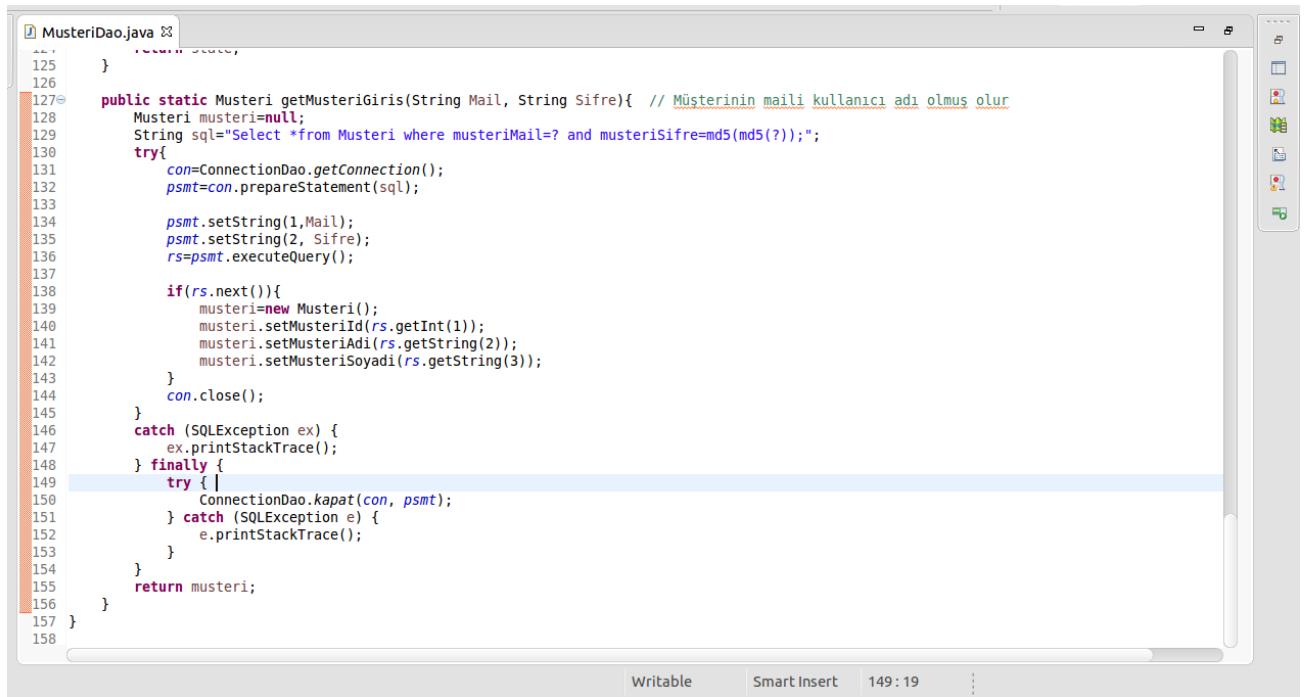
Writable Smart Insert 149 : 19

```
41         con.close();
42     }
43     catch(SQLException e) {
44         e.printStackTrace();
45     }
46     finally{
47         ConnectionDao.kapat(con, psmt,rs);
48     }
49
50     return butunMusteriler();
51 }
52
53    public static int musterilerEkle(Musteri musteri){
54        int state=0;
55        String sql="INSERT INTO Musteri(musteriAdi,musteriSoyadi,musteriMail,musteriSifre,musteriSehir, Yetki_yetkiId) VALUES(?,?,?,?,?,md5(md5(?)
56        try{
57            con=ConnectionDao.getConnection();
58            psmt=con.prepareStatement(sql);
59
60            psmt.setString(1, musteri.getMusteriAdi());
61            psmt.setString(2, musteri.getMusteriSoyadi());
62            psmt.setString(3, musteri.getMusteriMail());
63            psmt.setString(4, musteri.getMusteriSifre());
64            psmt.setString(5, musteri.getMusteriSehir());
65            psmt.setInt(6, musteri.getYetki_id());
66
67            state=psmt.executeUpdate();
68        }
69        catch(SQLException ex){
70            ex.printStackTrace();
71        }
72        finally {
73            try {
74                ConnectionDao.kapat(con, psmt);
75            }
```

Writable Smart Insert 149 : 19

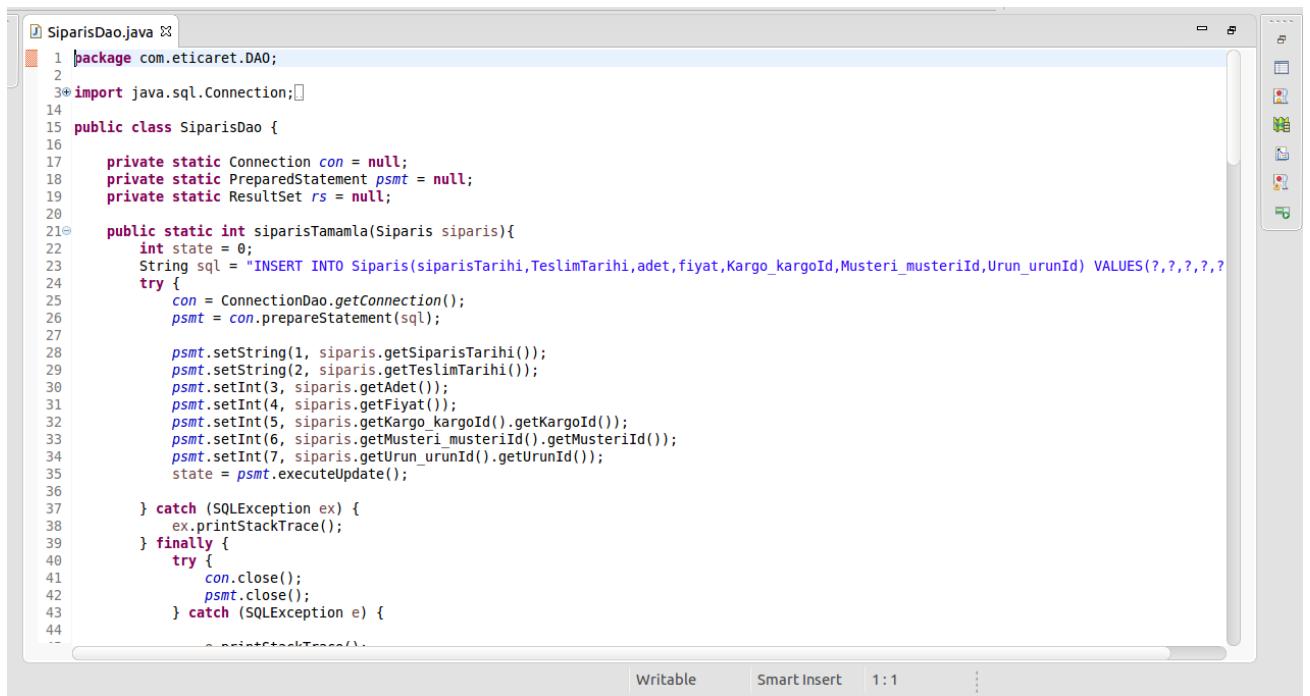
```
1 MusteriDao.java ☐
2
3     }
4     }
5     }
6     }
7     }
8     }
9     }
10    }
11    }
12    }
13    }
14    }
15    }
16    }
17    }
18    }
19    }
20    }
21    }
22    }
23    }
24    }
25    }
26    }
27    }
28    }
29    }
30    }
31    }
32    }
33    }
34    }
35    }
36    }
37    }
38    }
39    }
40    }
41    }
42    }
43    }
44    }
45    }
46    }
47    }
48    }
49    }
50    }
51    }
52    }
53    }
54    }
55    }
56    }
57    }
58    }
59    }
60    }
61    }
62    }
63    }
64    }
65    }
66    }
67    }
68    }
69    }
70    }
71    }
72    }
73    }
74    }
75    }
76    }
77    }
78    }
79    }
80    }
81    }
82    public static synchronized int musterisil(int musteriId) throws SQLException {
83        int state=0;
84        String sql="Delete from Musteri where musteriId=?;";
85        try{
86            con= ConnectionDao.getConnection();
87            psmt= con.prepareStatement(sql);
88            psmt.setInt(1, musteriId);
89            state=psmt.executeUpdate();
90        }
91        }
92        }
93        }
94        catch (SQLException ex){
95            ex.printStackTrace();
96        }
97        finally{
98            ConnectionDao.kapat(con, psmt);
99        }
100       }
101      }
102      }
103      public static synchronized int musteriguncelle(Musteri musteri) throws SQLException {
104          int state=0;
105          String sql="Update from Musteri set musteriAdi=?, musteriSoyadi=?, musteriMail=?, musterisifre=md5(md5(?)), musterisehir=? where musteriId=?";
106          try{
107              con=ConnectionDao.getConnection();
108              psmt=con.prepareStatement(sql);
109              psmt.setString(1, musteri.getmusteriAdi());
110              psmt.setString(2, musteri.getmusteriSoyadi());
111              psmt.setString(3, musteri.getmusteriMail());
112              psmt.setString(4, musteri.getmusterisifre());
113              psmt.setString(5, musteri.getmusterisehir());
114              psmt.setInt(6, musteri.getmusteriId());
115              state=psmt.executeUpdate();
116          }
117          }
118          }
119          }
120          }
121          }
122          }
123          }
124          }
125          }
126          }
127          public static Musteri getMusteriGiris(String Mail, String Sifre){ // Müşterinin maili kullanıcı adı olmuş olur
128              Musteri musteri=null;
129              String sql="Select *from Musteri where musteriMail=? and musterisifre=md5(md5(?));";
130              try{
131                  con=ConnectionDao.getConnection();
132                  psmt=con.prepareStatement(sql);
133
134                  psmt.setString(1, Mail);
135                  psmt.setString(2, Sifre);
136                  rs=psmt.executeQuery();
137
138                  if(rs.next()){
139                      musteri=new Musteri();
140                      musteri.setmusteriId(rs.getInt(1));
141                      musteri.setmusteriAdi(rs.getString(2));
142                      musteri.setmusteriSoyadi(rs.getString(3));
143                  }
144                  con.close();
145              }
146              catch (SQLException ex) {
147                  ex.printStackTrace();
148              } finally {
149                  try {
150                      ConnectionDao.kapat(con, psmt);
151                  } catch (SQLException e) {
152                      e.printStackTrace();
153                  }
154              }
155          }
156      }
157  }
```

```
1 MusteriDao.java ☐
2
3     }
4     }
5     }
6     }
7     }
8     }
9     }
10    }
11    }
12    }
13    }
14    }
15    }
16    }
17    }
18    }
19    }
20    }
21    }
22    }
23    }
24    }
25    }
26    }
27    }
28    }
29    }
30    }
31    }
32    }
33    }
34    }
35    }
36    }
37    }
38    }
39    }
40    }
41    }
42    }
43    }
44    }
45    }
46    }
47    }
48    }
49    }
50    }
51    }
52    }
53    }
54    }
55    }
56    }
57    }
58    }
59    }
50    }
51    }
52    }
53    }
54    }
55    }
56    }
57    }
58    }
59    }
60    }
61    }
62    }
63    }
64    }
65    }
66    }
67    }
68    }
69    }
60    }
61    }
62    }
63    }
64    }
65    }
66    }
67    }
68    }
69    }
70    }
71    }
72    }
73    }
74    }
75    }
76    }
77    }
78    }
79    }
70    }
71    }
72    }
73    }
74    }
75    }
76    }
77    }
78    }
79    }
80    }
81    }
82    }
83    }
84    }
85    }
86    }
87    }
88    }
89    }
80    }
81    }
82    }
83    }
84    }
85    }
86    }
87    }
88    }
89    }
90    }
91    }
92    }
93    }
94    }
95    }
96    }
97    }
98    }
99    }
100   }
101  }
102  }
103  }
104  }
105  }
106  }
107  }
108  }
109  }
100  }
101  }
102  }
103  }
104  }
105  }
106  }
107  }
108  }
109  }
110  }
111  }
112  }
113  }
114  }
115  }
116  }
117  }
118  }
119  }
110  }
111  }
112  }
113  }
114  }
115  }
116  }
117  }
118  }
119  }
120  }
121  }
122  }
123  }
124  }
125  }
126  }
127  }
128  }
129  }
130  }
131  }
132  }
133  }
134  }
135  }
136  }
137  }
138  }
139  }
140  }
141  }
142  }
143  }
144  }
145  }
146  }
147  }
148  }
149  }
150  }
151  }
152  }
153  }
154  }
155  }
156  }
157  }
```



```
125     }
126 }
127 public static Musteri getMusteriGiris(String Mail, String Sifre){ // Müsterinin maili kullanici adı olmuş olur
128     Musteri musteri=null;
129     String sql="Select *from Musteri where musterimail=? and musterisifre=md5(md5(?));";
130     try{
131         con=ConnectionDao.getConnection();
132         psmt=con.prepareStatement(sql);
133
134         psmt.setString(1,Mail);
135         psmt.setString(2, Sifre);
136         rspsmt.executeQuery();
137
138         if(rs.next()){
139             musteri=new Musteri();
140             musteri.setMusteriid(rs.getInt(1));
141             musteri.setMusteriAdi(rs.getString(2));
142             musteri.setMusteriSoyadi(rs.getString(3));
143         }
144         con.close();
145     } catch (SQLException ex) {
146         ex.printStackTrace();
147     } finally {
148         try {
149             ConnectionDao.kapat(con, psmt);
150         } catch (SQLException e) {
151             e.printStackTrace();
152         }
153     }
154     return musteri;
155 }
156 }
157 }
158 }
```

SiparisDao.java



```
1 package com.eticaret.DAO;
2
3 import java.sql.Connection;
4
5 public class SiparisDao {
6
7     private static Connection con = null;
8     private static PreparedStatement psmt = null;
9     private static ResultSet rs = null;
10
11    public static int siparisTamamla(Siparis siparis){
12        int state = 0;
13        String sql = "INSERT INTO Siparis(siparisTarihi,TeslimTarihi,adet,fiyat,Kargo_kargoId,Musteri_musteriid,Urun_urunId) VALUES(?,?,?,?,?,?,?)";
14        try {
15            con = ConnectionDao.getConnection();
16            psmt = con.prepareStatement(sql);
17
18            psmt.setString(1, siparis.getSiparisTarihi());
19            psmt.setString(2, siparis.getTeslimTarihi());
20            psmt.setInt(3, siparis.getAdet());
21            psmt.setInt(4, siparis.getFiyat());
22            psmt.setInt(5, siparis.getKargo_kargoId().getKargoId());
23            psmt.setInt(6, siparis.getMusteri_musteriid().getMusteriid());
24            psmt.setInt(7, siparis.getUrun_urunId().getUrundai());
25            state = psmt.executeUpdate();
26
27        } catch (SQLException ex) {
28            ex.printStackTrace();
29        } finally {
30            try {
31                con.close();
32                psmt.close();
33            } catch (SQLException e) {
34                e.printStackTrace();
35            }
36        }
37    }
38
39 }
```

The screenshot shows a Java code editor window titled "SiparisDao.java". The code is a static method that retrieves a list of orders from a database. It uses JDBC to connect to a database, prepare a statement, and execute a query. The result set is then iterated through to create new Siparis objects and add them to a list. Finally, the connection is closed. Error handling is included with try-catch blocks and stack trace printing.

```
48     return state;
49 }
50 public static ArrayList<Siparis> tumSiparisleriGetir() throws SQLException {
51     ArrayList<Siparis> tumSiparisler = new ArrayList<Siparis>();
52     try {
53         con = ConnectionDao.getConnection();
54         psmt = con.prepareStatement(siparisSQL);
55         rs = psmt.executeQuery();
56         while (rs.next()) {
57             Siparis siparis = new Siparis();
58
59             siparis.setSiparisId(rs.getInt(1));
60             siparis.setSiparisTarihi(rs.getString(2));
61             siparis.setTeslimTarihi(rs.getString(3));
62             siparis.setAdet(rs.getInt(4));
63             siparis.setFiyat(rs.getInt(5));
64             siparis.setKargo_kargoId(new Kargo(rs.getString(6)));
65             siparis.setMusteri_musteriId(new Musteri(rs.getString(7), rs.getString(8)));
66             siparis.setUrun_urunId(new Urun(rs.getInt(9), rs.getString(10)));
67             //urunAd,musteriAdi,musteriSoyadi,firmaAdi
68             tumSiparisler.add(siparis);
69     }
70     con.close();
71 } catch (SQLException ex) {
72     ex.printStackTrace();
73 } finally {
74     try {
75         con.close();
76         psmt.close();
77         rs.close();
78     } catch (SQLException e) {
79         e.printStackTrace();
80     }
81 }
```

The screenshot shows a Java code editor window titled "SiparisDao.java". The code contains two static methods: "musteriAdGetir" and "siparisSayisiGetir". The "musteriAdGetir" method retrieves a customer's name and surname from the database using a prepared statement and returns it as a string. The "siparisSayisiGetir" method retrieves the count of orders from the database. Both methods use JDBC to connect to a database, prepare statements, and execute queries. They include try-catch blocks for error handling and stack trace printing.

```
85
86 public static String musteriAdGetir(int musteriId){
87     String adsoyad = null;
88     String sql = "Select musteriAdi,musteriSoyadi from Musteri where musteriId=?;";
89     try {
90         con = ConnectionDao.getConnection();
91         psmt = con.prepareStatement(sql);
92         psmt.setInt(1, musteriId);
93
94         rs = psmt.executeQuery();
95
96         if(rs.next()){
97             adSoyad = rs.getString(1) + " " + rs.getString(2).toUpperCase();
98         }
99
100        con.close();
101    } catch (SQLException ex) {
102        ex.printStackTrace();
103    } finally {
104        try {
105            con.close();
106            psmt.close();
107            rs.close();
108        } catch (SQLException e) {
109            e.printStackTrace();
110        }
111    }
112 }
113
114 return adSoyad;
115 }
116
117 public static int siparisSayisiGetir(){
118     int adet = 0;
119     String sql = "Select count(urunId) from Cinsinici..
```

```
114     return dusayisi;
115 }
116
117 public static int siparisSayisiGetir(){
118     int adet = 0;
119     String sql = "Select count(siparisId) from Siparis";
120     try {
121         con = ConnectionDao.getConnection();
122         psmt = con.prepareStatement(sql);
123
124         rs = psmt.executeQuery();
125
126         if(rs.next()){
127             int sayi=(rs.getInt(1));
128             adet=sayi;
129         }
130     }
131
132     con.close();
133 } catch (SQLException ex) {
134     ex.printStackTrace();
135 } finally {
136     try {
137         con.close();
138         psmt.close();
139         rs.close();
140     } catch (SQLException e) {
141         e.printStackTrace();
142     }
143 }
144
145 }
146
147 }
```

The screenshot shows a Java code editor window with the file name 'SiparisDao.java'. The code is a static method named 'siparisSayisiGetir' which returns an integer. It uses a try-with-resources block to handle database connections and statements. It executes a SQL query to count the number of rows in the 'Siparis' table and returns the result.

```
147 public static int siparisSil(int siparisId){
148     try {
149         con = ConnectionDao.getConnection();
150         psmt = con.prepareStatement("Delete from Siparis where siparisId="+siparisId+";");
151         psmt.executeUpdate();
152         con.close();
153     } catch (SQLException ex) {
154         ex.printStackTrace();
155     } finally {
156         try {
157             con.close();
158             psmt.close();
159         } catch (SQLException e) {
160             e.printStackTrace();
161         }
162     }
163     return 1;
164 }
165
166
167 public static String siparisSQL = "SELECT "+
168 "S.siparisId, "+
169 "S.siparisTarihi, "+
170 "S.teslimTarihi, "+
171 "S.adet, "+
172 "S.fiyat, "+
173 "K.firmaAdi, "+
174 "M.musteriAdi, "+
175 "M.musteriSoyadi,U.urunId,U.urunAd "+
176 "FROM "+
177 "Musteri M INNER JOIN Siparis S ON M.musteriId = S.Musteri_musteriId "+
178 "INNER JOIN Kargo K ON S.Kargo_kargoId = K.kargoId "+
179 "INNER JOIN Urun U ON S.Urun_urunId = U.urunId";
180 }
```

The screenshot shows a Java code editor window with the file name 'SiparisDao.java'. The code is a static method named 'siparisSil' which takes an integer parameter 'siparisId'. It uses a try-with-resources block to handle database connections and statements. It executes a SQL DELETE statement to remove the row from the 'Siparis' table where the 'siparisId' matches the input parameter.

