

T.C. Çanakkale Onsekiz Mart Üniversitesi

Bilgisayar Mühendisliği Bölümü



Nesneye Yönelik Analiz ve Tasarım Dersi

Proje Konusu *Kargo Sistemi*

Rapor10 Tüm Raporlar

Proje Grubu *130401020 Merve ATA*

130401027 Hatice ERTÜRK

Dersin Danışman Hocası *Yrd.Doç.Dr. ALİ MURAT TİRYAKİ*

İÇİNDEKİLER

Vizyon	3
Actor – Goal Model	4
Use Case Model.....	5
En Kritik Use Case: Use Case 1	6
En Kritik Use Case: Use Case 2	8
En Kritik Use Case: Use Case 3	9
Supplementary Specification (Ek Gereksinimler)	11
Business Rules (Ticari Kurallar)	12
Glossary (Sözlük)	15
Use Case 1 Domain Model	16
Use Case 2 Domain Model	17
Use Case 3 Domain Model	18
Use Case 1 SSD Model	19
Use Case 1 Machine Model	19
Use Case 2 SSD Model	20
Use Case 2 Machine Model	20
Use Case 3 SSD Model	21
Use Case 3 Machine Model	21
Use Case 1 Operation Contract	22
Use Case 2 Operation Contract	24
Use Case 3 Operation Contract	25
Logical Architecture	26
Use Case 1 Interaction Diagram	27
Use Case 2 Interaction Diagram	28
Use Case 3 Interaction Diagram	29
Class Diagram.....	30
Use Case 1 Interaction Diagram(Facade).....	31
Use Case 1 Interaction Diagram (Facade).....	32
Use Case 1 Interaction Diagram(Facade).....	33
Class Diagram(Facade).....	34
Test.....	35

VİZYON

Kargo sistemi, kargo göndermek isteyen kişiler ile kargo taşıyan kişileri birleştirerek en iyi şekilde ve en ucuz fiyat ile kargo iletimini sağlamayı amaçlamaktadır.

Sistem, müşteri açısından kargo bilgilerini girme, taşınmasını istediği kargo için verebileceği maksimum ücreti belirtme, açık eksiltme sonucunda en düşük 3 taşıyıcı arasından seçim yapma ve kargo gönderimi sırasında kargonun durumu hakkını bilgi sahibi olacaktır. Taşıyıcı açısından, sahip olunan araç bilgisi ve taşıyabilecekleri ürünü belirtme, sistem tarafından uygun ürünlerin gösterilip açık eksiltmeye girme, kazanılan açık eksiltme sonucu alınan ürünün ulaşım sürecindeki yol durumunu sisteme girme işlemini yapacaktır.

Sistemin yetkili bir sistem yöneticisi olacaktır. Sistem mimarisini oluşturan parçaların sürekli, kararlı ve güvenli olacak şekilde çalışmasını sağlamak amacıyla sistemi izleyip performans değerlendirmesi yapacaktır. Sistem ve iletişim altyapısını kurmak veya geliştirmek amacıyla kurum ihtiyaçlarının analizini yapacaktır.

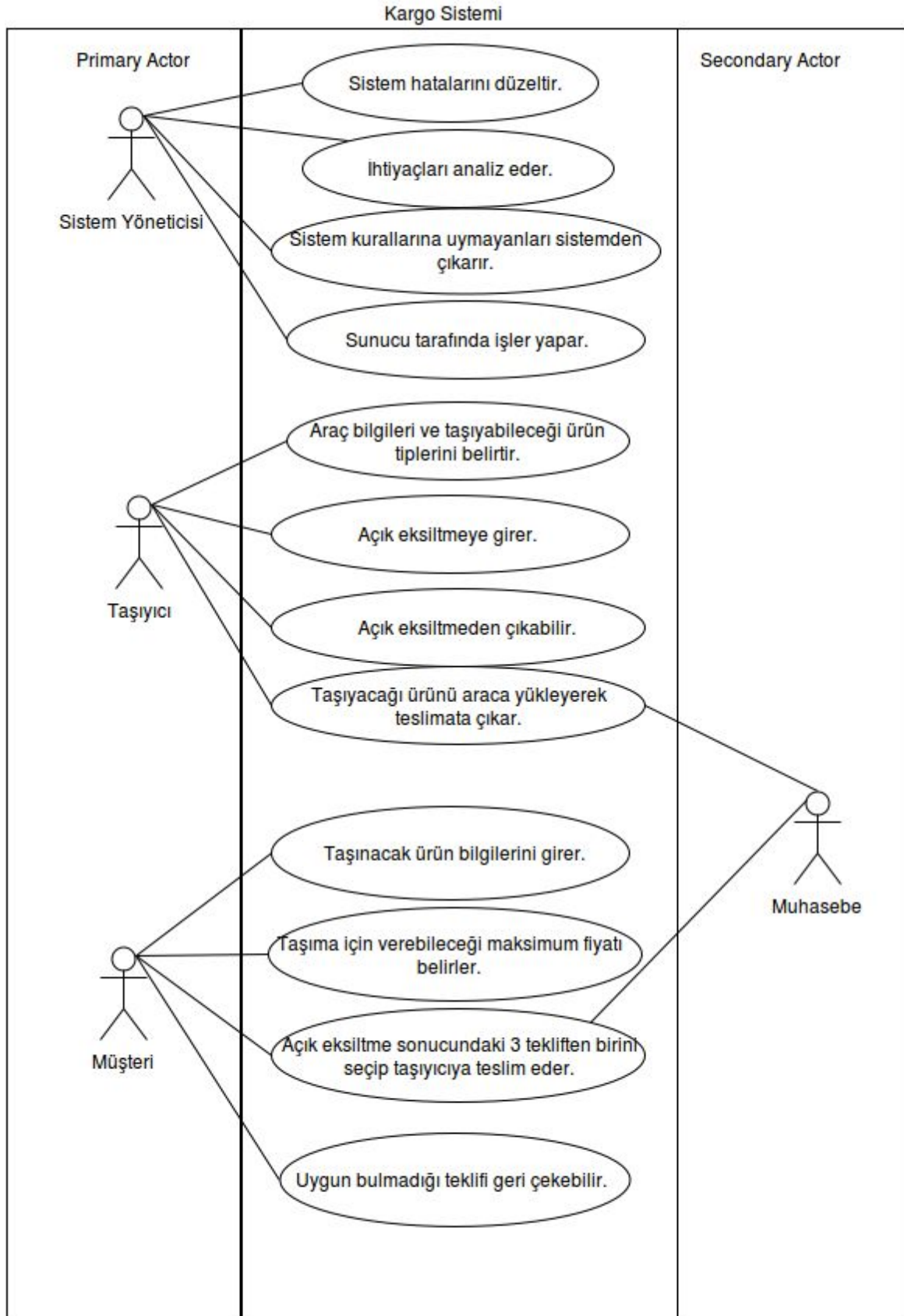
Belirli bir fiyatın üzerindeki taşıma işlemlerini sadece sistemin yüksek üyelikli yani altın üye taşıyıcıları taşıyabilecektir.

Müşteri her kargo işlemi sonucunda taşıyıcıya iyi veya kötü bir referans verip daha sonraki müşteriler için güvenilirlik düzeyini belirleyebilecektir. Bu sayede müşteri mağduriyetini en aza indirmek hedeflenmektedir.

ACTOR-GOAL MODEL

Sistem Yöneticisi	<ul style="list-style-type: none">• Sistemde meydana gelebilecek aksaklıkları giderir.• Kurum ihtiyaçlarını analiz yapar.• Sistem kurallarına uymayan müşteri veya taşıyıcıyı sistemden çıkarır.• Sunucu tarafında yapılacak işleri gerçekleştirir.
Taşıyıcı	<ul style="list-style-type: none">• Araç bilgilerini ve ne tür ürünleri taşıyabileceğini belirtir.• Belirttiği özellikler ile sistem tarafından karşısına taşıyabileceği ürünler çıkar.• Taşıyıcı karşısına çıkan ürünlerden taşımak istediği ürün için açık eksiltmeye girer.• Açık eksiltmede taşıyacağı ürün için verebileceği en düşük fiyatı belirler.• Açık eksiltmeye girdiğinde fiyat çok düşük gelirse çıkabilir.• Açık eksiltmede kazandığı ürünü sahip olduğu araca yükleyerek zamanında yerine teslim etmeye çalışır.• Teslimat yolunda çıkabilecek sorunlarla baş etmeye çalışır.
Müşteri	<ul style="list-style-type: none">• Taşınmasını istediği ürün bilgilerini ve taşıma için verebileceği maksimum fiyatı belirtir.• Sistemin açık eksiltme sonucunda kendisine gönderilecek 3 teklifi bekler.• Açık eksiltme sonucunda gelen 3 teklif arasından taşıyıcıların verdiği fiyata ve referanslarına bakar.• Açık eksiltme sonucunda gelen tekliflerden memnun değil veya güven duyamadıysa taşıma işlemini geri çeker.• Açık eksiltme sonucunda gelen tekliflerden kendisine en uygun olanı seçerek taşıyıcının ürünü iletmesini bekler ve iletim yolunu izler.

USE-CASE MODEL



SEÇİLEN EN KRİTİK 3 USE-CASE

USE-CASE 1 :

Müşterinin sisteme yeni bir kargo işi eklemesi

Scope : Kargo Ekleme Sistemi

Level : User Goal

Primary Actor : Müşteri

Stakeholders and Interests :

- Müşteri : Kargo için ödeyeceği ücretin belirlediği maksimum fiyattan olabildiğince düşük olmasını ister. Güvenilir ve özverili bir taşıyıcının ürününü taşımasını ister.

Preconditions : Müşteri sisteme kayıtlı olmalıdır ve giriş yapmış olmalıdır.

Success Guarantee : Müşteri sisteme yeni bir kargo eklemiş olmalıdır.

Main Success Scenario :

1. Müşteri sisteme kargo ekleme isteği yapar.
2. Sistem müşteriden kargolamak istediği ürünün bilgilerini girmesini ister.
3. Müşteri sisteme ürün tipini(sıvı, katı, atık vb.), ağırlığını, ebatlarını, adet sayısını girer ve ürünü kaydeder.
4. Sistem müşteriye ürün bilgilerini ve kargo bilgilerini gösterir.
5. Müşteri Açık Eksiltme oluşturma isteğinde bulunur.
6. Sistem Açık Eksiltme bilgilerinin girilmesini bekler.
7. Müşteri Açık Eksiltme için verebileceği maksimum ücreti, kargonun teslim alınacağı ve teslim edileceği adres bilgisini, taşımanın gerçekleşeceği tarih aralığını ve açık eksiltme bitiş tarihini girer.
8. Sistem girilen bilgileri kayıt eder.

Extensions :

*a. Sistem herhangi bir anda çöker ise,

1. Sistemi yeniden başlatır.
2. İşleme 2. adımdan devam eder.

3.a. Müşteri ürün bilgilerini eksik girer ise,

1. Sistem tarafından müşteriye hata mesajı gösterilir.

2. Müşteriden eksik bilgilerinin doldurulmasını istenir.
3. Müşteri eksik bilgileri doldurur ve ana başarı senaryosunun 4. adımından devam eder.

6.a. Müşteri Açık Eksiltme bilgilerini eksik girer ise,

1. Sistem tarafından müşteriye hata mesajı gösterilir.
2. Müşteriden eksik bilgilerinin doldurulmasını istenir.
3. Müşteri eksik bilgileri doldurur ve ana başarı senaryosunun 4. adımından devam eder.

USE-CASE 2 :**Taşıyıcının teklif vermesi**

Scope : Teklif Verme Sistemi

Level : User Goal

Primary Actor : Taşıyıcı

Stakeholders and Interests :

- Taşıyıcı : Hem kâr edip hemde en düşük fiyatı vererek ürün taşımayı ister. Yaptığı doğru ve güvenli taşıma sonrası müşteriler tarafından güzel referanslar verilmesini bekler.

Preconditions : Taşıyıcı sisteme kayıtlı olup giriş yapmış olmalıdır ve müşterinin sisteme kargoyu eklemiş olması gerekmektedir.

Success Guarantee : Müşteriye sunulacak olan 3 teklif belirlenmiş olmalıdır.

Main Success Scenario :

1. Taşıyıcı taşıyabileceği Açık Eksiltmeleri görmek için istekte bulunur.
2. Sistem taşıyıcıya katılabileceği Açık Eksiltmeleri gösterir.
3. Taşıyıcı seçtiği Açık Eksiltme için teklif verir.
4. Sistem taşıyıcıya, katıldığı Açık Eksiltme için diğer taşıyıcıların verdiği ücretleri gösterir.
5. Taşıyıcı Açık Eksiltmeyi kazanmak için sisteme daha ucuz teklifi girer.
6. Sistem teklifi kayıt eder.

Extensions :

*a. Sistem herhangi bir anda çöker ise,

1. Taşıyıcı sistemi yeniden başlatır ve tekrardan giriş yapar.
2. Ana başarı senaryosunun 1. adımından devam eder.

3.a. Diğer taşıyıcıların önerdikleri teklifler taşıyıcının minimum verebileceği fiyattan düşük ise,

1. Taşıyıcı Açık Eksiltmeden çıkar.
2. Ana başarı senaryosunun 1. adımından devam eder.

USE-CASE 3 :**Müşterinin teklif verme sonucuna göre istediği teklifi seçmesi**

Scope : Teklif Seçme İşlemi

Level : User Goal

Primary Actor : Müşteri

Stakeholders and Interests :

- Müşteri : Olabildiğince ucuz ve güvenilir olan taşıyıcının teklifini seçmeyi ister. Taşınacak olan kargosuna hiçbir şey olmadan ulaştırılmasını ister.
- Taşıyıcı : Beklediği fiyatı alarak sorunsuz bir şekilde kargoyu ulaştırıp güzel referanslar almayı ister.

Preconditions : Müşteri ve taşıyıcının sisteme kayıtlı olup giriş yapmış olmalı ve sistem tarafından en ucuz 3 teklif seçilmiş olmalıdır.

Success Guarantee : Müşteri kendisi için en uygun teklifi seçmiş olmalıdır.