UrunDao.java

```
1 package com.eticaret.DAO;
2
3* import java.sql.Connection;*
4
5 public class UrunDao {
6     private static Connection con = null;
7     private static PreparedStatement psmt = null;
8     private static ResultSet rs = null;
9
10    public static ArrayList<Urun> tumUrunleriKategoriyeGoreGetir(int kid) throws SQLException {
11        ArrayList<Urun> tumUrunler = new ArrayList<Urun>();
12        try {
13            con = ConnectionDao.getConnection();
14            psmt = con.prepareStatement("Select * from Urun where Kategori_kategoriId =?;");
15            psmt.setInt(1, kid);
16            rs = psmt.executeQuery();
17            while (rs.next()) {
18                Urun urun = new Urun();
19                urun.setUrunId(rs.getInt(1));
20                urun.setUrunAd(rs.getString(2));
21                urun.setUrunAciklama(rs.getString(3));
22                urun.setUrunFiyat(rs.getInt(4));
23                urun.setUrunMiktar(rs.getInt(5));
24                urun.setKategori_kategoriId(rs.getInt(8));
25                tumUrunler.add(urun);
26            }
27            con.close();
28        } catch (SQLException ex) {
29            ex.printStackTrace();
30        } finally {
31            try {
32                con.close();
33                psmt.close();
34            } catch (SQLException e) {
35            }
36        }
37    }
38}
39}
40}
41}
42}
43}
44}
45}
46}
47}
48}
49}
50}
51}
52}
53}
54}
55}
56}
57}
58}
59}
60}
61}
62}
63}
64}
65}
66}
67}
68}
69}
70}
71}
72}
73}
```

The code is a Java class named UrunDao. It contains two static methods: tumUrunleriKategoriyeGoreGetir(int kid) and tumUrunleriGetir(). The first method takes an integer parameter 'kid' and returns an ArrayList of Urun objects. It uses a PreparedStatement to execute a SQL query that filters products by category ID. The second method returns all products from the database. Both methods handle SQLExceptions and close their resources (Connection, PreparedStatement, ResultSet) in a finally block.

```
40        rs.close();
41    } catch (SQLException e) {
42        e.printStackTrace();
43    }
44}
45}
46}
47}
48}
49}
50}
51}
52}
53}
54}
55}
56}
57}
58}
59}
60}
61}
62}
63}
64}
65}
66}
67}
68}
69}
70}
71}
72}
73}
```

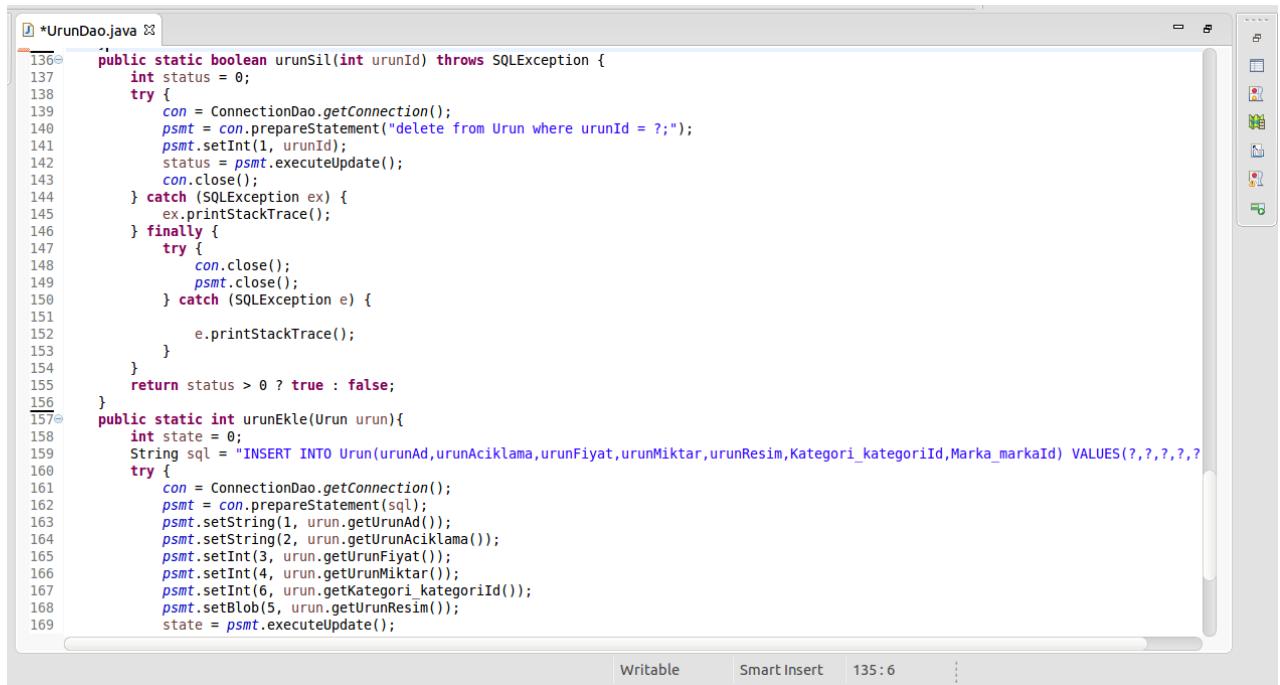
This is the continuation of the UrunDao.java code. It shows the completion of the tumUrunleriKategoriyeGoreGetir() and tumUrunleriGetir() methods. The tumUrunleriKategoriyeGoreGetir() method adds each retrieved product to a list and returns it. The tumUrunleriGetir() method performs a similar operation but without filtering by category ID. Both methods ensure proper resource cleanup by closing the ResultSet, PreparedStatement, and Connection in a finally block.

```
Urundao.java
73         rs.close();
74     } catch (SQLException e) {
75
76         e.printStackTrace();
77     }
78     return tumUrunler;
79 }
80 }
81 public static byte[] getImageByUrunId(int urunId) {
82     String query = "select urunResim from Urun where urunId=?";
83     Connection con = ConnectionDao.getConnection();
84     PreparedStatement psmt;
85     byte[] bytesImage = null;
86     try {
87         psmt = con.prepareStatement(query);
88         psmt.setInt(1, urunId);
89         ResultSet rs = psmt.executeQuery();
90         if (rs.next()) {
91             bytesImage = rs.getBytes("urunResim");
92         }
93         con.close();
94     } catch (SQLException e) {
95         e.printStackTrace();
96     }
97     return bytesImage;
98 }
99 }
100
101
102 public static ArrayList<Urun> tekUrunGetir(int urunID) throws SQLException {
103     ArrayList<Urun> urun = new ArrayList<Urun>();
104     try {
105         con = ConnectionDao.getConnection();
106         psmt = con.prepareStatement("select * from Urun where urunId=?");
```

Writable Smart Insert 1:1

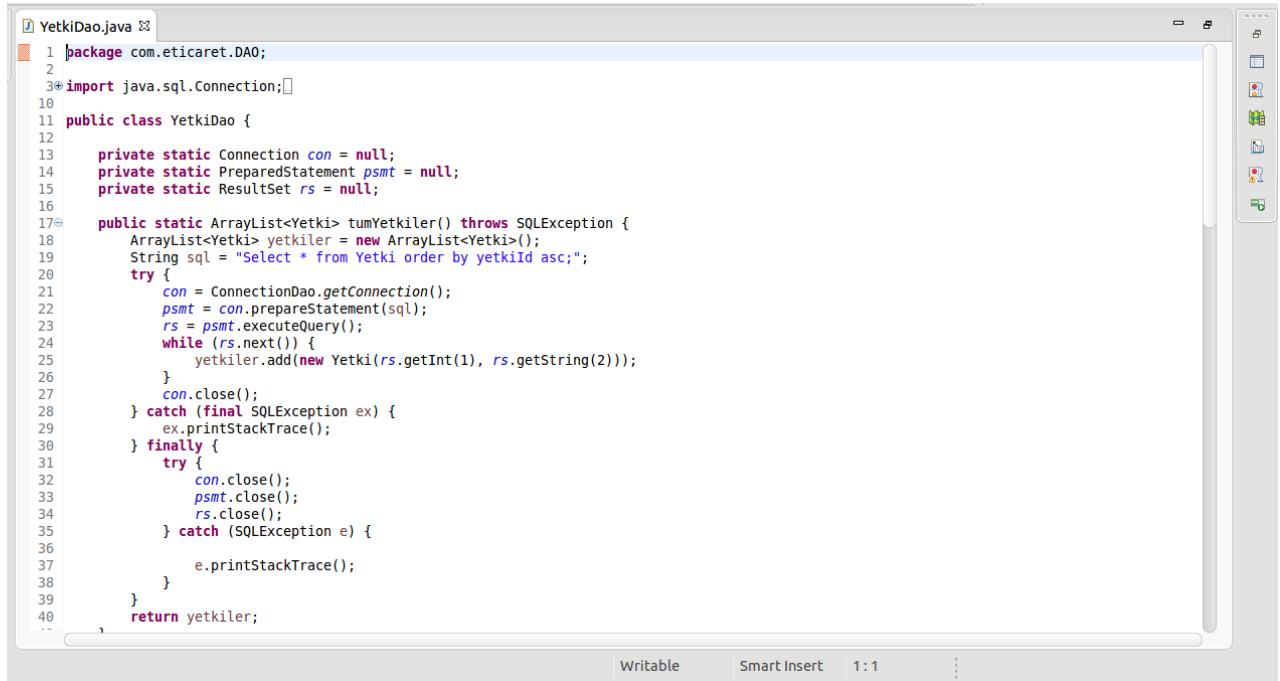
```
Urundao.java
106     psmt = con.prepareStatement("select * from Urun where urunId=?");
107     psmt.setInt(1, urunID);
108     rs = psmt.executeQuery();
109     if (rs.next()) {
110         Urun urunler = new Urun();
111
112         urunler.setUrunId(rs.getInt(1));
113         urunler.setUrunAd(rs.getString(2));
114         urunler.setUrunAciklama(rs.getString(3));
115         urunler.setUrunFiyat(rs.getInt(4));
116         urunler.setUrunMiktar(rs.getInt(5));
117         urunler.setKategori_kategoriId(rs.getInt(8));
118         urun.add(urunler);
119     }
120     con.close();
121 } catch (SQLException ex) {
122     ex.printStackTrace();
123 } finally {
124     try {
125         con.close();
126         psmt.close();
127         rs.close();
128     } catch (SQLException e) {
129         e.printStackTrace();
130     }
131 }
132 }
133 }
134 return urun;
135 }
136
137 public static boolean urunSil(int urunId) throws SQLException {
138     int status = 0;
139     try {
```

Writable Smart Insert 1:1



```
136     public static boolean urunSil(int urunId) throws SQLException {
137         int status = 0;
138         try {
139             con = ConnectionDao.getConnection();
140             psmt = con.prepareStatement("delete from Urun where urunId = ?;");
141             psmt.setInt(1, urunId);
142             status = psmt.executeUpdate();
143             con.close();
144         } catch (SQLException ex) {
145             ex.printStackTrace();
146         } finally {
147             try {
148                 con.close();
149                 psmt.close();
150             } catch (SQLException e) {
151                 e.printStackTrace();
152             }
153         }
154     }
155     return status > 0 ? true : false;
156 }
157 public static int urunEkle(Urun urun){
158     int state = 0;
159     String sql = "INSERT INTO Urun(urunAd,urunAciklama,urunFiyat,urunMiktar,urunResim,Kategori_kategoriId,Marka_markaId) VALUES(?,?,?,?,?,?,?,?)";
160     try {
161         con = ConnectionDao.getConnection();
162         psmt = con.prepareStatement(sql);
163         psmt.setString(1, urun.getUrunAd());
164         psmt.setString(2, urun.getUrunAciklama());
165         psmt.setInt(3, urun.getUrunFiyat());
166         psmt.setInt(4, urun.getUrunMiktar());
167         psmt.setInt(6, urun.getKategori_kategoriId());
168         psmt.setBlob(5, urun.getUrunResim());
169         state = psmt.executeUpdate();
170     }
```

Yetki Dao.java



```
1 package com.eticaret.DAO;
2
3 import java.sql.Connection;
4
5 public class YetkiDAO {
6
7     private static Connection con = null;
8     private static PreparedStatement psmt = null;
9     private static ResultSet rs = null;
10
11     public static ArrayList<Yetki> tumYetkiler() throws SQLException {
12         ArrayList<Yetki> yetkiler = new ArrayList<Yetki>();
13         String sql = "Select * from Yetki order by yetkiId asc;";
14         try {
15             con = ConnectionDao.getConnection();
16             psmt = con.prepareStatement(sql);
17             rs = psmt.executeQuery();
18             while (rs.next()) {
19                 yetkiler.add(new Yetki(rs.getInt(1), rs.getString(2)));
20             }
21             con.close();
22         } catch (final SQLException ex) {
23             ex.printStackTrace();
24         } finally {
25             try {
26                 con.close();
27                 psmt.close();
28                 rs.close();
29             } catch (SQLException e) {
30                 e.printStackTrace();
31             }
32         }
33     }
34     return yetkiler;
35 }
```

The screenshot shows a Java code editor window titled "YetkiDao.java". The code implements a DAO interface for managing "Yetki" objects. It includes methods for saving, updating, and deleting Yetki objects, along with their respective error handling logic. The code uses JDBC to interact with a database.

```
76    }
77    }
78    }
79    }
80    }
81    }
82    }
83    }
84    }
85    }
86    }
87    }
88    }
89    }
90    public static int yetkiGuncelle(Yetki yetki) throws SODException {
91        int state = 0;
92        String sql = "UPDATE Yetki Set yetkiAdi = ? Where yetkiId = ?";
93        try {
94            con = ConnectionDao.getConnection();
95            psmt = con.prepareStatement(sql);
96            psmt.setString(1, yetki.getYetkiAdi());
97            psmt.setInt(2, yetki.getYetkiId());
98            state = psmt.executeUpdate();
99            con.close();
100        } catch (SQLException ex) {
101            ex.printStackTrace();
102        } finally {
103            try {
104                con.close();
105                psmt.close();
106                rs.close();
107            } catch (SQLException e) {
108                e.printStackTrace();
109            }
110        }
111    }
112    }
113    return state;
114}
```

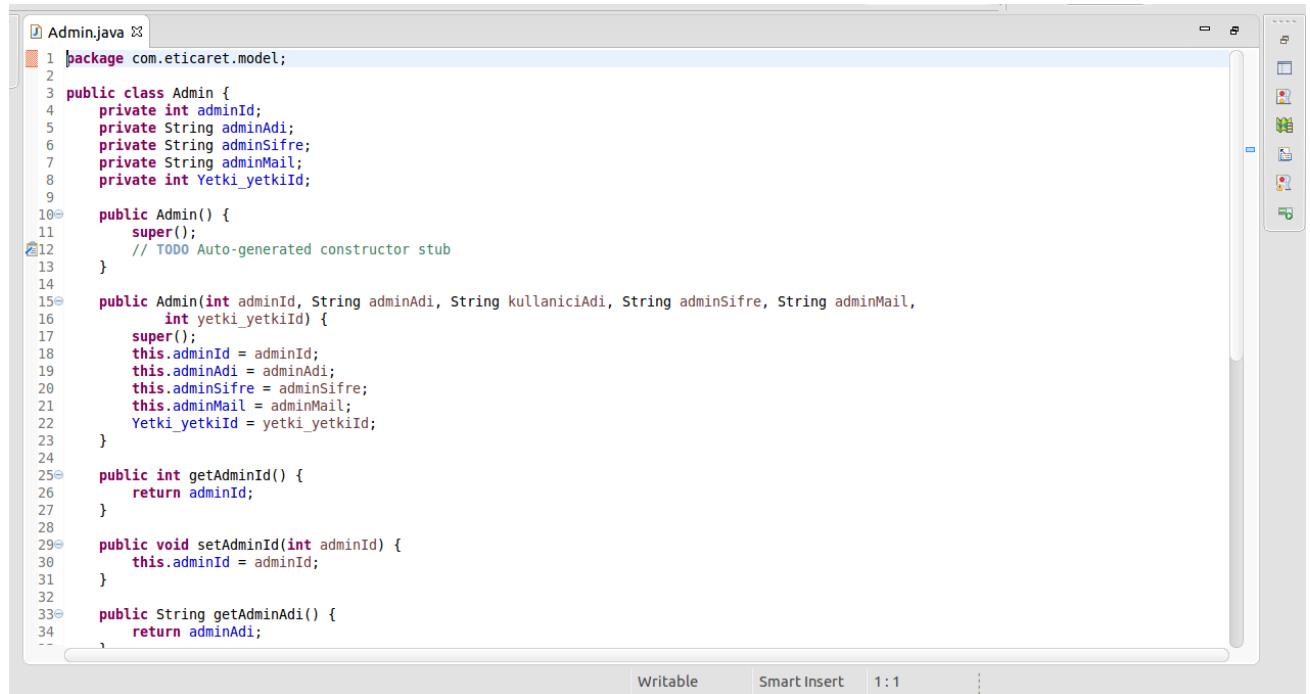
This screenshot shows the same Java code as the first one, but with some changes. The code now includes a "yetkiGuncelle" method that takes a "Yetki" object as a parameter and returns an integer. The "try" block has been moved outside the "finally" block, and the "state" variable is now being returned from the method. The code still handles SQLExceptions and closes database resources.

```
76    }
77    }
78    }
79    }
80    }
81    }
82    }
83    }
84    }
85    }
86    }
87    }
88    }
89    }
90    public static int yetkiGuncelle(Yetki yetki) throws SODException {
91        int state = 0;
92        String sql = "UPDATE Yetki Set yetkiAdi = ? Where yetkiId = ?";
93        try {
94            con = ConnectionDao.getConnection();
95            psmt = con.prepareStatement(sql);
96            psmt.setString(1, yetki.getYetkiAdi());
97            psmt.setInt(2, yetki.getYetkiId());
98            state = psmt.executeUpdate();
99            con.close();
100        } catch (SQLException ex) {
101            ex.printStackTrace();
102        } finally {
103            try {
104                con.close();
105                psmt.close();
106                rs.close();
107            } catch (SQLException e) {
108                e.printStackTrace();
109            }
110        }
111    }
112    }
113    return state;
114}
```

Model Sınıfları

- ▼ com.eticaret.model
 - ▶ Admin.java
 - ▶ Kargo.java
 - ▶ Kategori.java
 - ▶ Marka.java
 - ▶ Musteri.java
 - ▶ Siparis.java
 - ▶ Urun.java

Admin.java



The screenshot shows a Java code editor window with the Admin.java file open. The code defines a class named Admin with private fields for adminId, adminAdi, adminSifre, adminMail, and Yetki_yetkiId. It includes a constructor that takes these parameters and initializes the fields. There are also getter and setter methods for each field.

```
1 package com.eticaret.model;
2
3 public class Admin {
4     private int adminId;
5     private String adminAdi;
6     private String adminSifre;
7     private String adminMail;
8     private int Yetki_yetkiId;
9
10    public Admin() {
11        super();
12        // TODO Auto-generated constructor stub
13    }
14
15    public Admin(int adminId, String adminAdi, String kullaniciAdi, String adminSifre, String adminMail,
16                int yetki_yetkiId) {
17        super();
18        this.adminId = adminId;
19        this.adminAdi = adminAdi;
20        this.adminSifre = adminSifre;
21        this.adminMail = adminMail;
22        Yetki_yetkiId = yetki_yetkiId;
23    }
24
25    public int getAdminId() {
26        return adminId;
27    }
28
29    public void setAdminId(int adminId) {
30        this.adminId = adminId;
31    }
32
33    public String getAdminAdi() {
34        return adminAdi;
35    }
36}
```

At the bottom of the editor, there are buttons for 'Writable', 'Smart Insert', and a ratio of '1:1'.

The screenshot shows a Java code editor window with the file `Admin.java` open. The code defines a class `Admin` with various getters and setters for properties like `adminAdi`, `adminSifre`, `adminMail`, and `yetki_yetkiId`. The code is annotated with line numbers from 33 to 66.

```
33 public String getAdminAdi() {
34     return adminAdi;
35 }
36
37 public void setAdminAdi(String adminAdi) {
38     this.adminAdi = adminAdi;
39 }
40
41 public String getAdminSifre() {
42     return adminSifre;
43 }
44
45 public void setAdminSifre(String adminSifre) {
46     this.adminSifre = adminSifre;
47 }
48
49 public String getAdminMail() {
50     return adminMail;
51 }
52
53 public void setAdminMail(String adminMail) {
54     this.adminMail = adminMail;
55 }
56
57 public int getYetki_yetkiId() {
58     return Yetki_yetkiId;
59 }
60
61 public void setYetki_yetkiId(int yetki_yetkiId) {
62     Yetki_yetkiId = yetki_yetkiId;
63 }
64
65 }
66 }
```

Kargo.java

The screenshot shows a Java code editor window with the file `Kargo.java` open. The code defines a class `Kargo` with constructors that initialize properties `kargoId`, `firmaAdi`, and `fiyat`. The code is annotated with line numbers from 1 to 34.

```
1 package com.eticaret.model;
2
3 public class Kargo {
4     private int kargoId;
5     private String firmaAdi;
6     private String fiyat;
7
8     public Kargo() {
9         super();
10        // TODO Auto-generated constructor stub
11    }
12
13     public Kargo(int kargoId) {
14         super();
15         this.kargoId = kargoId;
16     }
17
18     public Kargo(String firmaAdi) {
19         super();
20         this.firmaAdi = firmaAdi;
21     }
22
23     public Kargo(int kargoId, String firmaAdi) {
24         super();
25         this.kargoId = kargoId;
26         this.firmaAdi = firmaAdi;
27     }
28
29     public Kargo(String firmaAdi, String fiyat) {
30         super();
31         this.firmaAdi = firmaAdi;
32         this.fiyat = fiyat;
33     }
34 }
```

The screenshot shows a Java code editor window with the file `Kargo.java` open. The code defines a class `Kargo` with private fields `kargoId`, `firmaAdi`, and `fiyat`. It includes constructor, getters, and setters for these fields.

```
35 public Kargo(int kargoId, String firmaAdi, String fiyat) {
36     super();
37     this.kargoId = kargoId;
38     this.firmaAdi = firmaAdi;
39     this.fiyat = fiyat;
40 }
41
42
43
44 public int getKargoId() {
45     return kargoId;
46 }
47
48 public void setKargoId(int kargoId) {
49     this.kargoId = kargoId;
50 }
51
52 public String getFirmaAdi() {
53     return firmaAdi;
54 }
55
56 public void setFirmaAdi(String firmaAdi) {
57     this.firmaAdi = firmaAdi;
58 }
59
60 public String getFiyat() {
61     return fiyat;
62 }
63
64 public void setFiyat(String fiyat) {
65     this.fiyat = fiyat;
66 }
67
68 }
```

Kategori.java

The screenshot shows a Java code editor window with the file `Kategori.java` open. The code defines a class `Kategori` with private fields `kategoriId` and `kategoriAdi`. It includes constructor, getters, and setters for these fields.

```
1 package com.eticaret.model;
2
3 public class Kategori {
4
5     private int kategoriId;
6     private String kategoriAdi;
7
8     public Kategori() {
9         super();
10    }
11
12     public Kategori(int kategoriId, String kategoriAdi) {
13         super();
14         this.kategoriId = kategoriId;
15         this.kategoriAdi = kategoriAdi;
16     }
17
18     public int getKategoriId() {
19         return kategoriId;
20     }
21
22     public void setKategoriId(int kategoriId) {
23         this.kategoriId = kategoriId;
24     }
25
26     public String getKategoriAdi() {
27         return kategoriAdi;
28     }
29
30     public void setKategoriAdi(String kategoriAdi) {
31         this.kategoriAdi = kategoriAdi;
32     }
33 }
34 }
```

Marka.java

```
1 package com.eticaret.model;
2 import java.io.InputStream;
3
4
5 public class Marka {
6
7     private int markaId;
8     private String markaAdi;
9     private String markaAciklama;
10    private String markaLogo;
11    private InputStream streamImg;
12
13    public Marka() {
14        super();
15        // TODO Auto-generated constructor stub
16    }
17
18    public Marka(int markaId, String markaAdi, String markaAciklama, String markaLogo, InputStream streamImg) {
19        super();
20        this.markaId = markaId;
21        this.markaAdi = markaAdi;
22        this.markaAciklama = markaAciklama;
23        this.markaLogo = markaLogo;
24        this.streamImg = streamImg;
25    }
26
27    public int getMarkaId() {
28        return markaId;
29    }
30
31    public void setMarkaId(int markaId) {
32        this.markaId = markaId;
33    }
34
35    public String getMarkaAdi() {
36        return markaAdi;
37    }
38
39    public void setMarkaAdi(String markaAdi) {
40        this.markaAdi = markaAdi;
41    }
42
43    public String getMarkaAciklama() {
44        return markaAciklama;
45    }
46
47    public void setMarkaAciklama(String markaAciklama) {
48        this.markaAciklama = markaAciklama;
49    }
50
51    public String getMarkaLogo() {
52        return markaLogo;
53    }
54
55    public void setMarkaLogo(String markaLogo) {
56        this.markaLogo = markaLogo;
57    }
58
59    public InputStream getStreamImg() {
60        return streamImg;
61    }
62
63    public void setStreamImg(InputStream streamImg) {
64        this.streamImg = streamImg;
65    }
66}
```

The screenshot shows the initial state of the `Marka.java` file in a Java IDE. The code defines a class `Marka` with fields for `markaId`, `markaAdi`, `markaAciklama`, `markaLogo`, and `streamImg`. It includes a default constructor and a constructor that takes all these parameters. The code is annotated with Javadoc-style comments and contains several TODO items.

```
34
35    public String getMarkaAdi() {
36        return markaAdi;
37    }
38
39    public void setMarkaAdi(String markaAdi) {
40        this.markaAdi = markaAdi;
41    }
42
43    public String getMarkaAciklama() {
44        return markaAciklama;
45    }
46
47    public void setMarkaAciklama(String markaAciklama) {
48        this.markaAciklama = markaAciklama;
49    }
50
51    public String getMarkaLogo() {
52        return markaLogo;
53    }
54
55    public void setMarkaLogo(String markaLogo) {
56        this.markaLogo = markaLogo;
57    }
58
59    public InputStream getStreamImg() {
60        return streamImg;
61    }
62
63    public void setStreamImg(InputStream streamImg) {
64        this.streamImg = streamImg;
65    }
66}
```

The screenshot shows the completed state of the `Marka.java` file. The code has been refactored to remove the redundant code from the previous version. The class now has four standard getters and setters for its properties: `markaAdi`, `markaAciklama`, `markaLogo`, and `streamImg`.

Musteri.java

The screenshot shows a Java code editor window with the file 'Musteri.java' open. The code defines a class 'Musteri' with private fields for musteriId, musteriAdi, musteriSoyadi, musteriMail, musterisifre, musterisehir, and Yetki_yetkiId. It includes several constructors and getter/setter methods for these fields. The code is color-coded, and the interface includes standard Java IDE features like toolbars and status bars.

```
1 package com.eticaret.model;
2
3 public class Musteri {
4     private int musteriId;
5     private String musteriAdi;
6     private String musteriSoyadi;
7     private String musteriMail;
8     private String musterisifre;
9     private String musterisehir;
10    private int Yetki_yetkiId;
11
12    public Musteri(int musteriId) {
13        super();
14        this.musteriId = musteriId;
15    }
16
17    public Musteri() {
18        super();
19        // TODO Auto-generated constructor stub
20    }
21    public Musteri(String musteriAdi, String musteriSoyadi) {
22        super();
23        this.musteriAdi = musteriAdi;
24        this.musteriSoyadi = musteriSoyadi;
25    }
26    public Musteri(int musteriId, String musteriAdi, String musteriSoyadi, String musteriMail, String musterisifre,
27                    String musterisehir, int Yetki_yetkiId) {
28        super();
29        this.musteriId = musteriId;
30        this.musteriAdi = musteriAdi;
31        this.musteriSoyadi = musteriSoyadi;
32        this.musteriMail = musteriMail;
33        this.musterisifre = musterisifre;
34        this.musterisehir = musterisehir;
35    }
36
37    public int getMusteriId() {
38        return musteriId;
39    }
40
41    public void setMusteriId(int musteriId) {
42        this.musteriId = musteriId;
43    }
44    public String getMusteriAdi() {
45        return musteriAdi;
46    }
47    public void setMusteriAdi(String musteriAdi) {
48        this.musteriAdi = musteriAdi;
49    }
50    public String getMusteriSoyadi() {
51        return musteriSoyadi;
52    }
53    public void setMusteriSoyadi(String musteriSoyadi) {
54        this.musteriSoyadi = musteriSoyadi;
55    }
56
57    public String getMusteriMail() {
58        return musteriMail;
59    }
60
61    public void setMusteriMail(String musteriMail) {
62        this.musteriMail = musteriMail;
63    }
64    public String getMusterisifre() {
65        return musterisifre;
66    }
67    public void setMusterisifre(String musterisifre) {
68    }
}
```

This screenshot shows the continuation of the 'Musteri.java' file from the previous window. It contains the remaining methods and fields of the 'Musteri' class, including getters and setters for musteriSifre and musterisehir, and a final method setMusterisifre with a parameter. The code is color-coded, and the interface includes standard Java IDE features.

```
69    public void setMusterisifre(String musterisifre) {
70    }
71
72    public String getMusterisehir() {
73        return musterisehir;
74    }
75    public void setMusterisehir(String musterisehir) {
76        this.musterisehir = musterisehir;
77    }
78
79    @Override
80    public String toString() {
81        return "Musteri{" +
82                "musteriId=" + musteriId +
83                ", musteriAdi='" + musteriAdi + '\'' +
84                ", musteriSoyadi='" + musteriSoyadi + '\'' +
85                ", musteriMail=" + musteriMail +
86                ", musterisifre=" + musterisifre +
87                ", musterisehir=" + musterisehir +
88                '}';
89    }
90}
```

The screenshot shows a Java code editor window with the file 'Musteri.java' open. The code defines a class 'Musteri' with various getters and setters for its attributes: musteriSoyadi, musteriMail, musteriSifre, musteriSehir, and yetki_yetkiId. The code is color-coded, and the editor interface includes tabs for 'Writable' and 'Smart Insert' at the bottom.

```
1 package com.eticaret.model;
2
3 public class Musteri {
4     private int musteriId;
5     private String musteriSoyadi;
6     private String musteriMail;
7     private String musteriSifre;
8     private String musteriSehir;
9     private int yetki_yetkiId;
10
11     public int getMusteriId() {
12         return musteriId;
13     }
14
15     public void setMusteriId(int musteriId) {
16         this.musteriId = musteriId;
17     }
18
19     public String getMusteriSoyadi() {
20         return musteriSoyadi;
21     }
22
23     public void setMusteriSoyadi(String musteriSoyadi) {
24         this.musteriSoyadi = musteriSoyadi;
25     }
26
27     public String getMusteriMail() {
28         return musteriMail;
29     }
30
31     public void setMusteriMail(String musteriMail) {
32         this.musteriMail = musteriMail;
33     }
34
35     public String getMusteriSifre() {
36         return musteriSifre;
37     }
38
39     public void setMusteriSifre(String musteriSifre) {
40         this.musteriSifre = musteriSifre;
41     }
42
43     public String getMusteriSehir() {
44         return musteriSehir;
45     }
46
47     public void setMusteriSehir(String musteriSehir) {
48         this.musteriSehir = musteriSehir;
49     }
50
51     public int getYetki_yetkiId() {
52         return Yetki_yetkiId;
53     }
54
55     public void setYetki_yetkiId(int yetki_yetkiId) {
56         Yetki_yetkiId = yetki_yetkiId;
57     }
58 }
59
60 }
```

Siparis.java

The screenshot shows a Java code editor window with the file 'Siparis.java' open. The code defines a class 'Siparis' with attributes: siparisId, siparisTarihi, teslimTarihi, adet, fiyat, musteriId, kargoId, and urunId. It includes a constructor, getters, and setters for these attributes. The code is color-coded, and the editor interface includes tabs for 'Writable' and 'Smart Insert' at the bottom.

```
1 package com.eticaret.model;
2
3 public class Siparis {
4     private int siparisId;
5     private String siparisTarihi;
6     private String teslimTarihi;
7     private int adet;
8     private int fiyat;
9     private Musteri musteri_id;
10    private Kargo kargo_id;
11    private Urun urun_id;
12
13    public Siparis() {
14        super();
15        // TODO Auto-generated constructor stub
16    }
17    public int getSiparisId() {
18        return siparisId;
19    }
20    public void setSiparisId(int siparisId) {
21        this.siparisId = siparisId;
22    }
23    public String getSiparisTarihi() {
24        return siparisTarihi;
25    }
26    public void setSiparisTarihi(String siparisTarihi) {
27        this.siparisTarihi = siparisTarihi;
28    }
29    public String getTeslimTarihi() {
30        return teslimTarihi;
31    }
32    public void setTeslimTarihi(String teslimTarihi) {
33        this.teslimTarihi = teslimTarihi;
34    }
35 }
```

The screenshot shows a Java code editor window with the file `Siparis.java` open. The code defines a class `Siparis` with various methods for setting and getting properties like `teslimTarihi`, `adet`, `fiyat`, `Musteri_musteriId`, `Kargo_kargoId`, and `Urun_urunId`. The code is annotated with line numbers from 33 to 66. The editor interface includes tabs for `Writable` and `Smart Insert`, and a status bar showing `53:6`.

```
1 package com.eticaret.model;
2
3 import java.io.InputStream;
4
5 public class Siparis {
6
7     private int teslimTarihi;
8     private int adet;
9     private int fiyat;
10    private Musteri musteri;
11    private Kargo kargo;
12    private Urun urun;
13
14    public void setTeslimTarihi(String teslimTarihi) {
15        this.teslimTarihi = teslimTarihi;
16    }
17
18    public int getAdet() {
19        return adet;
20    }
21
22    public void setAdet(int adet) {
23        this.adet = adet;
24    }
25
26    public int getFiyat() {
27        return fiyat;
28    }
29
30    public void setFiyat(int fiyat) {
31        this.fiyat = fiyat;
32    }
33    public Musteri getMusteri() {
34        return musteri;
35    }
36
37    public void setMusteri(Musteri musteri) {
38        this.musteri = musteri;
39    }
40
41    public Kargo getKargo() {
42        return kargo;
43    }
44
45    public void setKargo(Kargo kargo) {
46        this.kargo = kargo;
47    }
48
49    public Urun getUrun() {
50        return urun;
51    }
52
53    public void setUrun(Urun urun) {
54        this.urun = urun;
55    }
56
57    public String toString() {
58        return "Siparis{" +
59                "teslimTarihi=" + teslimTarihi +
60                ", adet=" + adet +
61                ", fiyat=" + fiyat +
62                ", musteri=" + musteri +
63                ", kargo=" + kargo +
64                ", urun=" + urun +
65                '}';
66}
```