Main Success Scenario :

1. Müşteri sisteme teklif seçme isteğinde bulunur.
2. Sistem müşteriye süresi bitmiş Açık Eksiltmelerini gösterir.
3. Müşteri süresi bitmiş Açık Eksiltmelerinden birini seçer.
4. Sistem müşteriye seçtiği Açık Eksiltme için verilen en düşük 3 teklifi gösterir.
5. Müşteri sisteme teklifleri veren taşıyıcı bilgilerini görme isteğinde bulunur.
6. Sistem teklifi veren taşıyıcı bilgilerini gösterir.
7. Müşteri 3 tekliften kendisine uygun olanı seçer.
8. Sistem seçilen teklifi kayıt eder.

Extensions :

*a. Sistem herhangi bir anda çöker ise,

1. Sistem yeniden başlatılır.

1.a. Açık Eksiltmeye hiçbir teklif gelmez ise,

1. Müşteri Açık Eksiltme işlemini iptal eder.

1.b. Açık Eksiltme sonucunda hiçbir teklif gelmez ise,

1. Müşteri Açık Eksiltme işlemini bekletir.

2. Daha sonraki günlerde Teklif Vermeyi tekrar başlatır.

2.a. Müşteri 3 tekliften hiçbirini seçmek istemez ise,

1. Açık Eksiltmeyi iptal eder.

2.b. Müşteri 3 tekliften hiçbirini seçmek istemez ise,

1. Fiyat aralığını yeniden belirler.

2. Teklif Vermeyi yeniden başlatır.

2.c. Müşteri 3 tekliften hiçbirini seçmek istemez ise,

1. Açık Eksiltmeyi bekletir.

2. Daha sonraki günlerde Teklif Vermeyi tekrar başlatır.

SUPPLEMENTARY SPECIFICATION (EK GEREKSİNİMLER)

Functionality (Fonksiyonellik) :

- Sistem kullanıcıları sisteme kullanıcı adı ve parola ile giriş yapmalıdır.
- Aynı kullanıcı adına sahip üye olmamalıdır.
- Üyeler sistemde çıkan sorunları sistem yöneticisine bildirebilmelidir.
- Müşteri ürün takibini yapabilmelidir.

Security (Güvenlik) :

- Sisteme sadece sistem yöneticisi tarafından erişilebilmelidir.
- Şifre politikası uygulanmalıdır. (Minimum bir büyük, küçük harf ve bir sayı.)
- Üye bilgileri kimse ile paylaşılmamalıdır.
- Sistem bakımını sadece sistem yöneticisi yapmalıdır.
- Yetkili kişiler sistemi güvenli bir şekilde kullanmalıdır.

Usability (Kullanılabilirlik) :

- Sistem her kullanıcı türü için anlaşılır, sade ve kullanılabilir olmalıdır.
- Sistemde yapılmak istenen işlemler en hızlı şekilde yapılmalıdır.

Performance (Performans) :

- Sistem, kapatılana kadar sorunsuzca çalışmalıdır.
- Üyelerin sistemde yapacakları işlemler kısa süre içerisinde olmalıdır.
- Sistemin aynı anda birden fazla kullanıcıyı kaldırma kapasitesi olmalıdır.

Supportability (Bakım) :

- Sistemin bakımı üyelerin en az işlem yaptığı vakitlerde yapılmalıdır.
- Sistem bakımı yapılırken oluşabilecek sorunlara karşı, sistemde giriş yapmış üyeler uyarılarak çıkışları yapılmalıdır.
- Bakım yapılacağına yakın mağduriyet olmaması için daha önceden üyelere haber verilmelidir.
- Sisteme yeni donanım eklendiğinde uyumluluk sağlanmalıdır.

Implementation Constraints (Sistem Kısıtlamaları) :

- Sistemde bulunması gereken belgeler ve sertifikalar eksiksiz bir şekilde sistem yöneticisinde bulunmalıdır.
- Belli zamanlarda sistem istatistiği tutulmalıdır.
- Sistemi kullanmak için mutlaka üyelerin interneti olmalıdır.
- Üyeler sistem kurallarına uymalıdır.
- Sistem kurallarına uymayan üyeler yöneticiler tarafından sistemden uzaklaştırılmalıdır.

Domain (Business) Rules (Ticari Kurallar)

ID	Rule	Changeability	Source
RULE1	<p>Taşıyıcılar İçin Genel Kurallar</p> <ul style="list-style-type: none">- Taşıyıcılar, araçlarına uygun ehliyete sahip olmak zorundadırlar.- Uzun yol taşımacılığı yapacak olan taşıyıcılar, araç da yardımcı şoför bulundurmak zorundadırlar.- Müşteri ile yapılan sözleşmeye uymak zorundadırlar.- Taşıyıcılar, ürün güvenliğini en üst seviyede tutmak zorundadırlar.- Taşıyıcılar bilgilerini doğru belirtmek zorundadır.(Ev adresi, telefon, vb.)- Taşıyıcılar, bakanlıktan taşıma belgesi almak zorundadırlar.	Düşük Değişilebilirlik	Sistem Taşıyıcılar Politikası
RULE2	<p>Sahtecilik Kuralı</p> <ul style="list-style-type: none">- Taşıyıcı veya müşteri sahtecilik yapmaya çalışırsa, para cezası veya hapis cezası alır.	Düşük Değişilebilirlik	Ceza Hukuku ve Suç Politikası
RULE3	<p>Gecikme Kesinti Parası</p> <ul style="list-style-type: none">- Taşıyıcı, son teslim tarihini geçirdiği her gün başına ücretin %0.5'lik miktarını müşteriye geri ödemek zorundadır.	Olağan Değişilebilirlik	Sistem Ticari Politikası

RULE4	<p>Müşteriler İçin Genel Kurallar</p> <ul style="list-style-type: none"> - Taşıyıcı ile yapılan sözleşmeye uymak zorundadırlar. - Müşteriler bilgilerini doğru belirtmek zorundadırlar.(Ev adresi, telefon, vb.) 	Düşük Değişilebilirlik	Sistem Müşteriler Politikası
RULE5	<p>3.4. Kargo ve Lojistik İşletmeleri 11/6/2009 tarih ve 27255 sayılı Resmî Gazete’de yayımlanan Karayolu Taşıma Yönetmeliği uyarınca M1, M2 ve M3 yetki belgesi alan kargo ve lojistik işletmecileri, sahip oldukları bu yetki belgeleri kapsamında takvim yılının birer aylık süreleri içerisinde gerçekleştirdikleri kargo teslimleri ile ilgili olarak; a) Göndericinin ad soyad/unvan ve TCKN/VKN bilgileri ile adres bilgilerini, aracı hizmet sağlayıcılar tarafından yapılan gönderimlerde gönderici bilgileri yanında aracı hizmet sağlayıcının da ad soyad/unvan ve TCKN/VKN bilgilerini, b) Gönderici bazında teslim edilen toplam gönderi adedini, (Başkanlık tarafından belirlenecek adedin üzerindeki gönderi sayıları bildirilecektir.) c) Gönderici bazında kapıda ödeme yapılan toplam gönderi adedini, ödeme şeklini ve toplam ödeme tutarlarını, bu Tebliğin 4 üncü bölümünde açıklanan yöntemle elektronik ortamda Başkanlık sistemine ileteceklerdir. 5018 sayılı Kamu Mali Yönetimi</p>	Düşük Değişilebilirlik	VERGİ USUL KANUNU

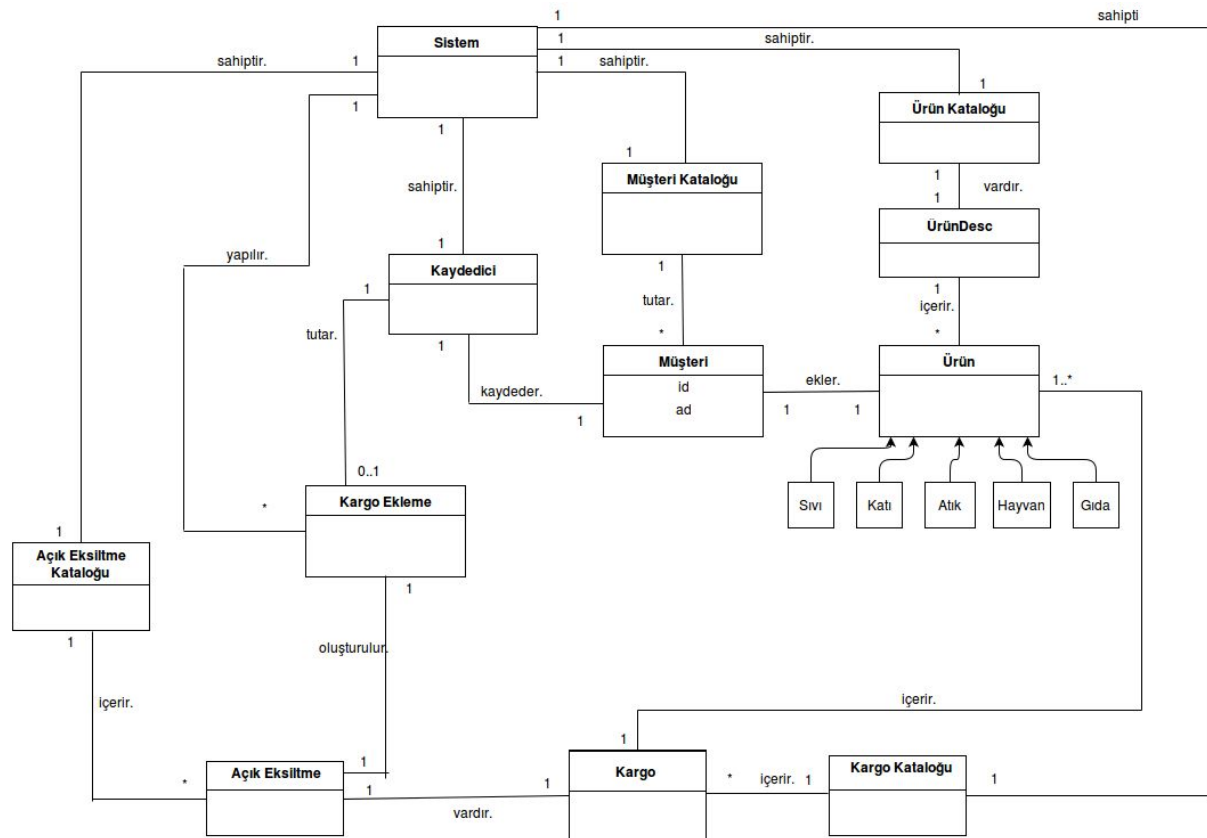
	<p>ve Kontrol Kanununa ekli cetvellerde yer alan idare, kurum ve kuruluşlar, idaresinde veya sermayesinde kamu payı bulunan kurumlar, kamu kurumu niteliğindeki meslek kuruluşları ve diplomatik misyonlar adına yapılan kargo teslimlerine ait bilgiler ileilmeyecektir.</p> <p>Kargo ve lojistik işletmeleri ticari nitelikteki gönderileri Vergi Usul Kanunu uyarınca düzenlenmesi zorunlu tevsik edici belgeler ile birlikte kabul edecekler ve taşıma esnasında istenildiğinde ibraz edilecek şekilde gönderiye ekleyeceklerdir.</p>		
--	---	--	--

GLOSSARY

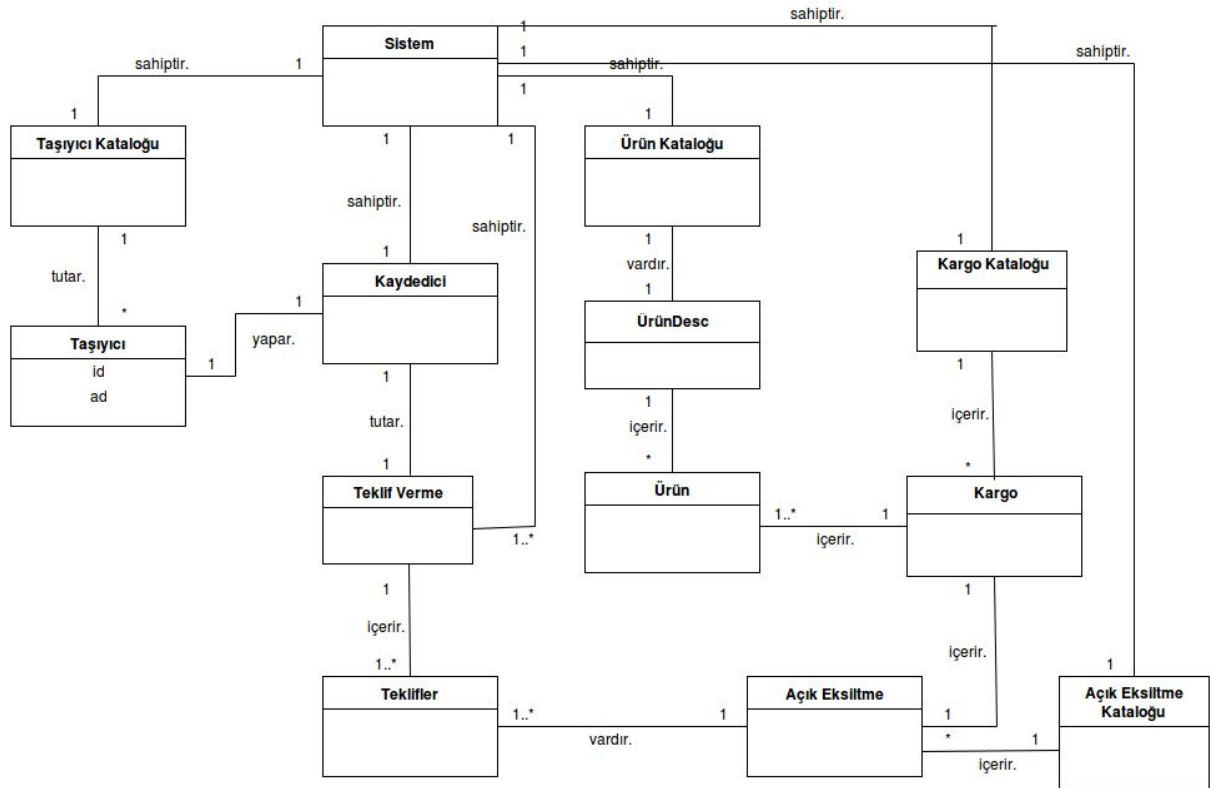
Taşıyıcı	Ücret karşılığında, herhangi bir araç ile ürünü teslim adresine ulaştıran kişidir.
Müşteri	Ücret ödeyerek sahip olduğu ürünü taşımak isteyen kişidir.
Kargo	Kara, hava ve deniz yollarında gönderilen yük, eşya.
Sistem Yöneticisi	Sistem mimarisini oluşturan parçaların sürekli, kararlı ve güvenli olacak şekilde çalışmasını sağlamak amacıyla sistemleri izleyip performans değerlendirmelerini yapan kişidir.
Açık Eksiltme	Müşteri ürününü taşımak için verilen tekliflerden daha ucuzunu vererek, taşıma işleminin kazanılmaya çalışıldığı bir yarıştıdır.
Hata Mesajı	Sistemde yolunda gitmeyen durumlarda, sistemin yanlış kullanılması gibi durumlarda kullanıcıyı bilgilendirmek amacıyla gösterilen bilgidir.
Referans	Taşıyıcının yaptığı işler sonucunda müşteriler tarafından aldığı iyi veya kötü değerlendirmelerdir.
Sözleşme	Kişiler arasında yapılan ve yasalarca desteklenen belgede yazılı olan kurallara uyacaklarını teyit eden anlaşmadır.
Feshetmek	Bir sözleşmeyi karşılıklı veya tek taraflı olarak geçersiz kılmak.