Urun.java

The screenshot shows a Java code editor window with the file `Urun.java` open. The code defines a class `Urun` with fields for `urunId`, `urunAd`, `urunAciklama`, `urunFiyat`, `urunMiktar`, `Kategori_kategoriId`, `Marka_markaId`, and `urunResim`. It includes a constructor stub and methods for setting and getting these properties. The code is annotated with line numbers from 1 to 34. The editor interface includes tabs for `Writable` and `Smart Insert`, and a status bar showing `71:6`.

```
1 package com.eticaret.model;
2
3 import java.io.InputStream;
4
5 public class Urun {
6
7     private int urunId;
8     private String urunAd;
9     private String urunAciklama;
10    private int urunFiyat;
11    private int urunMiktar;
12    private int Kategori_kategoriId;
13    private int Marka_markaId;
14    private InputStream urunResim;
15
16    public Urun() {
17        super();
18        // TODO Auto-generated constructor stub
19    }
20
21    public Urun(int urunId, String urunAd) {
22        super();
23        this.urunId = urunId;
24        this.urunAd = urunAd;
25    }
26    public int getUrundai() {
27        return urunId;
28    }
29    public void setUrundai(int urunId) {
30        this.urunId = urunId;
31    }
32    public String getUrund() {
33        return urunAd;
34    }
35}
```

The screenshot shows a Java code editor window titled "Urun.java". The code defines a class "Urun" with various methods for getting and setting properties like price, quantity, category ID, brand ID, and image stream. The code is color-coded, and the editor interface includes tabs for "Writable" and "Smart Insert" at the bottom.

```
44     }
45     public int getUrunFiyat() {
46         return urunFiyat;
47     }
48     public void setUrunFiyat(int urunFiyat) {
49         this.urunFiyat = urunFiyat;
50     }
51     public int getUrunMiktar() {
52         return urunMiktar;
53     }
54     public void setUrunMiktar(int urunMiktar) {
55         this.urunMiktar = urunMiktar;
56     }
57     public int getKategori_kategoriId() {
58         return Kategori_kategoriId;
59     }
60     public void setKategori_kategoriId(int kategori_kategoriId) {
61         Kategori_kategoriId = kategori_kategoriId;
62     }
63     public int getMarka_markaId() {
64         return Marka_markaId;
65     }
66     public void setMarka_markaId(int marka_markaId) {
67         Marka_markaId = marka_markaId;
68     }
69     public InputStream getUrunResim() {
70         return urunResim;
71     }
72     public void setUrunResim(InputStream urunResim) {
73         this.urunResim = urunResim;
74     }
75 }
76 }
77 }
```

Yetki.java

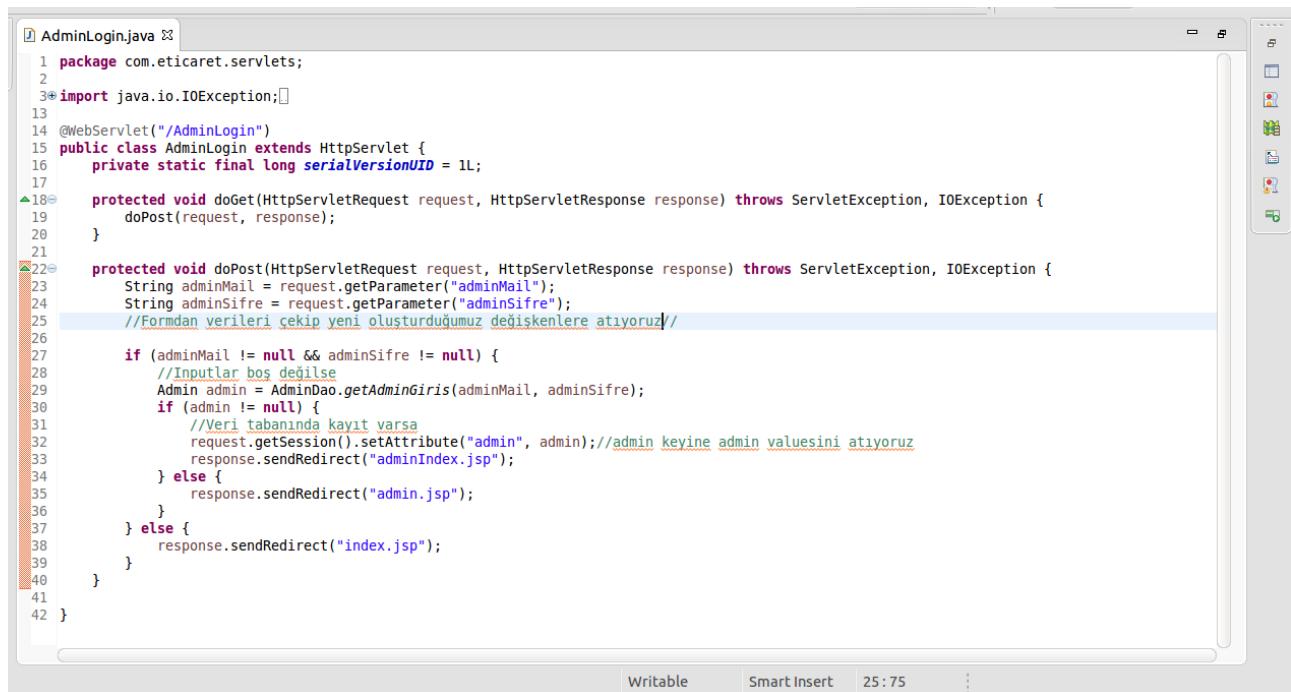
The screenshot shows a Java code editor window titled "Yetki.java". The code defines a class "Yetki" with private fields "yetkiId" and "yetkiAdi", and methods for getting and setting these values. A constructor stub is present, and a TODO comment is included. The code is color-coded, and the editor interface includes tabs for "Writable" and "Smart Insert" at the bottom.

```
1 package com.eticaret.model;
2
3 public class Yetki {
4
5     private int yetkiId;
6     private String yetkiAdi;
7
8     public Yetki() {
9         super();
10        // TODO Auto-generated constructor stub
11    }
12
13    public Yetki(int yetkiId, String yetkiAdi) {
14        super();
15        this.yetkiId = yetkiId;
16        this.yetkiAdi = yetkiAdi;
17    }
18
19    public int getYetkiId() {
20        return yetkiId;
21    }
22
23    public void setYetkiId(int yetkiId) {
24        this.yetkiId = yetkiId;
25    }
26
27    public String getYetkiAdi() {
28        return yetkiAdi;
29    }
30
31    public void setYetkiAdi(String yetkiAdi) {
32        this.yetkiAdi = yetkiAdi;
33    }
34 }
```

Controller- Servlet Sınıfları:

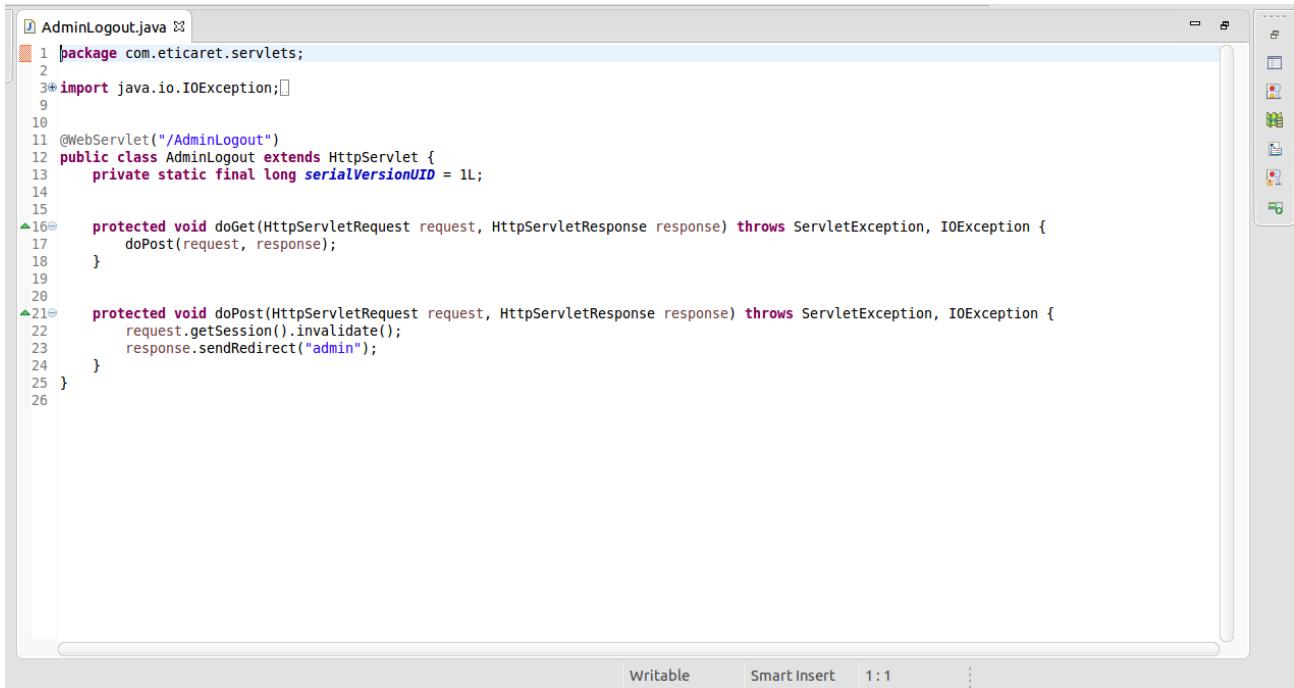
```
▼ com.eticaret.servlets
  ► AdminLogin.java
  ► AdminLogout.java
  ► KargoEkle.java
  ► KargoSil.java
  ► KategoriEkle.java
  ► KategoriUrunGetir.java
  ► MusteriEkle.java
  ► MusteriLogin.java
  ► MusteriLogout.java
  ► MusteriSil.java
  ► SepeteEkle.java
  ► SepettenCikar.java
  ► SiparisEkle.java
  ► SiparisSil.java
  ► SiparisTamamla.java
  ► UrunEkle.java
  ► UrunResimGetir.java
  ► UrunSil.java
```

AdminLogin.java



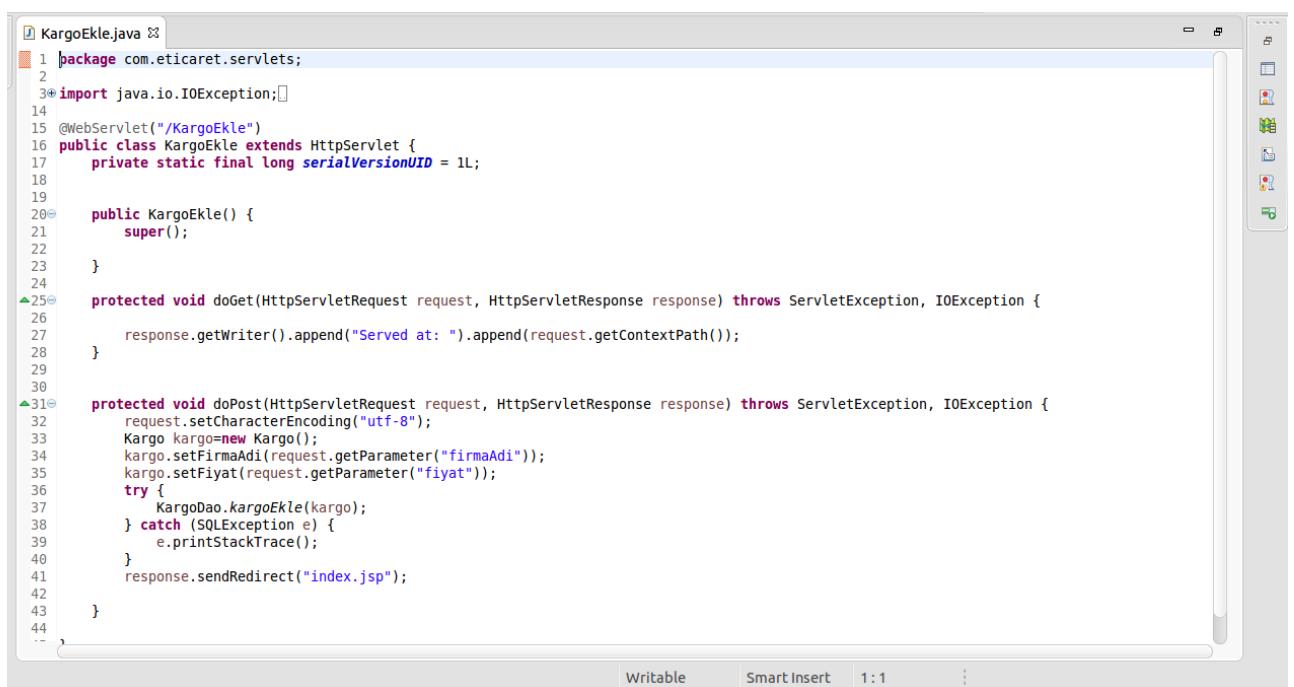
```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/AdminLogin")
6 public class AdminLogin extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
10         doPost(request, response);
11     }
12
13     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
14         String adminMail = request.getParameter("adminMail");
15         String adminSifre = request.getParameter("adminSifre");
16         //Formdan verileri cekip yeni oluşturduğumuz değişkenlere atıyoruz
17
18         if (adminMail != null && adminSifre != null) {
19             //Inputlar bos degilse
20             Admin admin = AdminDao.getAdminGiris(adminMail, adminSifre);
21             if (admin != null) {
22                 //Veri tabanında kayıt varsa
23                 request.getSession().setAttribute("admin", admin); //admin keyine admin valuesini atıyoruz
24                 response.sendRedirect("adminIndex.jsp");
25             } else {
26                 response.sendRedirect("admin.jsp");
27             }
28         } else {
29             response.sendRedirect("index.jsp");
30         }
31     }
32 }
```

AdminLogout.java



```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5
6 @WebServlet("/AdminLogout")
7 public class AdminLogout extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9
10
11     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
12         doPost(request, response);
13     }
14
15
16     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
17         request.getSession().invalidate();
18         response.sendRedirect("admin");
19     }
20
21 }
22
23
24
25
26
```

KargoEkle.java



```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5
6 @WebServlet("/KargoEkle")
7 public class KargoEkle extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9
10
11     public KargoEkle() {
12         super();
13     }
14
15
16     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
17
18         response.getWriter().append("Served at: ").append(request.getContextPath());
19     }
20
21
22
23
24
25     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
26
27         request.setCharacterEncoding("utf-8");
28         Kargo kargo=new Kargo();
29         kargo.setFirmaAdi(request.getParameter("firmaAdi"));
30         kargo.setFiyat(request.getParameter("fiyat"));
31         try {
32             KargoDao.kargoEkle(kargo);
33         } catch (SQLException e) {
34             e.printStackTrace();
35         }
36         response.sendRedirect("index.jsp");
37     }
38
39
40
41
42
43
44 }
```

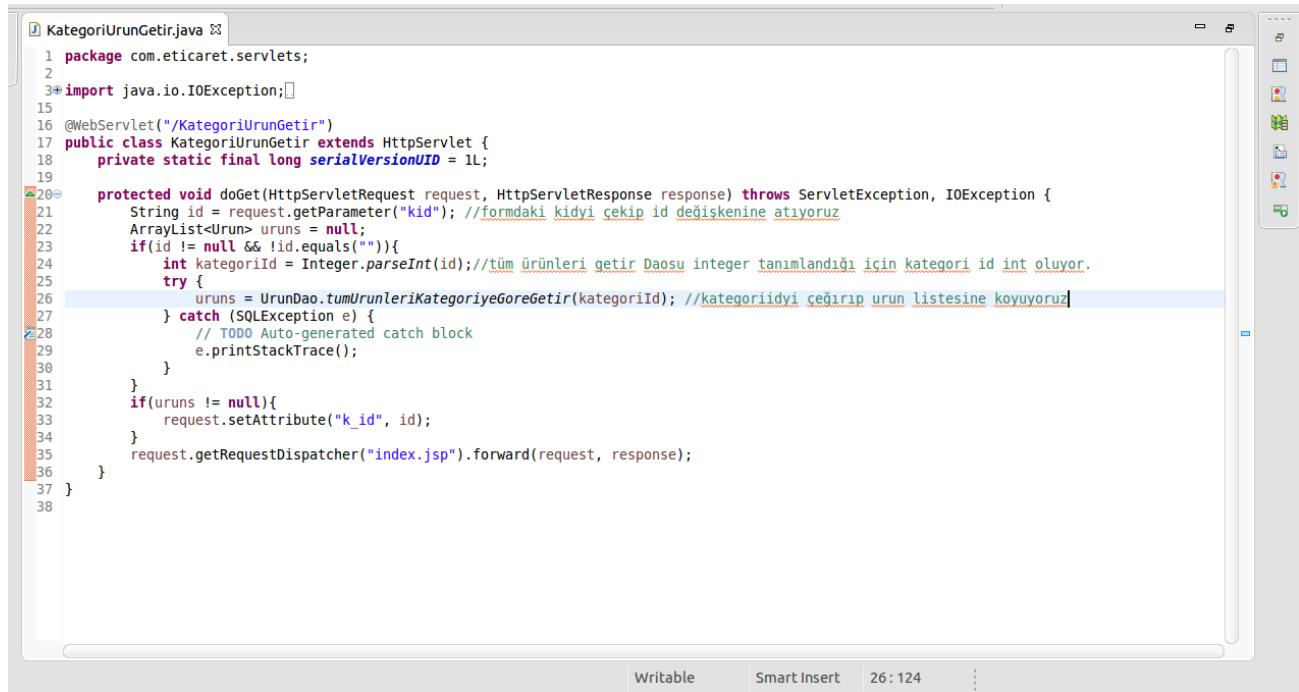
KargoSil.java

```
1 package com.eticaret.servlets;
2
3@import java.io.IOException;
4
5
6 @WebServlet("/KargoSil")
7 public class KargoSil extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9
10
11     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
12         doPost(request, response);
13     }
14
15
16     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
17         request.setCharacterEncoding("UTF-8");
18         response.setCharacterEncoding("UTF-8");
19         Kargo kargo = new Kargo();
20         kargo.setKargoId(Integer.parseInt(request.getParameter("kargoId"))); //formdan namesi kargoId olan veriyi çektiğimizde
21         try {
22             int state = KargoDao.kargoSil(kargo);
23             String message = (state == 1) ? "Kargo şirketi başarıyla silindi" : "İşlem sırasında bir hata oluştu";
24             request.setAttribute("stateSilme", message); //mesaj stringini başka sayfada kullanmak için attr oluşturuyoruz
25             request.getRequestDispatcher("kargo").forward(request, response);
26         } catch (SQLException e) {
27             e.printStackTrace();
28         }
29     }
30
31 }
32
33
34
35
36
37
38
39
40
41 }
```

KategoriEkle.java

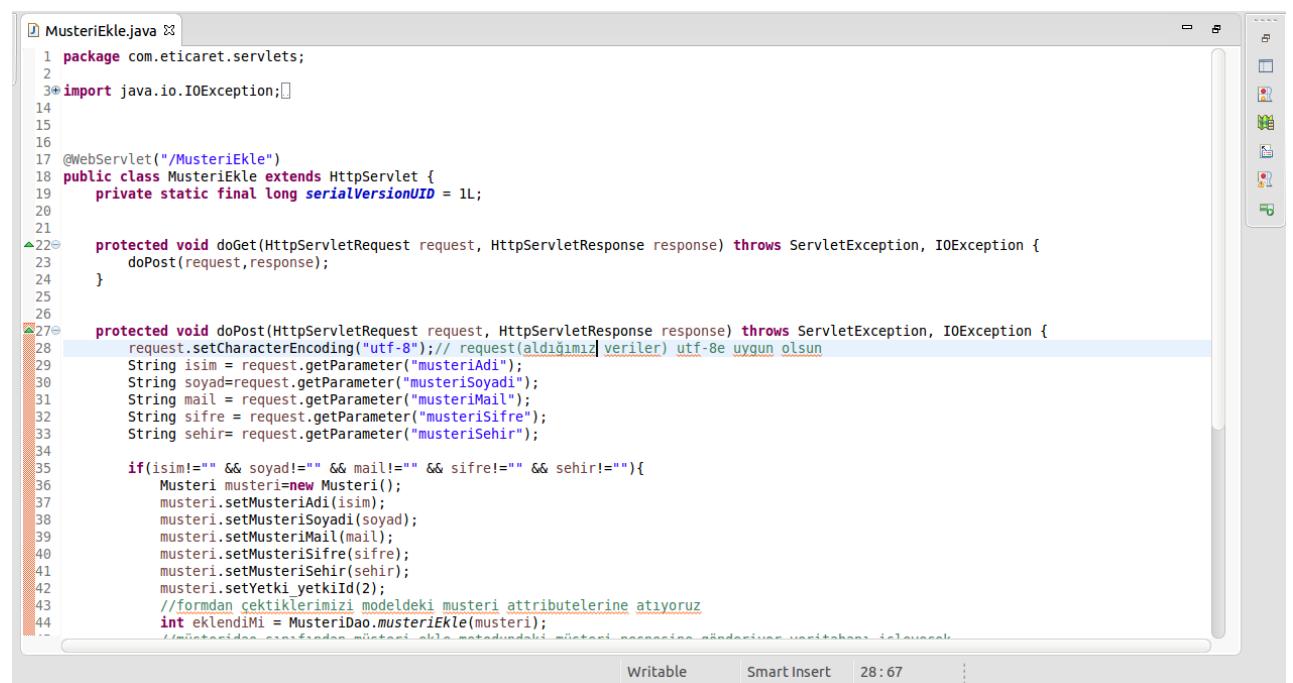
```
1 package com.eticaret.servlets;
2
3@import java.io.IOException;
4
5
6 @WebServlet("/KategoriEkle")
7 public class KategoriEkle extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9
10     public KategoriEkle() {
11         super();
12         // TODO Auto-generated constructor stub
13     }
14
15     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
16         // TODO Auto-generated method stub
17         response.getWriter().append("Served at: ").append(request.getContextPath());
18     }
19
20     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
21         request.setCharacterEncoding("utf-8");
22         Kategori kategori = new Kategori();
23         kategori.setKategoriAdi(request.getParameter("kategoriAdi"));
24         try {
25             KategoriDao.kategoriEkle(kategori);
26         } catch (SQLException e) {
27             e.printStackTrace();
28         }
29         response.sendRedirect("index.jsp");
30     }
31 }
```

KategoriUrunGetir.java

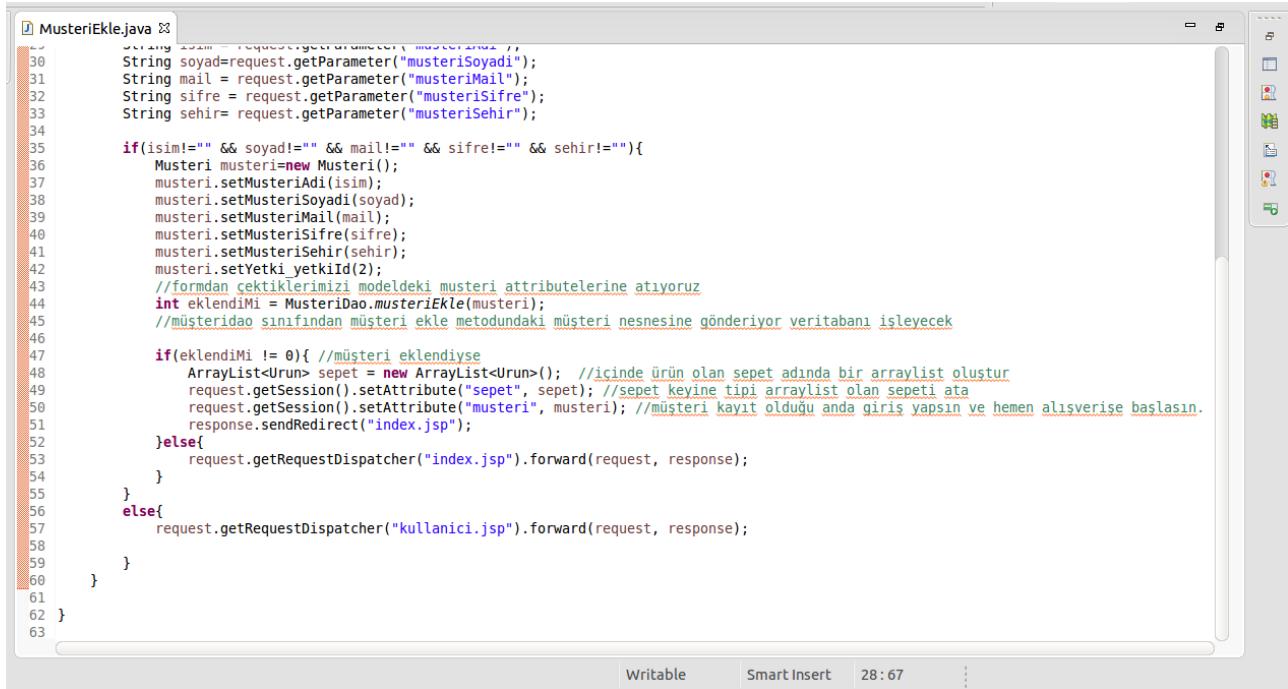


```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/KategoriUrunGetir")
6 public class KategoriUrunGetir extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9
10    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
11        String id = request.getParameter("kid"); //formdaki kidyi cekip id deäiskenine atiyoruz
12        ArrayList<Urun> uruns = null;
13        if(id != null && !id.equals("")){
14            int kategoriId = Integer.parseInt(id); //tüm ürünler getir DaoSu integer tanımladığı için kategori id int oluyor.
15            try {
16                uruns = UrunDao.tumUrunleriKategoriyeGoreGetir(kategoriId); //kategori idyi ceğırip ürün listesine koymuyoruz
17            } catch (SQLException e) {
18                // TODO Auto-generated catch block
19                e.printStackTrace();
20            }
21        }
22        if(uruns != null){
23            request.setAttribute("k_id", id);
24        }
25        request.getRequestDispatcher("index.jsp").forward(request, response);
26    }
27 }
28 }
```

MusteriEkle.java



```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5
6
7 @WebServlet("/MusteriEkle")
8 public class MusteriEkle extends HttpServlet {
9     private static final long serialVersionUID = 1L;
10
11    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
12        doPost(request, response);
13    }
14
15
16
17    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
18        request.setCharacterEncoding("utf-8");// request[alديğimiz] veriler utf-8e uygun olsun
19        String isim = request.getParameter("musteriAdi");
20        String soyad=request.getParameter("musteriSoyadi");
21        String mail = request.getParameter("musteriMail");
22        String sifre = request.getParameter("musteriSifre");
23        String sehir= request.getParameter("musteriSehir");
24
25        if(isim!="" & soyad!="" & mail!="" & sifre!="" & sehir!=""){
26            Musteri musteri=new Musteri();
27            musteri.setMusteriAdi(isim);
28            musteri.setMusteriSoyadi(soyad);
29            musteri.setMusteriMail(mail);
30            musteri.setMusteriSifre(sifre);
31            musteri.setMusteriSehir(sehir);
32            musteri.setYetki_yetkili(2);
33            //formdan çektiğimizi modeldeki musteri attributelerine atiyoruz
34            int eklenildi=MusteriDao.musteriEkle(musteri);
35        }
36    }
37 }
```

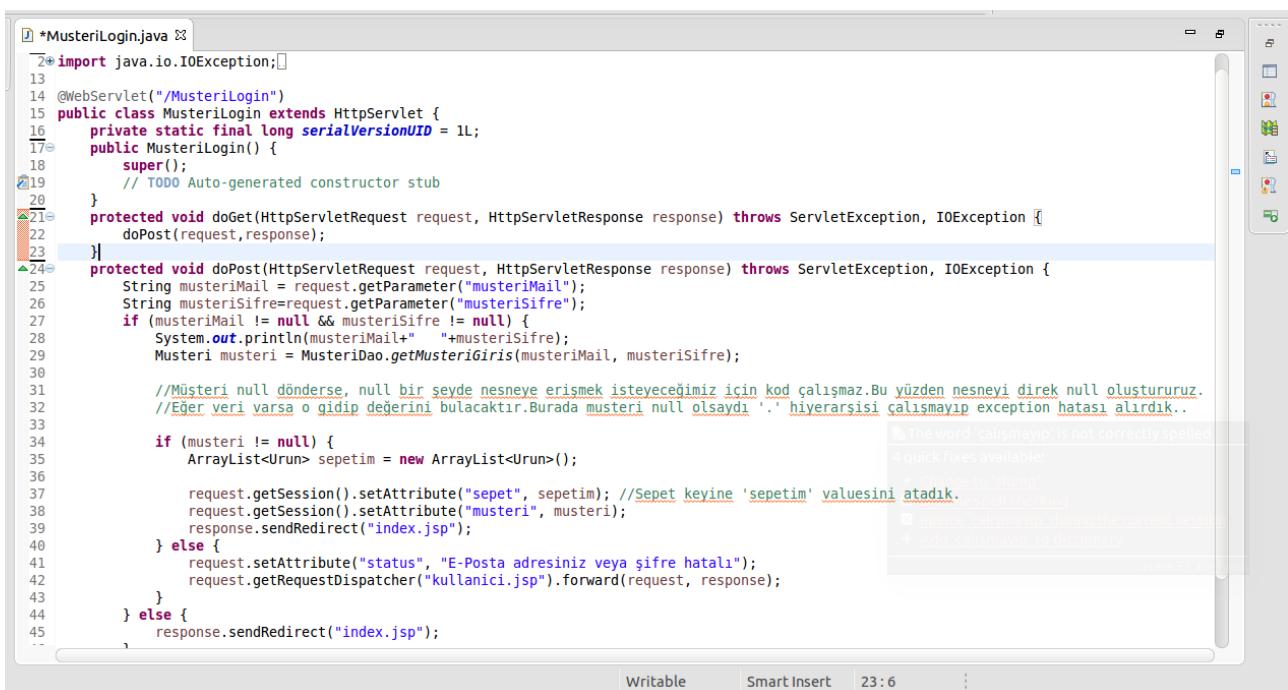


```

1 package com.ankit;
2 import java.io.*;
3 import javax.servlet.*;
4 import javax.servlet.http.*;
5 import com.ankit.model.Musteri;
6 import com.ankit.dao.MusteriDao;
7
8 public class MusteriEkle extends HttpServlet {
9     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
10        String isim = request.getParameter("musteriAdi");
11        String soyad = request.getParameter("musteriSoyadi");
12        String mail = request.getParameter("musteriMail");
13        String sifre = request.getParameter("musteriSifre");
14        String sehir = request.getParameter("musteriSehir");
15
16        if(isim != "" & soyad != "" & mail != "" & sifre != "" & sehir != ""){
17            Musteri musteri = new Musteri();
18            musteri.setMusteriAdi(isim);
19            musteri.setMusteriSoyadi(soyad);
20            musteri.setMusteriMail(mail);
21            musteri.setMusteriSifre(sifre);
22            musteri.setMusteriSehir(sehir);
23            musteri.setYetki_yetkiId(2);
24            //formdan cektilerimizi modeldeki musteri attributelerine atiyoruz
25            int eklendiMi = MusteriDao.musteriEkle(musteri);
26            //musteridao sinifindan musteri ekle metodundaki musteri nesnesine gönderiyor veritabanı isleyecek
27
28            if(eklendiMi != 0){ //musteri eklenirse
29                ArrayList<Urun> sepet = new ArrayList<Urun>(); //icinde ürün olan sepet içinde bir arraylist olustur
30                request.getSession().setAttribute("sepet", sepet); //sepet keyine tipi arraylist olan sepeti ata
31                request.getSession().setAttribute("musteri", musteri); //musteri kayıt olduğu anda giriş yapın ve hemen alışverişe baslasın.
32                response.sendRedirect("index.jsp");
33            }else{
34                request.getRequestDispatcher("index.jsp").forward(request, response);
35            }
36        }else{
37            request.getRequestDispatcher("kullanici.jsp").forward(request, response);
38        }
39    }
40 }

```

MusteriLogin.java



```

1 package com.ankit;
2 import java.io.IOException;
3
4 @WebServlet("/MusteriLogin")
5 public class MusteriLogin extends HttpServlet {
6     private static final long serialVersionUID = 1L;
7     public MusteriLogin() {
8         super();
9         // TODO Auto-generated constructor stub
10    }
11    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
12        doPost(request, response);
13    }
14    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15        String musteriMail = request.getParameter("musteriMail");
16        String musteriSifre = request.getParameter("musteriSifre");
17        if (musteriMail != null & musteriSifre != null) {
18            System.out.println(musteriMail + " " + musteriSifre);
19            Musteri musteri = MusteriDao.getMusteriGiris(musteriMail, musteriSifre);
20
21            //Müşteri null dönerse, null bir seye nesneye erismek isteyeceğimiz için kod çalışmaz.Bu yüzden nesneyi direkt null oluştururuz.
22            //Eğer veri varsa o gidip değerini bulacaktır.Burada musteri null olsaydı '.' hiperarşisi çalışmayıp exception hatası alırdı..
23
24            if (musteri != null) {
25                ArrayList<Urun> sepetim = new ArrayList<Urun>();
26
27                request.getSession().setAttribute("sepet", sepetim); //Sepet keyine 'sepetim' valuesini atadık.
28                request.getSession().setAttribute("musteri", musteri);
29                response.sendRedirect("index.jsp");
30            } else {
31                request.setAttribute("status", "E-Posta adresiniz veya şifre hatalı");
32                request.getRequestDispatcher("kullanici.jsp").forward(request, response);
33            }
34        } else {
35            response.sendRedirect("index.jsp");
36        }
37    }
38 }

```

MusteriLogout.java

The screenshot shows a Java code editor window with the file 'MusteriLogout.java' open. The code defines a servlet for user logout. It includes imports for IOException and Musteri, and annotations for WebServlet and HttpServlet. The class contains doGet and doPost methods that invalidate the session and redirect to 'index.jsp'. The code is color-coded with syntax highlighting.

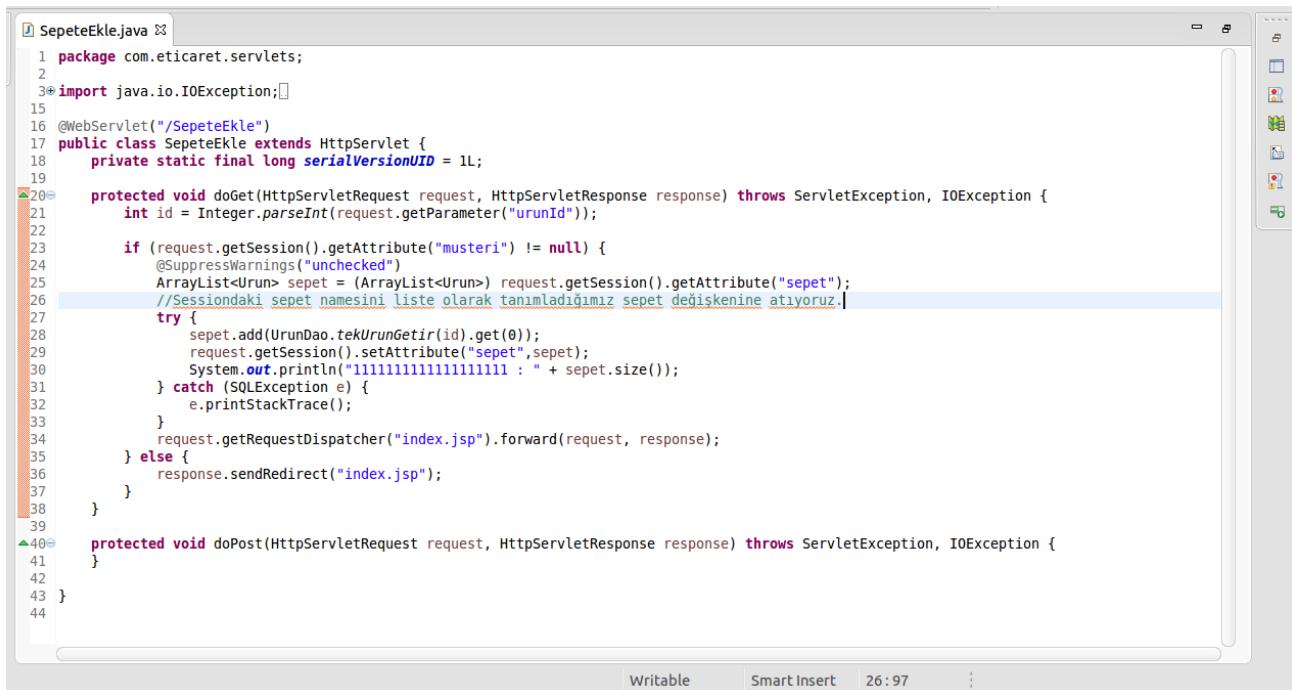
```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 //import com.eticaret.model.Musteri;
6
7 @WebServlet("/MusteriLogout")
8 public class MusteriLogout extends HttpServlet {
9     private static final long serialVersionUID = 1L;
10
11
12
13
14
15
16
17
18     public MusteriLogout() {
19         super();
20         // TODO Auto-generated constructor stub
21     }
22
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         doPost(request, response);
25     }
26
27
28     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29         request.getSession().invalidate();
30         response.sendRedirect("index.jsp");
31     }
32
33 }
```

MusteriSil.java

The screenshot shows a Java code editor window with the file 'MusteriSil.java' open. The code defines a servlet for deleting a user. It includes imports for IOException and MusteriDao, and annotations for WebServlet and HttpServlet. The class contains doGet and doPost methods. The doPost method retrieves the user ID from the request parameter 'sil', calls the MusteriDao's musteriSil method, and then redirects to 'admin.jsp'. It also handles SQL exceptions and prints the stack trace. The code is color-coded with syntax highlighting.