DOMAIN MODEL

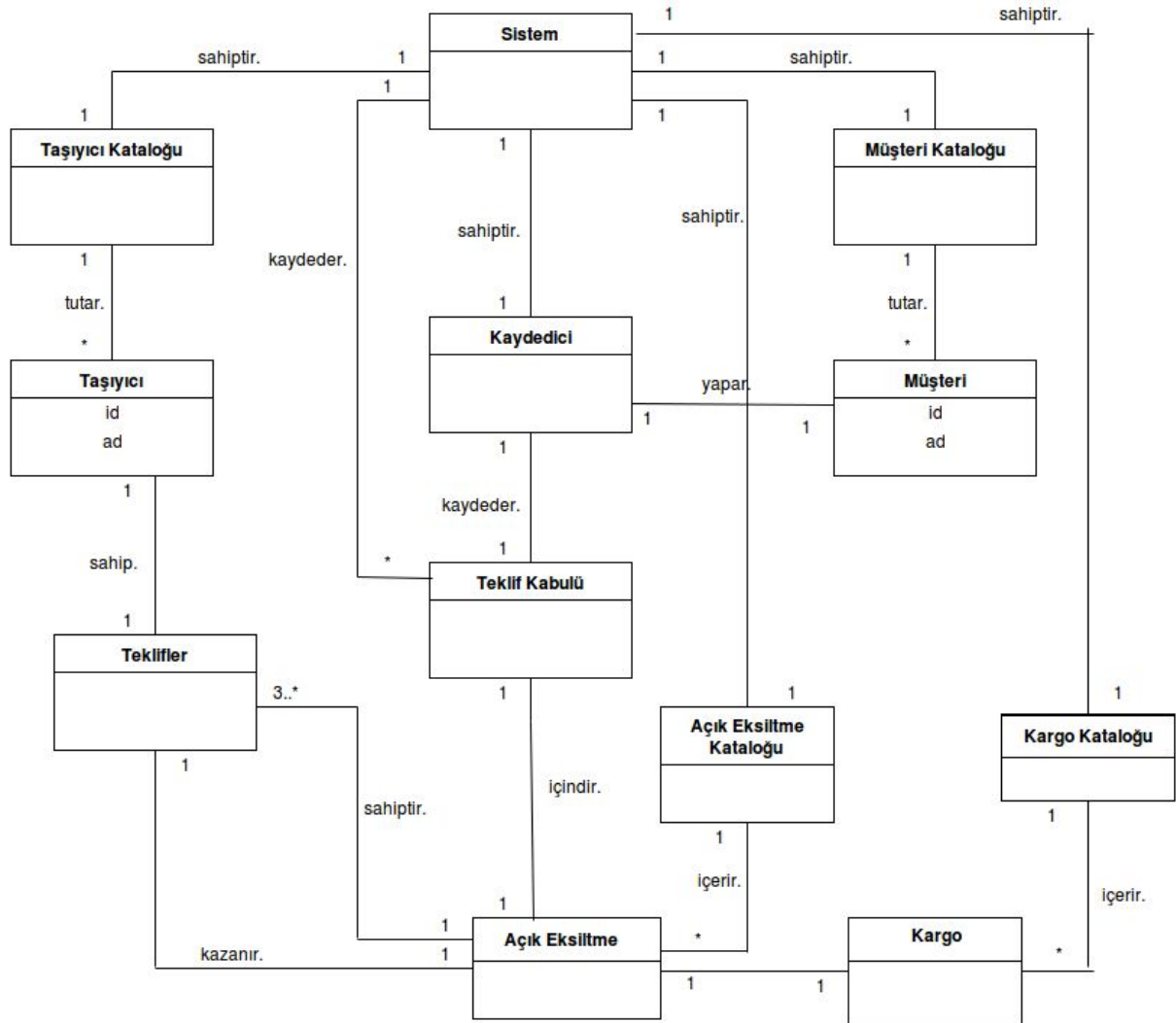
USE-CASE1 : Kargo Ekleme



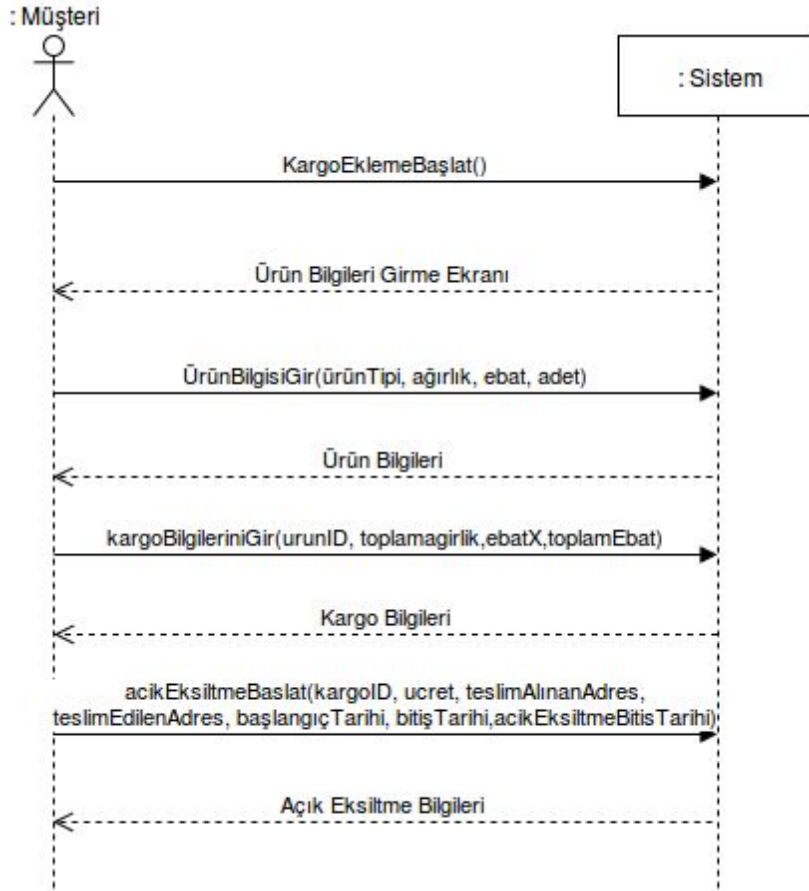
USE-CASE2 : Taşıyıcının Teklif Vermesi



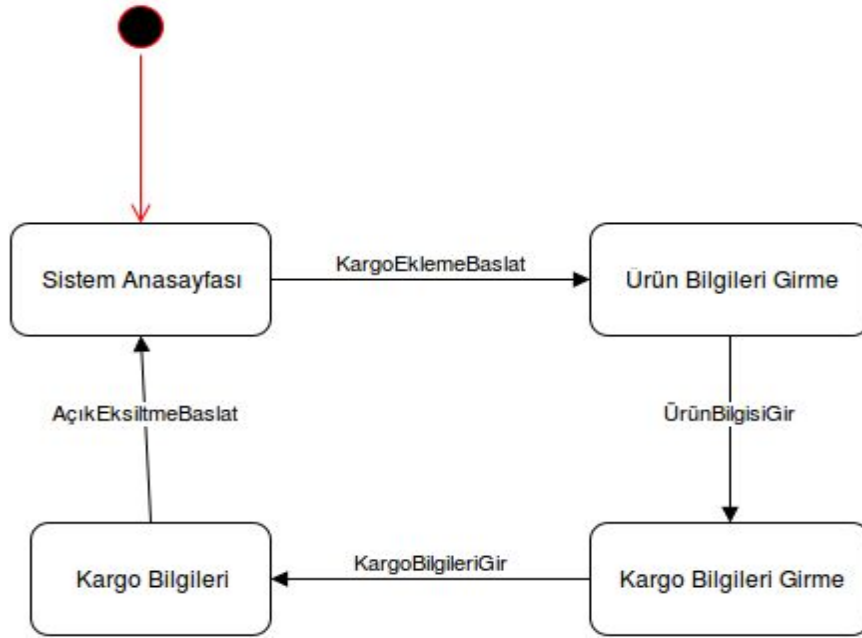
USE-CASE3 : Teklif Seçilmesi



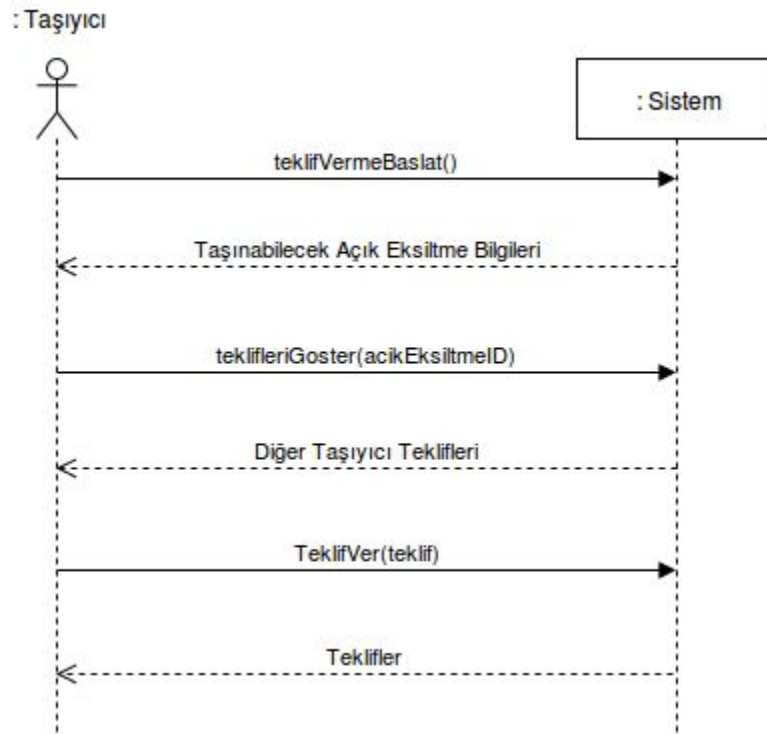
USE CASE 1 SSD MODEL



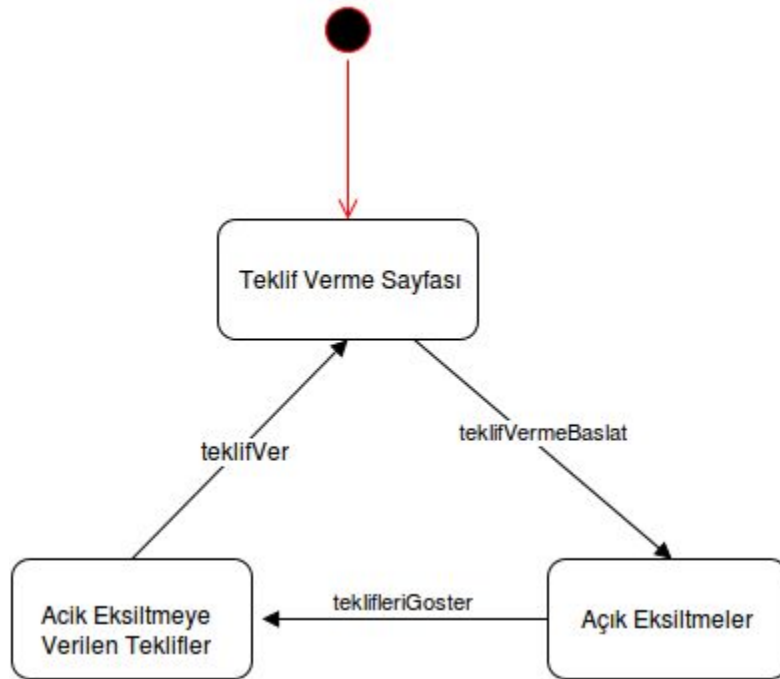
USE CASE 1 STATE MACHINE DIAGRAM



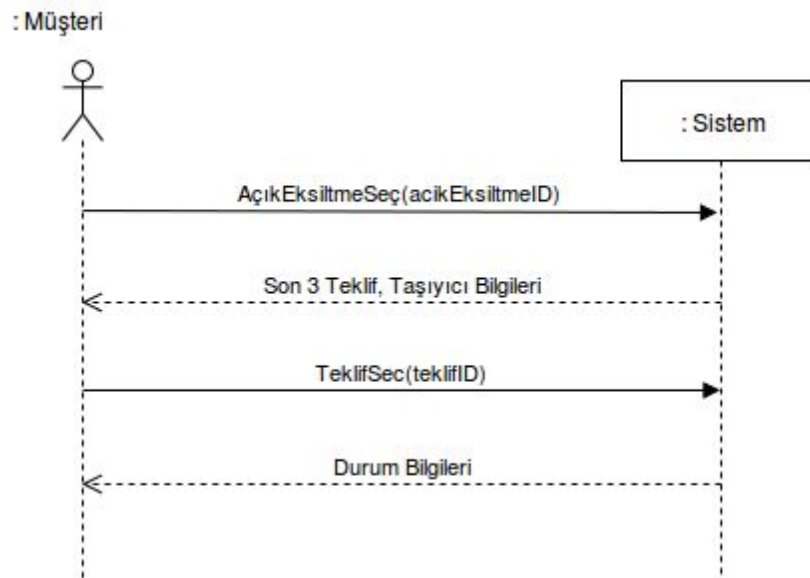
USE CASE 2 SSD MODEL



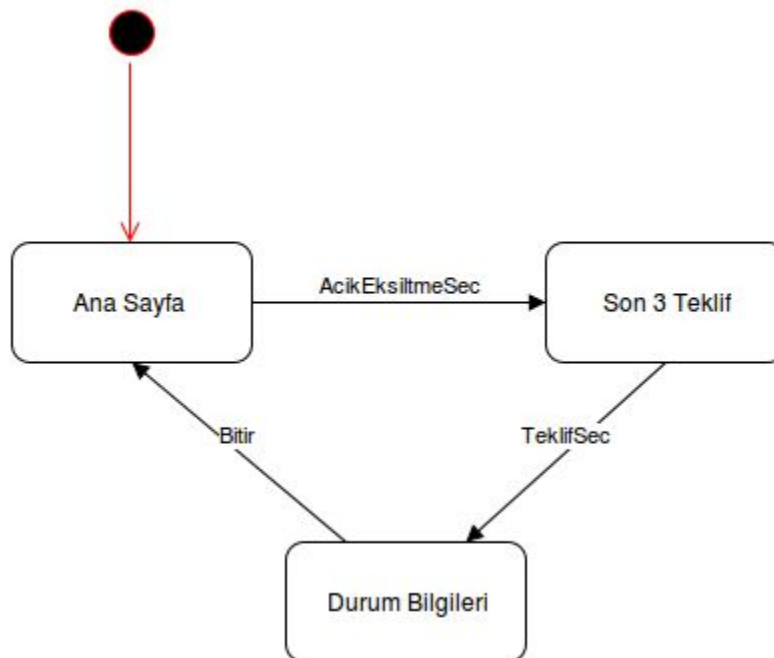
USE CASE 2 STATE MACHINE DIAGRAM



USE CASE 3 SSD MODEL



USE CASE 3 STATE MACHINE DIAGRAM



USE - CASE 1 OPERATION CONTRACT

O.C.1 = KargoEklemeBaslat() :

Operation	KargoEklemeBaslat()
Cross References	Use Cases: Kargo Ekleme
Precondition	
Postcondition	<ul style="list-style-type: none">- Kargo Ekleme sınıfının ke isminde bir nesnesi oluşturulmuş olmalıdır.- ke nesnesi Kaydedici sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.

OC2 = urunBilgisiGir() :

Operation	urunBilgisiGir(urunTipi, agirlık, ebat, adet)
Cross References	Use Cases: Kargo Ekleme
Precondition	Devam eden bir Kargo Ekleme nesnesi var olmalıdır.
Postcondition	<ul style="list-style-type: none">- u isminde bir Ürün nesnesi oluşturulmuş olmalıdır.- u nesnesi ile Müşteri sınıfının nesnesi ilişkilendirilmiş olmalıdır.- u nesnesi ÜrünDesc sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.