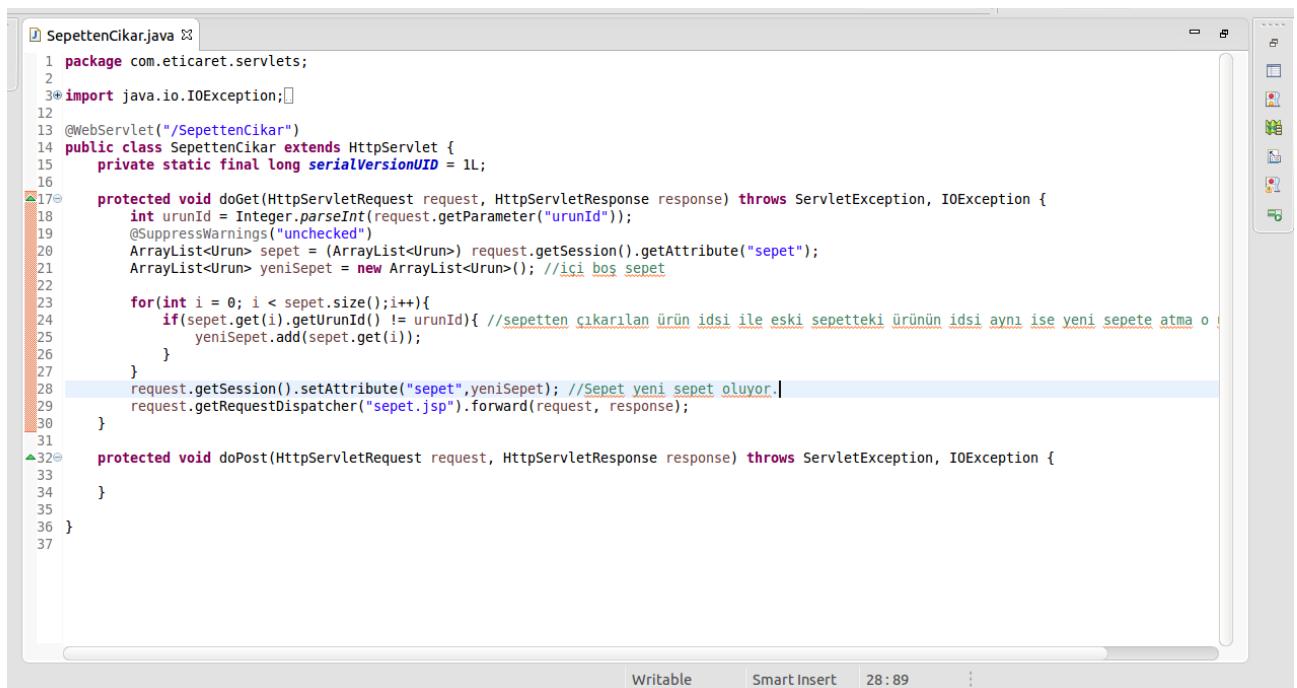
```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/MusteriSil")
6 public class MusteriSil extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9
10
11
12
13
14
15
16     public MusteriSil() {
17         super();
18         // TODO Auto-generated constructor stub
19     }
20
21
22
23
24
25
26     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27         doPost(request, response);
28     }
29
30
31     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32
33         int id = Integer.parseInt(request.getParameter("sil"));
34         try {
35             MusteriDao.musteriSil(id);
36             response.sendRedirect("admin.jsp");
37         } catch (SQLException e) {
38             // TODO Auto-generated catch block
39             e.printStackTrace();
40         }
41     }
42
43 }
```

SepeteEkle.java



```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/SepeteEkle")
6 public class SepeteEkle extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
10         int id = Integer.parseInt(request.getParameter("urunId"));
11
12         if (request.getSession().getAttribute("musteri") != null) {
13             @SuppressWarnings("unchecked")
14             ArrayList<Urun> sepet = (ArrayList<Urun>) request.getSession().getAttribute("sepet");
15             //Sessiondaki sepet namesini liste olarak tanımladığımız sepet değişkenine atıyoruz.
16             try {
17                 sepet.add(UrunDao.tekUrutGetir(id).get(0));
18                 request.getSession().setAttribute("sepet", sepet);
19                 System.out.println("111111111111111111 : " + sepet.size());
20             } catch (SQLException e) {
21                 e.printStackTrace();
22             }
23             request.getRequestDispatcher("index.jsp").forward(request, response);
24         } else {
25             response.sendRedirect("index.jsp");
26         }
27     }
28
29     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
30     }
31
32 }
33
34 }
```

SepettenCikar.java



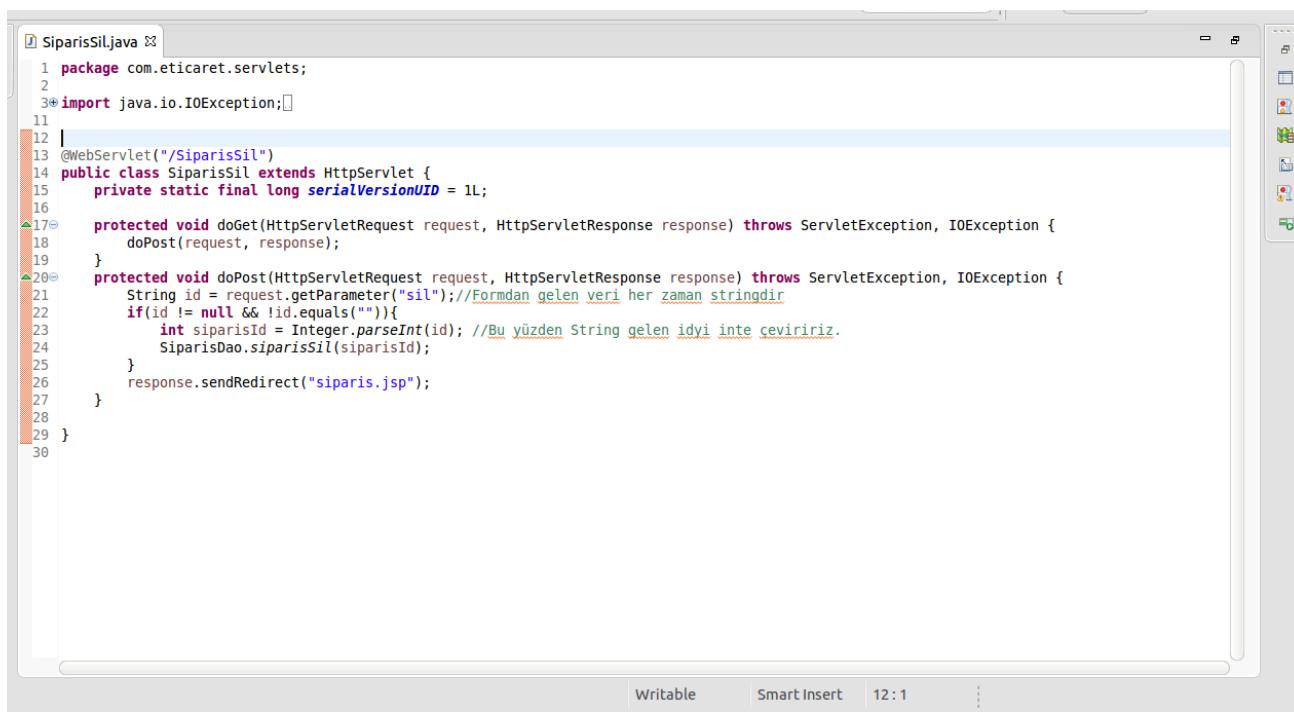
```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/Sepettencikar")
6 public class SepettenCikar extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
10         int urunId = Integer.parseInt(request.getParameter("urunId"));
11         @SuppressWarnings("unchecked")
12         ArrayList<Urun> sepet = (ArrayList<Urun>) request.getSession().getAttribute("sepet");
13         ArrayList<Urun> yeniSepet = new ArrayList<Urun>(); //İci boş sepet
14
15         for(int i = 0; i < sepet.size();i++){
16             if(sepet.get(i).getUrunId() != urunId){ //sepetteki ürünlerin id'si ile eski sepetteki ürünlerin id'si aynı ise yeni sepete atma olsun.
17                 yeniSepet.add(sepet.get(i));
18             }
19         }
20         request.getSession().setAttribute("sepet",yeniSepet); //Sepet yeni sepet oluyor.
21         request.getRequestDispatcher("sepet.jsp").forward(request, response);
22     }
23
24     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
25     }
26
27 }
28
29 }
```

SiparisEkle.java



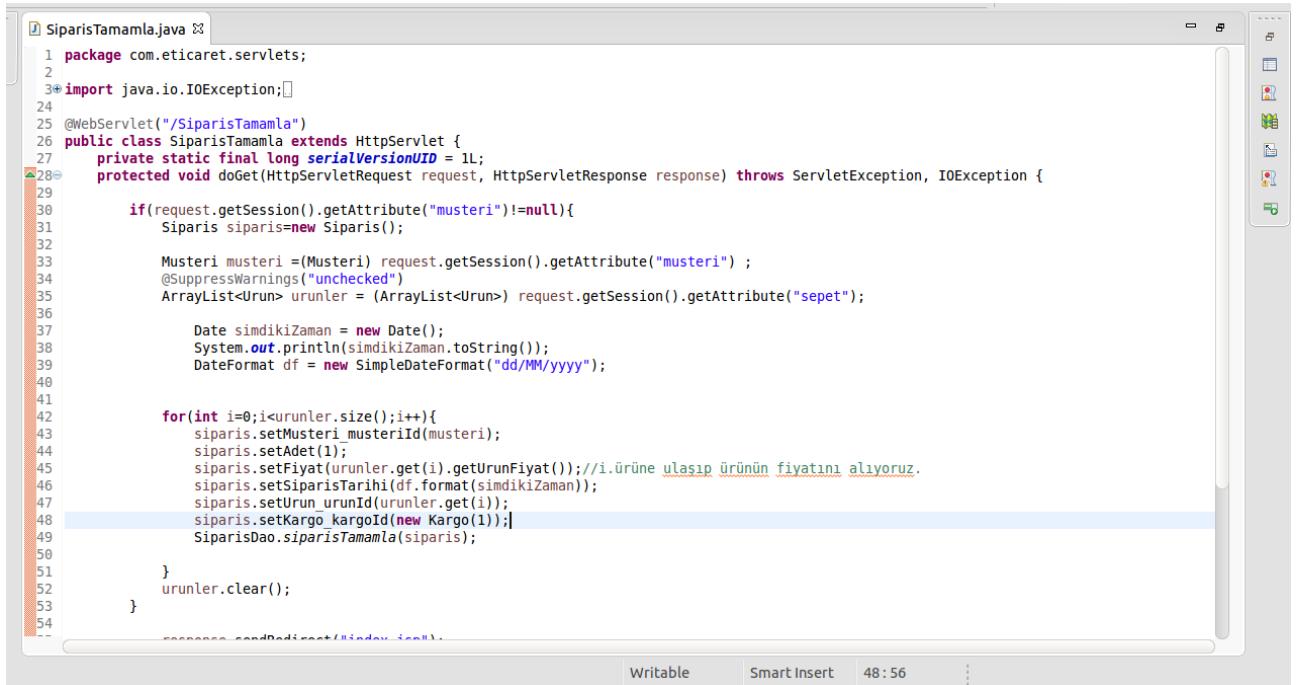
```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5
6 @WebServlet("/SiparisEkle")
7 public class SiparisEkle extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9
10
11     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
12         doPost(request, response);
13     }
14
15
16     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
17         //TODO
18     }
19
20
21 }
```

SiparisSil.java



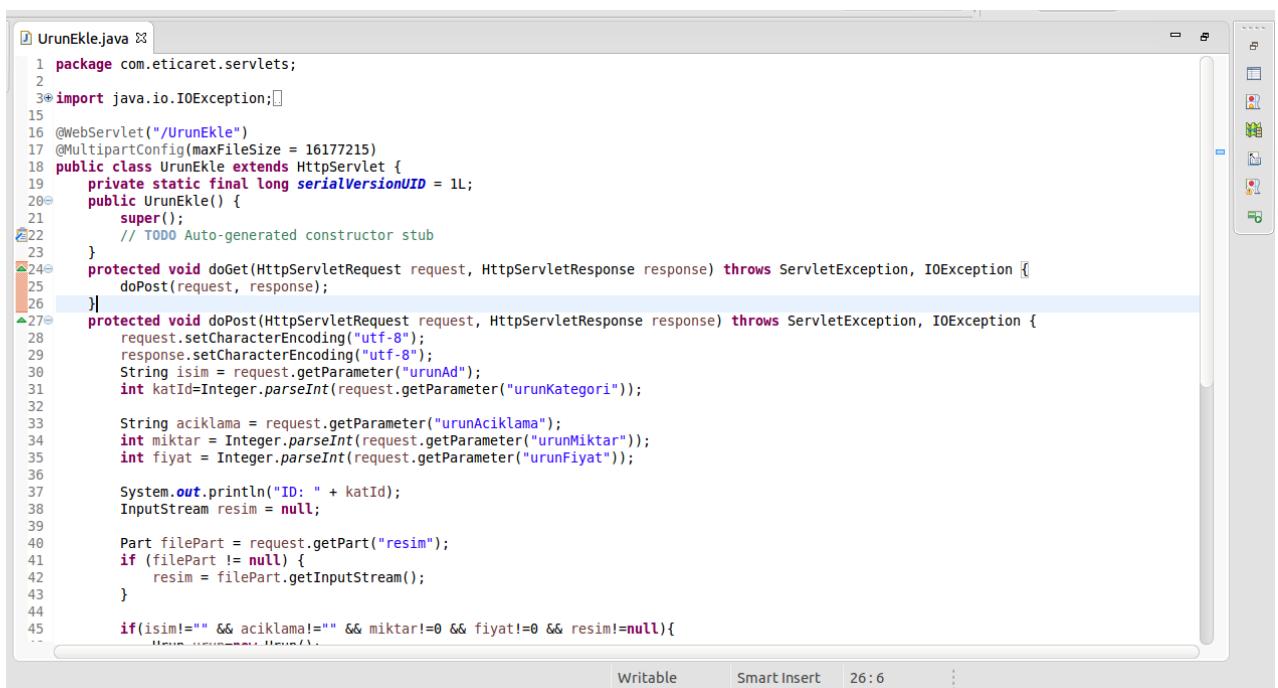
```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5
6 @WebServlet("/SiparisSil")
7 public class SiparisSil extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9
10
11     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
12         doPost(request, response);
13     }
14
15     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
16         String id = request.getParameter("sil");//Formdan gelen veri her zaman stringdir
17         if(id != null && id.equals("")){
18             int siparisId = Integer.parseInt(id); //Bu yüzden String gelen idyi inte ceviririz.
19             SiparisDao.siparisSil(siparisId);
20         }
21         response.sendRedirect("siparis.jsp");
22     }
23
24 }
```

SiparisTamamla.java



```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/SiparisTamamla")
6 public class SiparisTamamla extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
9
10         if(request.getSession().getAttribute("musteri")!=null){
11             Siparis siparis=new Siparis();
12
13             Musteri musteri =(Musteri) request.getSession().getAttribute("musteri");
14             @SuppressWarnings("unchecked")
15             ArrayList<Urun> urunler = (ArrayList<Urun>) request.getSession().getAttribute("sepet");
16
17             Date simdikiZaman = new Date();
18             System.out.println(simdikiZaman.toString());
19             DateFormat df = new SimpleDateFormat("dd/MM/yyyy");
20
21             for(int i=0;i<urunler.size();i++){
22                 siparis.setMusteri_musteriId(musteri);
23                 siparis.setAdet(1);
24                 siparis.setFiyat(urunler.get(i).getUrutFiyat()); //i.ürüne ulaşıp ürünün fiyatını alıyoruz.
25                 siparis.setSiparisTarihi(df.format(simdikiZaman));
26                 siparis.setUrut_urunId(urunler.get(i));
27                 siparis.setKargo_kargoId(new Kargo(1));
28                 SiparisDao.siparisTamamla(siparis);
29
30             }
31             urunler.clear();
32         }
33     }
34 }
```

UrunEkle.java



```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/UrunEkle")
6 @MultipartConfig(maxFileSize = 16177215)
7 public class UrunEkle extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9     public UrunEkle() {
10         super();
11         // TODO Auto-generated constructor stub
12     }
13     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
14         doPost(request, response);
15     }
16     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
17         request.setCharacterEncoding("utf-8");
18         response.setCharacterEncoding("utf-8");
19         String isim = request.getParameter("urunAd");
20         int katId=Integer.parseInt(request.getParameter("urunKategori"));
21
22         String aciklama = request.getParameter("urunAciklama");
23         int miktar = Integer.parseInt(request.getParameter("urunMiktar"));
24         int fiyat = Integer.parseInt(request.getParameter("urunFiyat"));
25
26         System.out.println("ID: " + katId);
27         InputStream resim = null;
28
29         Part filePart = request.getPart("resim");
30         if (filePart != null) {
31             resim = filePart.getInputStream();
32         }
33
34         if(isim!="" && aciklama!="" && miktar!=0 && fiyat!=0 && resim!=null){
35             // Process the uploaded file
36         }
37     }
38 }
```

The screenshot shows a Java code editor window with the file 'UrunEkle.java' open. The code handles the creation of a new product ('Urunk'). It checks if a file part named 'resim' is present in the request. If so, it creates a new 'Urunk' object, sets its attributes (product name, quantity, price, and image), and saves it to the database using a DAO. If successful, it sends a redirect to 'urun.jsp?urunId=' followed by the product's ID. If there's an error or if no file part is provided, it sends a redirect to 'urunIslemleri.jsp'. The code uses try-catch blocks and includes imports for HttpServletRequest, HttpServletResponse, and IOException.

```
39
40     Part filePart = request.getPart("resim");
41     if (filePart != null) {
42         resim = filePart.getInputStream();
43     }
44
45     if(isim!="" && aciklama!="" && miktar!=0 && fiyat!=0 && resim!=null){
46         Urunk urun=new Urunk();
47         urun.setUrunkAd(isim);
48         urun.setUrunkAciklama(aciklama);
49         urun.setUrunkMiktar(miktar);
50         urun.setUrunkFiyat(fiyat);
51         urun.setKategori_kategoriId(katId);
52         urun.setUrunkResim(resim);
53         int eklenidiMi = UrunkDao.urunkEkle(urun);
54         if(eklenidiMi != 0){
55             int id=0;
56             try{
57                 int size = UrunkDao.tumUrunkleriGetir().size();
58                 id = UrunkDao.tumUrunkleriGetir().get(size-1).getUrunkId();
59                 response.sendRedirect("urun.jsp?urunId=" + id);
60             }catch(Exception err){
61             }
62         }else{
63             response.sendRedirect("urunIslemleri.jsp");
64         }
65     }
66     else{
67         response.sendRedirect("urunIslemleri.jsp");
68     }
69 }
70 }
71 }
72 }
```

UrunSil.java

The screenshot shows a Java code editor window with the file 'UrunSil.java' open. This is a servlet for deleting a product ('UrunSil'). It first checks if a parameter 'sil' is present in the request. If found, it prints the ID to the console, converts it to an integer, and calls a DAO method to delete the product. If an SQL exception occurs during deletion, it prints the error message and stack trace. Finally, it sends a redirect to 'urunIslemleri.jsp'. The code also includes a doPost method which currently does nothing. It uses imports for HttpServlet, HttpServletRequest, HttpServletResponse, ServletException, IOException, and PrintWriter.