OC3 = kargoBilgileriGir() :

Operation	kargoBilgileriGir(urunID, toplamAgirlik, ebatX, toplamEbat)
Cross References	Use Cases: Kargo Ekleme
Precondition	Devam eden bir Kargo Ekleme nesnesi ve Ürün nesnesi var olmalıdır.
Postcondition	<ul style="list-style-type: none">- Kargo sınıfının k isminde bir nesnesi oluşturulmuş olmalıdır.- k nesnesi Kargo Kataloğu sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.- k nesnesi parametre olarak gelen urunID'nin belirttiği Ürün sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.

OC4 = acikEksiltmeBaslat() :

Operation	acikEksiltmeBaslat(kargoID,ucret,teslimAlinanAdres,teslimEdilenAdres,baslangicTarihi,bitisTarihi,acikEksiltmeBitisTarihi)
Cross References	Use Cases: Kargo Ekleme
Precondition	Devam eden bir Kargo Ekleme nesnesi, Ürün nesnesi ve Kargo nesnesi var olmalıdır.
Postcondition	<ul style="list-style-type: none">- Açık Eksiltme sınıfının ae isminde bir nesnesi oluşturulmuş olmalıdır.- ae nesnesi Açık Eksiltme Kataloğu ile ilişkilendirilmiş olmalıdır.- ae nesnesi parametre olarak gelen kargoID'nin belirttiği Kargo sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.- ae nesnesi Kargo Ekleme sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.- ae nesnesinin özellikleri başlatılmış olmalıdır.- Kargo Ekleme sınıfının nesnesi Sistem sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.