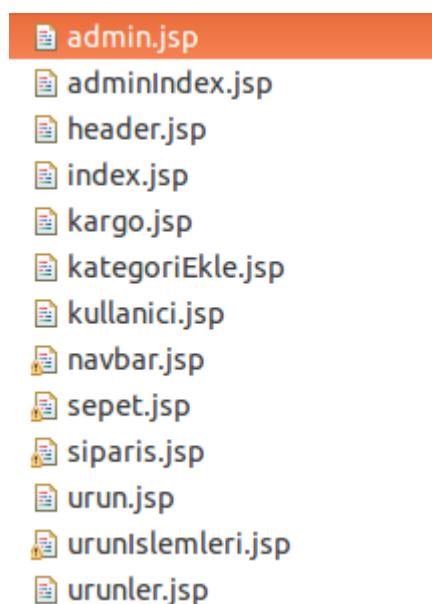
```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/UrunSil")
6 public class UrunSil extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
10         request.setCharacterEncoding("UTF-8");
11         response.setCharacterEncoding("UTF-8");
12
13         String id = request.getParameter("sil");
14         System.out.println("ID: " + id);
15         if(id != null && !id.equals("")){
16             try {
17                 int urunkId = Integer.parseInt(id);
18                 UrunkDao.urunkSil(urunkId);
19             } catch (SQLException e) {
20                 System.out.println("error " + e);
21                 e.printStackTrace();
22             }
23         }
24         response.sendRedirect("urunIslemleri.jsp");
25     }
26
27     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28     }
29 }
```

UrunResimGetir.java

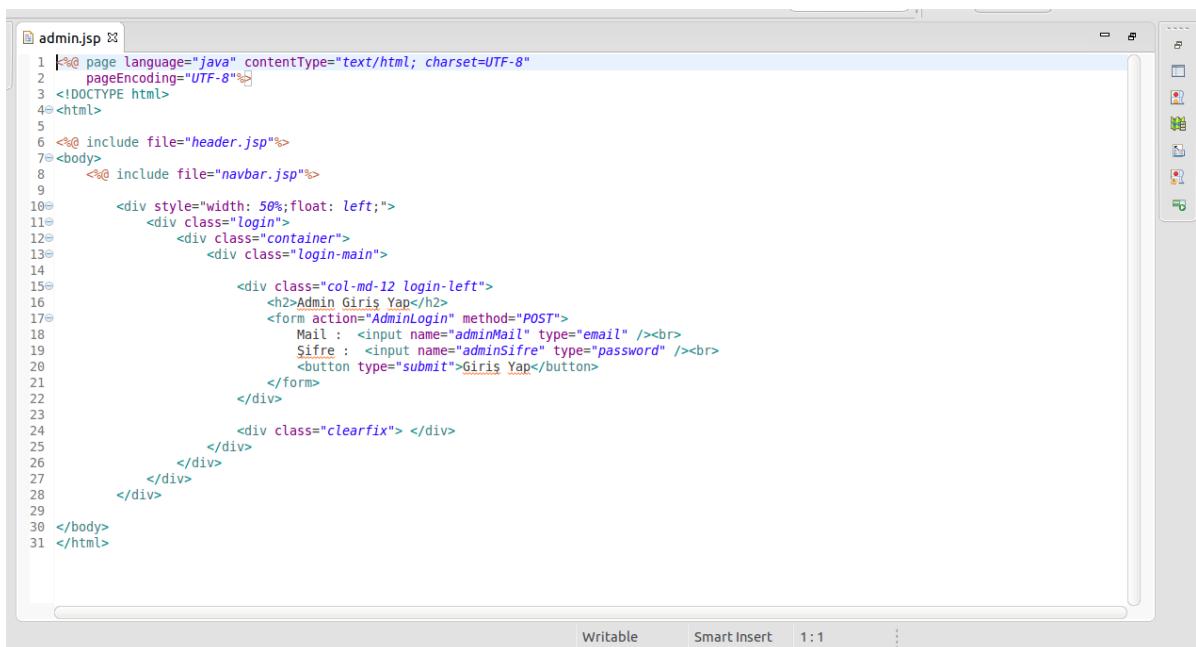
The screenshot shows a Java code editor window with the file 'UrunResimGetir.java' open. The code implements a HttpServlet for handling image requests. It includes imports for package com.eticaret.servlets, java.io.IOException, and javax.servlet.annotation.WebServlet. The class UrunResimGetir extends HttpServlet and overrides doGet and doPost methods. The doPost method retrieves an image by its ID from a database, sets the response content type to 'image/jpeg', and writes the image bytes to the response output stream. A comment in the code explains the conversion of the entire image to byte format and its writing to the response.

```
1 package com.eticaret.servlets;
2
3 import java.io.IOException;
4
5 @WebServlet("/UrunResimGetir")
6 public class UrunResimGetir extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9
10    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
11        doPost(request, response);
12    }
13
14    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15        int resimId = Integer.parseInt(request.getParameter("urunResim"));
16        response.setContentType("image/jpeg");
17        byte[] image = UrunDao.getImageByUrunId(resimId); //Bütün eklenen resimler byte koda dönüştürülür
18        response.setContentLength(image.length); //Resimin boyutunu döndürüyoruz.
19        response.getOutputStream().write(image); //imangenin içine btyeyi yazıyoruz. Resim veri tabanına byte byte işleniyor.
20    }
21
22}
23
24
25
26
27
28}
29
30}
31}
```

VİEW- Jsp Sınıfları:

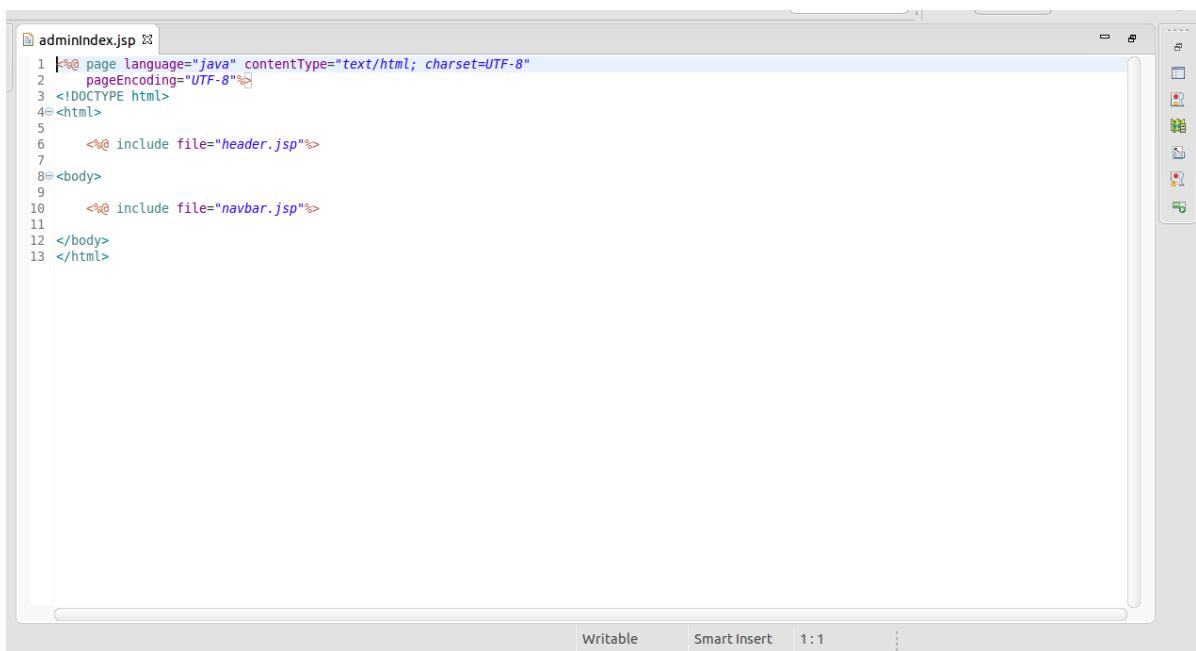


Admin.jsp



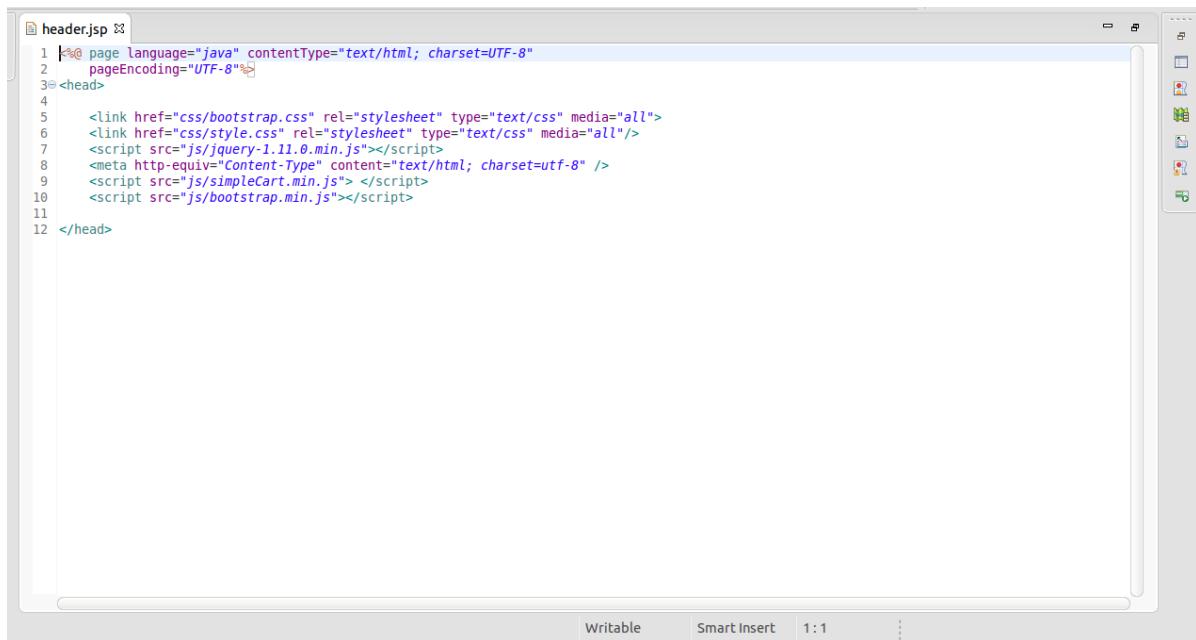
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5
6 <%@ include file="header.jsp"%>
7 <body>
8     <%@ include file="navbar.jsp"%>
9
10    <div style="width: 50%;float: left;">
11        <div class="login">
12            <div class="container">
13                <div class="login-main">
14
15                    <div class="col-md-12 login-left">
16                        <h2>Admin Giriş Yap</h2>
17                        <form action="AdminLogin" method="POST">
18                            Mail : <input name="adminMail" type="email" /><br>
19                            Sifre : <input name="adminSifre" type="password" /><br>
20                            <button type="submit">Giris Yap</button>
21
22                    </form>
23
24                    <div class="clearfix"> </div>
25
26                </div>
27            </div>
28        </div>
29
30    </body>
31 </html>
```

AdminIndex.jsp



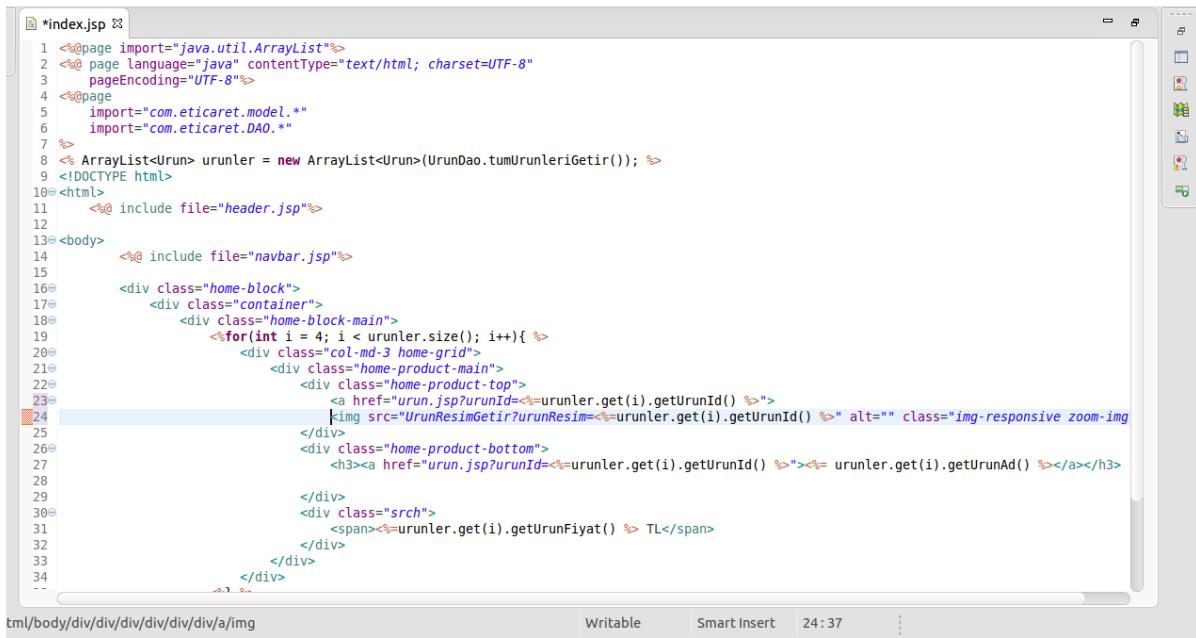
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5
6     <%@ include file="header.jsp"%>
7
8 <body>
9     <%@ include file="navbar.jsp"%>
10
11 </body>
12 </html>
```

Header.jsp



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3<head>
4
5   <link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all">
6   <link href="css/style.css" rel="stylesheet" type="text/css" media="all"/>
7   <script src="js/jquery-1.11.0.min.js"></script>
8   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
9   <script src="js/simpleCart.min.js"></script>
10  <script src="js/bootstrap.min.js"></script>
11
12 </head>
```

Index.jsp



```
1 <%@page import="java.util.ArrayList"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3   pageEncoding="UTF-8"%>
4 <%@page
5   import="com.eticaret.model.*"
6   import="com.eticaret.DAO.*"
7 %>
8 <% ArrayList<Urun> urunler = new ArrayList<Urun>(UrunDao.tumUrunleriGetir()); %>
9 <!DOCTYPE html>
10<html>
11   <%@ include file="header.jsp"%>
12
13<body>
14   <%@ include file="navbar.jsp"%>
15
16   <div class="home-block">
17     <div class="container">
18       <div class="home-block-main">
19         <%for(int i = 4; i < urunler.size(); i++){ %>
20           <div class="col-md-3 home-grid">
21             <div class="home-product-main">
22               <div class="home-product-top">
23                 <a href="urun.jsp?urunId=<%=urunler.get(i).getUrunId() %>">
24                   
26               <div class="home-product-bottom">
27                 <h3><a href="urun.jsp?urunId=<%=urunler.get(i).getUrunId() %>><%= urunler.get(i).getUrunAd() %></a></h3>
28
29               </div>
30             <div class="srch">
31               <span><%=urunler.get(i).getUrunFiyat() %> TL</span>
32             </div>
33           </div>
34         </div>
35       . . .
36     </div>
37   </div>
38
39   <div class="row">
40     <div class="col-md-12 text-center">
41       
42       <p>Copyright © 2015. All Rights Reserved. E-Ticaret Platformu</p>
43     </div>
44   </div>
45
46 </body>
47
48 </html>
```

KategoriEkle.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4<html>
5 <%@ include file="header.jsp"%>
6<body>
7     <%@ include file="navbar.jsp"%>
8
9     <div style="width: 50%; float: left;">
10        <div class="login">
11            <div class="container">
12                <div class="login-main">
13                    <div class="col-md-12 login-left">
14                        <h2>Kategori Ekle</h2>
15                        <form action="KategoriEkle" method="POST">
16                            Kategori Adı : <input name="kategoriAdi" type="text" /><br>
17                            <button type="submit">Kategori Ekle</button>
18                        </form>
19                    </div>
20                    <div class="clearfix"> </div>
21                </div>
22            </div>
23        </div>
24    </div>
25
26 </body>
27 </html>
```

Kullanici.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4<html>
5 <%@ include file="header.jsp"%>
6<body>
7     <%@ include file="navbar.jsp"%>
8
9     <div style="width: 50%; float: left;">
10        <div class="login">
11            <div class="container">
12                <div class="login-main">
13                    <div class="col-md-12 login-left">
14                        <h2>Kayıt Ol</h2>
15                        <form action="MusteriEkle" method="POST">
16                            Adı : <input name="musteriAdi" type="text" /><br>
17                            Soyadı : <input name="musteriSoyadi" type="text" /><br>
18                            Mail : <input name="musteriMail" type="email" /><br>
19                            Şifre : <input name="musteriSifre" type="password" /><br>
20                            Şehir : <input name="musteriSehir" type="text" /><br>
21                            <button type="submit">Kayıt Ol</button>
22                        </form>
23                    </div>
24                    <div class="clearfix"> </div>
25                </div>
26            </div>
27        </div>
28    </div>
29
30    <div style="width: 50%; float: left;">
31        <div class="login">
32            <div class="container">
33                <div class="login-main">
34                    <div class="col-md-12 login-left">
```

kullanici.jsp

```

16@    <form action="MusteriEkle" method="POST">
17@        Adı : <input name="musteriAdi" type="text" /><br>
18@        Soyadı : <input name="musteriSoyadi" type="text" /><br>
19@        Mail : <input name="musteriMail" type="email" /><br>
20@        Şifre : <input name="musteriSifre" type="password" /><br>
21@        Şehir : <input name="musteriSehir" type="text" /><br>
22@        <button type="submit">Kayıt Ol</button>
23@    </form>
24@    </div>
25@    <div class="clearfix"> </div>
26@    </div>
27@    </div>
28@  </div>
29@>
30@  <div style="width: 50%;float: left;">
31@    <div class="login">
32@      <div class="container">
33@        <div class="login-main">
34@          <div class="col-md-12 login-left">
35@            <h2>Giriş Yap</h2>
36@            <form action="MusteriLogin" method="POST">
37@                Mail : <input name="musteriMail" type="email" /><br>
38@                Şifre : <input name="musteriSifre" type="password" /><br>
39@                <button type="submit">Giriş Yap</button>
40@            </form>
41@            </div>
42@            <div class="clearfix"> </div>
43@        </div>
44@      </div>
45@    </div>
46@  </div>
47@>
48</body>
49</html>

```

p:directive.page Writable Smart Insert 1:1

Navbar.jsp

navbar.jsp

```

1  <%@page import="java.util.ArrayList"%>
2  <%@ page language="java" contentType="text/html; charset=UTF-8"%
3  %>
4  <%@page
5  import="com.eticaret.model.*"
6  import="com.eticaret.DAO.*"
7  %>
8  <% ArrayList<Kategori> kategoriler_navbar = new ArrayList<Kategori>(KategoriDao.butunKategoriler()); %>
9
10@   <nav class="navbar navbar-default" role="navigation">
11@     <div class="navbar-header">
12@       <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
13@         <span class="sr-only">Toggle navigation</span>
14@         <span class="icon-bar"></span>
15@         <span class="icon-bar"></span>
16@         <span class="icon-bar"></span>
17@       </button>
18@       <div class="navbar-brand logo">
19@         <a href="index.html"></a>
20@       </div>
21@     </div>
22@     <!--.navbar-header-->
23@     <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
24@       <ul class="nav navbar-nav">
25@         <li><a href="index.jsp">Ana Sayfa</a></li>
26@         <%for(int i = 0; i < kategoriler_navbar.size(); i++){ %>
27@           <li class="dropdown">
28@             <a href="urunler.jsp?kategoriId=<%= kategoriler_navbar.get(i).getKategoriId() %>" class="dropdown-toggle"><%= kategori
29@             </li>
30@           <%}>
31@           <%if(session.getAttribute("musteri") == null && session.getAttribute("admin") == null){ %>
32@             <li class="dropdown">
33@               <a href="kullanici.jsp" class="dropdown-toggle">Giriş & Kayıt</a>
34@             </li>
35@           <%}>

```

Writable Smart Insert 1:1

```

24<ul class="nav navbar-nav">
25    <li><a href="index.jsp">Ana Sayfa</a></li>
26    <%for(int i = 0; i < kategoriler-navbar.size(); i++) { %>
27        <li class="dropdown">
28            <a href="urunler.jsp?kategoriId=<=% kategoriler-navbar.get(i).getKategoriId() %>" class="dropdown-toggle"><=% kategori
29            </a>
30        <%}>
31        <%if(session.getAttribute("musteri") == null && session.getAttribute("admin") == null){ %>
32            <li class="dropdown">
33                <a href="kullanici.jsp" class="dropdown-toggle">Giris & Kayit</a>
34            </li>
35        <%}>
36        <%if(session.getAttribute("musteri") != null || session.getAttribute("admin") != null){ %>
37            <li class="dropdown">
38                <a href="sepet.jsp" class="dropdown-toggle">Sepet</a>
39            </li>
40            <li class="dropdown">
41                <a href="MusteriLogout" class="dropdown-toggle">Oturumu Kapat</a>
42            </li>
43        <%}>
44        <%if(session.getAttribute("admin") != null){ %>
45            <li class="dropdown">
46                <a href="admin.jsp" class="dropdown-toggle">Yönetim Paneli</a>
47            </li>
48            <li class="dropdown">
49                <a href="urunIslemleri.jsp" class="dropdown-toggle">Ürün İşlemleri</a>
50            </li>
51            <li class="dropdown">
52                <a href="kategoriEkle.jsp" class="dropdown-toggle"> Kategori İşlemleri</a>
53            </li>
54        <%}>
55    </ul>
56</div>
57</nav>

```

Sepet.jsp

```

1 <%@page import="java.util.ArrayList"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
3 <%@page
4     import="com.eticaret.model.*"
5     import="com.eticaret.DAO.*"
6 %>
7 <% ArrayList<Urun> sepet = new ArrayList<Urun> ((ArrayList<Urun>) request.getSession().getAttribute("sepet")); %>
8 <!DOCTYPE html>
9 <html>
10 <body>
11     <%@ include file="header.jsp"%>
12     <%@ include file="navbar.jsp"%>
13     <div class="ccheckout">
14         <div class="container">
15             <div class="ccheckout-top">
16                 <div class="cart-items heading">
17                     <h1>Sepetim</h1>
18                     <div class="in-check">
19                         <ul class="unit">
20                             <li><span>Ürün</span></li>
21                             <li><span>Ürün Adı</span></li>
22                             <li><span>Fiyat</span></li>
23                         <div class="clearfix"> </div>
24                     </ul>
25                     <%for(int i = 0; i < sepet.size(); i++){ %>
26                         <ul class="cart-header simpleCart_shelfItem">
27                             <li class="ring-in">
28                             <li><span><=% sepet.get(i).getUrundai() %></span></li>
29                             <li><span class="item_price"><=% sepet.get(i).getUrunFiyat() %></span></li>
30                             <li> <a href="SepetCikar?urunId=<=% sepet.get(i).getUrundai() %>" class="add-cart cart-check item_add">Sepette</a>
31                             <div class="clearfix"> </div>
32                         </ul>
33                     <%}>
34                     <a href="SiparisTamamla"><button class="btn btn-warning">Satın Al</button></a>

```

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%@ include file="header.jsp"%>
<%@ include file="navbar.jsp"%>
<div class="checkout">
    <div class="container">
        <div class="checkout-top">
            <div class="cart-items heading">
                <h1>Sepetim</h1>
                <div class="in-check">
                    <ul class="unit">
                        <li><span>Ürün</span></li>
                        <li><span>Ürün Adı</span></li>
                        <li><span>Fiyat</span></li>
                    </ul>
                    <div class="clearfix"> </div>
                </div>
                <ul class="cart-header simpleCart_shelfItem">
                    <li class="ring-in"><span><%= sepet.get(i).getUrunAd() %></span></li>
                    <li><span class="item_price"><%=sepet.get(i).getUrunFiyat() %></span></li>
                    <li ><a href="SepettenCikar?urunId=<%=sepet.get(i).getUrunId() %>" class="add-cart cart-check item_add">Sepetten Çıkar</a></li>
                </ul>
            </div>
            <a href="SiparisTamamla"><button class="btn btn-warning">Satın Al</button></a>
        </div>
    </div>
</div>
</div>
</body>
</html>

```

Siparis.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@include file="header.jsp"%>
<%@include file="navbar.jsp"%>
<div class="checkout">
    <div class="container">
        <div class="checkout-top">
            <div class="cart-items heading">
                <div class="in-check">
                    <ul class="unit">
                        <li><span>Adı - Soyadı</span></li>
                        <li><span>Ürün Adı</span></li>
                        <li><span>Adet</span></li>
                        <li><span>Fiyat</span></li>
                        <li><span>Siparis Tarihi</span></li>
                    </ul>
                    <div class="clearfix"> </div>
                </div>
                <ul class="cart-header simpleCart_shelfItem">
                    <li><${siparis.getMusteri_musteriId().getMusteriAdi()}- ${siparis.getMusteri_musteriId().getMusteriSoyadi()}></li>
                    <li><${siparis.getUrun_urunId().getUrunAd()}></li>
                    <li><${siparis.getAdet()}></li>
                    <li><${siparis.getFiyat()}></li>
                    <li><${siparis.getSiparisTarihi()}></li>
                </ul>
                <div class="clearfix"> </div>
            </div>
            <c:forEach items="${siparisler}" var="siparis">
                <ul class="cart-header simpleCart_shelfItem">
                    <li><${siparis.getMusteri_musteriId().getMusteriAdi()}- ${siparis.getMusteri_musteriId().getMusteriSoyadi()}></li>
                    <li><${siparis.getUrun_urunId().getUrunAd()}></li>
                    <li><${siparis.getAdet()}></li>
                    <li><${siparis.getFiyat()}></li>
                    <li><${siparis.getSiparisTarihi()}></li>
                </ul>
                <div class="clearfix"> </div>
            </c:forEach>
        </div>
    </div>
</div>
</div>
</body>
</html>

```

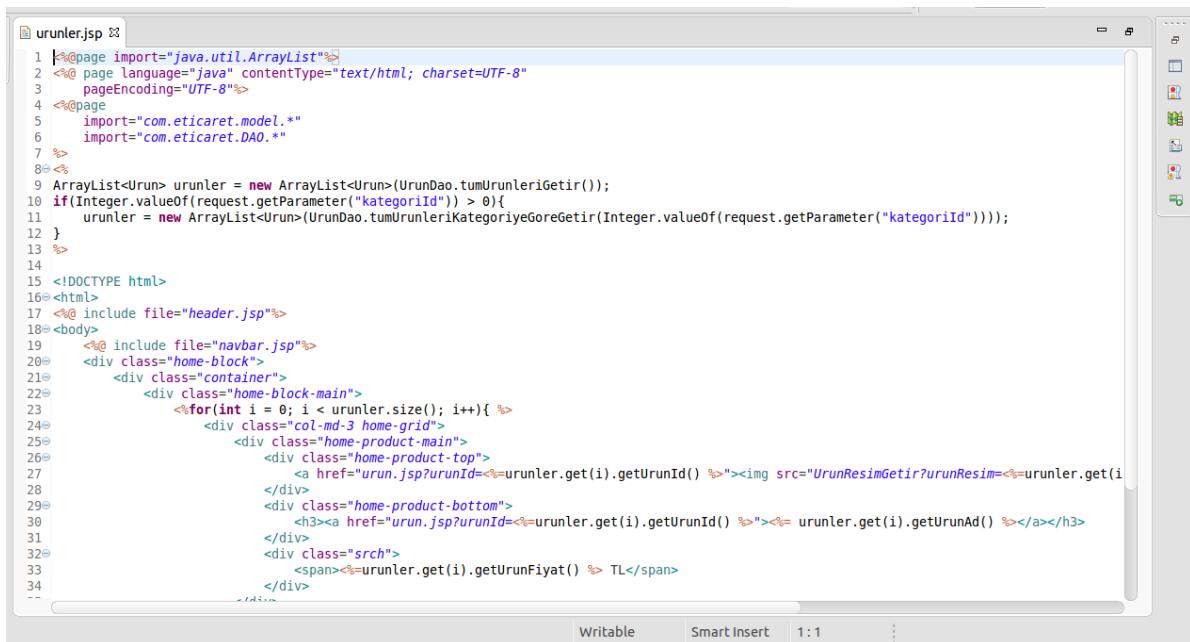
Urun.jsp

```
1 <%@page import="java.util.ArrayList"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3   pageEncoding="UTF-8"%>
4 <%@page
5   import="com.eticaret.model.*"
6   import="com.eticaret.DAO.*"
7 %>
8 <% ArrayList<Urun> urun = new ArrayList<Urun>(UrunDao.tekUrunGetir(Integer.valueOf(request.getParameter("urunId"))));%> <!-- yolladığım urunId -->
9 <!DOCTYPE html>
10 <html>
11 <%@ include file="header.jsp"%>
12 <body>
13   <%@ include file="navbar.jsp"%>
14   <div class="single">
15     <div class="container">
16       <div class="single-main">
17         <div class="single-top-main">
18           <div class="col-md-5 single-top">
19             
20           </div>
21           <div class="col-md-7 single-top-left simpleCart_shelfItem">
22             <h1><%= urun.get(0).getUrundai() %></h1>
23             <h6 class="item_price"><%=urun.get(0).getUrunFiyat() %></h6>
24             <p><%=urun.get(0).getUrunAciklama() %></p>
25             <ul class="bann-btns">
26               <li><a href="SepeteEkle?urunId=<%=urun.get(0).getUrundai() %>" class="item_add">Sepete At</a></li>
27             </ul>
28           </div>
29           <div class="clearfix"> </div>
30         </div>
31       </div>
32     </div>
33   </div>
34 </body>
-->
```

UrunIslemleri.jsp

```
1 <%@page import="java.util.ArrayList"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3   pageEncoding="UTF-8"%>
4 <%@page
5   import="com.eticaret.model.*"
6   import="com.eticaret.DAO.*"
7 %>
8 <% ArrayList<Urun> urunler = new ArrayList<Urun>(UrunDao.tumUrunleriGetir()); %>
9 <% ArrayList<Marka> markalar = new ArrayList<Marka>(MarkaDao.markalariGetir()); %>
10 <% ArrayList<Kategori> kategoriler = new ArrayList<Kategori>(KategoriDao.butunKategoriler()); %>
11 <!DOCTYPE html>
12 <html>
13 <%@ include file="header.jsp"%>
14 <body>
15   <%@ include file="navbar.jsp"%>
16   <div style="width: 50%;float: left;">
17     <div class="login">
18       <div class="container">
19         <div class="login-main">
20           <div class="col-md-12 login-left">
21             <h2>Ürün Ekle</h2>
22             <form role="form" method="Post" action="UrunEkle" enctype="multipart/form-data">
23               Ürün Adı : <input name="urunAd" type="text" /><br>
24               Ürün Açıklama : <input name="urunAciklama" type="text" /><br>
25               Ürün Fiyat : <input name="urunFiyat" type="number" /><br>
26               Ürün Miktar : <input name="urunMiktar" type="number" /><br>
27
28               Ürün Kategori :
29               <select name="urunKategori">
30                 <for(int i = 0; i < kategoriler.size(); i++)%>
31                   <option value="<%= kategoriler.get(i).getKategoriId() %>"><%= kategoriler.get(i).getKategoriAdi() %></option>
32                 <%}>
33               </select>
34             <br>
-->
```

Urunler.jsp



```
1 <%@page import="java.util.ArrayList"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3   pageEncoding="UTF-8"%>
4 <%@page
5   import="com.eticaret.model.*"
6   import="com.eticaret.DAO.*"
7 %>
8 <%>
9 ArrayList<Urun> urunler = new ArrayList<Urun>(Urundao.tumUrunleriGetir());
10 if(Integer.valueOf(request.getParameter("kategoriId")) > 0){
11   urunler = new ArrayList<Urun>(Urundao.tumUrunleriKategoriyeGoreGetir(Integer.valueOf(request.getParameter("kategoriId"))));
12 }
13 <%>
14
15 <!DOCTYPE html>
16 <html>
17 <%@ include file="header.jsp"%>
18 <body>
19   <%@ include file="navbar.jsp"%>
20   <div class="home-block">
21     <div class="container">
22       <div class="home-block-main">
23         <%for(int i = 0; i < urunler.size(); i++){ %>
24           <div class="col-md-3 home-grid">
25             <div class="home-product-main">
26               <div class="home-product-top">
27                 <a href="urun.jsp?urunId=<%=urunler.get(i).getUrundai() %>"></a>
28               <div class="home-product-bottom">
29                 <h3><a href="urun.jsp?urunId=<%=urunler.get(i).getUrundai() %>"><%=urunler.get(i).getUrundai() %><%= urunler.get(i).getUrunAd() %></a></h3>
30                 <div class="srch">
31                   <span><%=urunler.get(i).getUrunFiyat() %> TL</span>
32                 </div>
33               </div>
34             </div>
35           </div>
36         <%}>
37       </div>
38     </div>
39   </div>
40 </body>
41 </html>
```

Proje Maven projesi olduğu için pom.xml içeriği:



```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.hatice</groupId>
5   <artifactId>e-ticaret</artifactId>
6   <version>0.0.1</version>
7   <packaging>war</packaging>
8   <build>
9     <sourceDirectory>src</sourceDirectory>
10    <plugins>
11      <plugin>
12        <artifactId>maven-compiler-plugin</artifactId>
13        <version>3.3</version>
14        <configuration>
15          <source>1.7</source>
16          <target>1.7</target>
17        </configuration>
18      </plugin>
19      <plugin>
20        <artifactId>maven-war-plugin</artifactId>
21        <version>2.6</version>
22        <configuration>
23          <warSourceDirectory>WebContent</warSourceDirectory>
24          <failOnMissingWebXml>false</failOnMissingWebXml>
25        </configuration>
26      </plugin>
27    </plugins>
28  </build>
29
30
31  <dependencies>
32
33
```

The screenshot shows the IntelliJ IDEA IDE with the pom.xml file open in the main editor window. The code is displayed in a syntax-highlighted format with line numbers on the left. The code includes configuration for build plugins, dependencies on MySQL connector, JSTL, and Java Servlet API, and a dependency management section. Below the editor, there is a navigation bar with tabs for Overview, Dependencies, Dependency Hierarchy, Effective POM, and pom.xml. On the right side of the interface, there is a vertical toolbar with various icons for navigating through the project and performing other operations.

```
24         <failOnMissingWebXml>false</failOnMissingWebXml>
25     </configuration>
26   </plugin>
27 </plugins>
28 </build>
29
30
31 <dependencies>
32
33   <dependency>
34     <groupId>mysql</groupId>
35     <artifactId>mysql-connector-java</artifactId>
36     <version>5.1.32</version>
37   </dependency>
38
39   <dependency>
40     <groupId>javax.servlet</groupId>
41     <artifactId>jstl</artifactId>
42     <version>1.2</version>
43   </dependency>
44
45   <dependency>
46     <groupId>javax.servlet</groupId>
47     <artifactId>javax.servlet-api</artifactId>
48     <version>3.1.0</version>
49   </dependency>
50
51
52
53
54 </dependencies>
55 </project>
```