USE - CASE 2 OPERATION CONTRACT

OC1 = teklifVermeBaslat() :

Operation	teklifVermeBaslat()
Cross References	Use Cases: Taşıyıcının Teklif Vermesi
Precondition	
Postcondition	<ul style="list-style-type: none">- Teklif Verme sınıfının tv isminde bir nesnesi oluşturulmuş olmalıdır.- tv nesnesi ile Kaydedici sınıfının nesnesi ilişkilendirilmiş olmalıdır.

OC2 = teklifleriGoster() :

Operation	teklifleriGoster(acikEksiltmeID)
Cross References	Use Cases: Taşıyıcının Teklif Vermesi
Precondition	
Postcondition	<ul style="list-style-type: none">- Teklifler sınıfının t isminde bir nesnesi oluşturulmuş olmalıdır.- t nesnesi parametre olarak gelen acikEksiltmeID'nin belirttiği Açık Eksiltme sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.

OC3 = teklifVer() :

Operation	teklifVer(teklif)
Cross References	Use Cases: Taşıyıcının Teklif Vermesi
Precondition	Devam eden Teklif Verme nesnesi ve Teklifler nesnesi var olmalıdır.
Postcondition	<ul style="list-style-type: none">- Teklifler sınıfının nesnesi ile Teklif Verme sınıfının nesnesi ilişkilendirilmiş olmalıdır.- Teklif Verme sınıfının nesnesi ile Sistem sınıfının nesnesi ilişkilendirilmiş olmalıdır.

USE - CASE 3 OPERATION CONTRACT

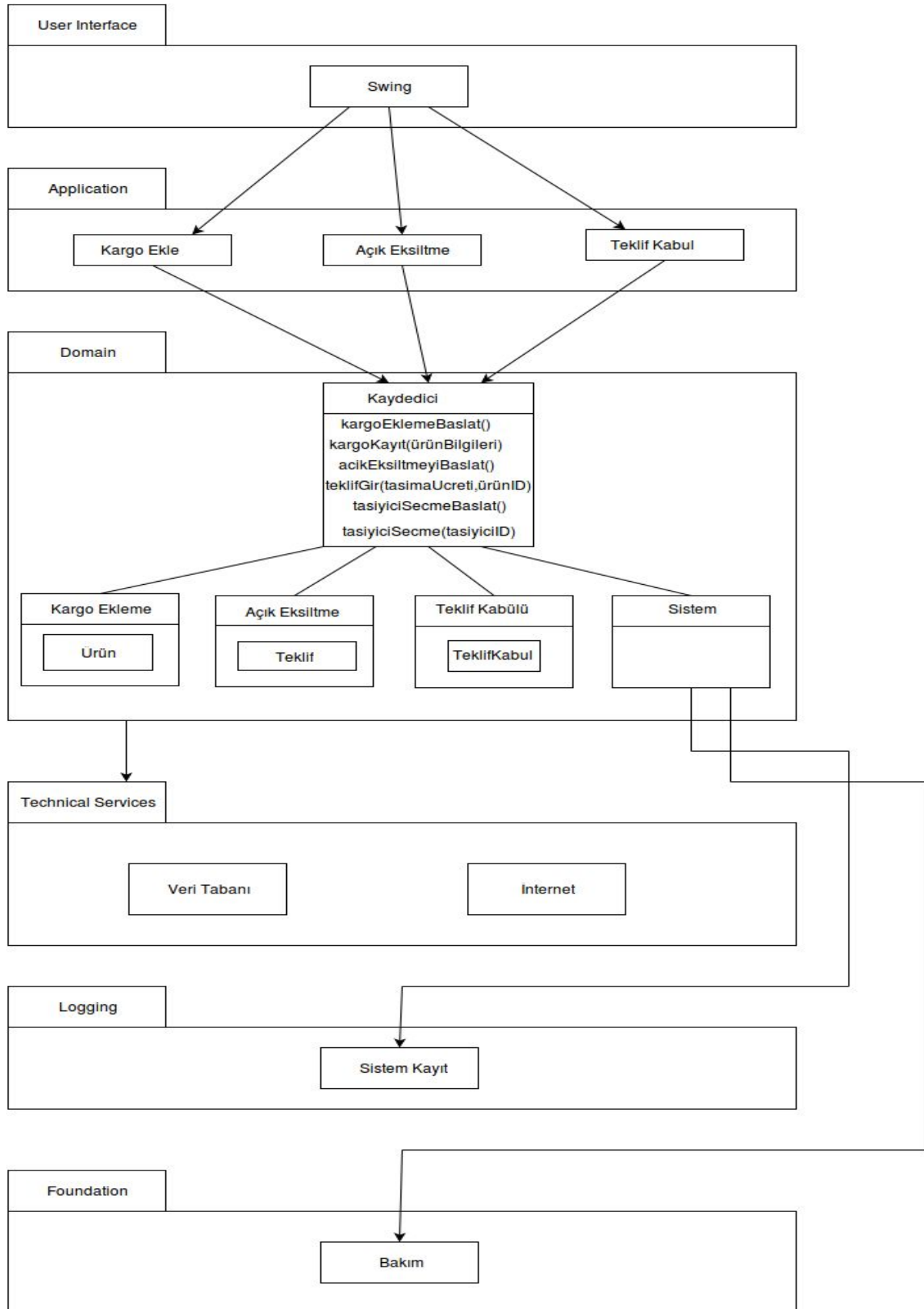
OC1 = acikEksiltmeSec() :

Operation	acikEksiltmeSec(acikEksiltmeID)
Cross References	Use Cases: Teklif Seçimi
Precondition	
Postcondition	<ul style="list-style-type: none">- Teklif Kabulü sınıfının tk isminde bir nesnesi oluşturulmuş olmalıdır.- tk nesnesi Kaydedici sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.- tk nesnesi parametre olarak gelen acikEksiltmeID'nin belirttiği Açık Eksiltme sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.

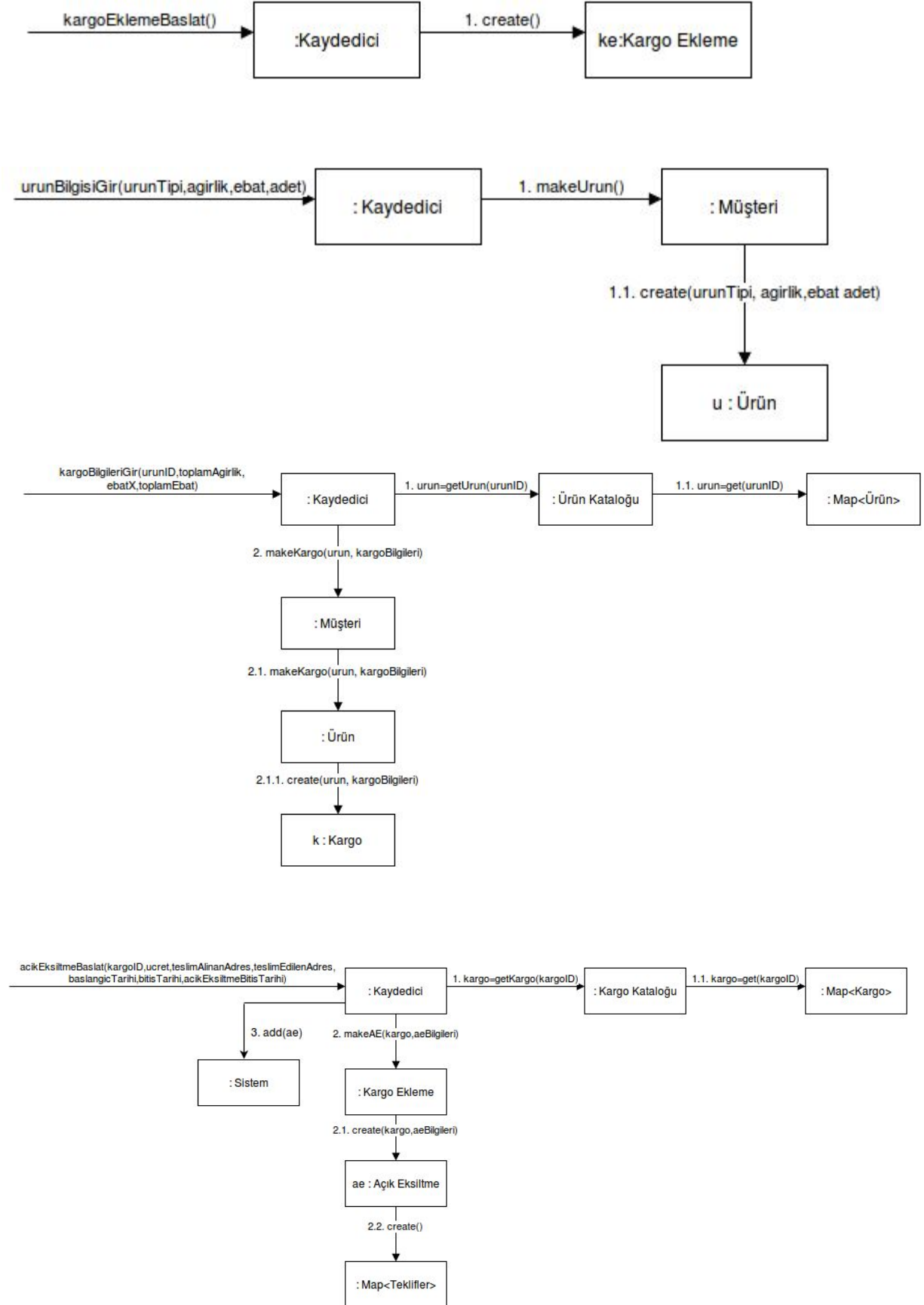
OC2 = TeklifSec() :

Operation	TeklifSec(teklifID)
Cross References	Use Cases: Teklif Seçimi
Precondition	Devam eden Teklif Kabulü nesnesi olmalıdır.
Postcondition	<ul style="list-style-type: none">- Devam eden Açık Eksiltme nesnesi parametre olarak gelen teklifID'nin belirttiği Teklifler sınıfının nesnesi ile ilişkilendirilmiş olmalıdır.- Teklif Kabulü sınıfının nesnesi ile Sistem sınıfının nesnesi ilişkilendirilmiş olmalıdır.

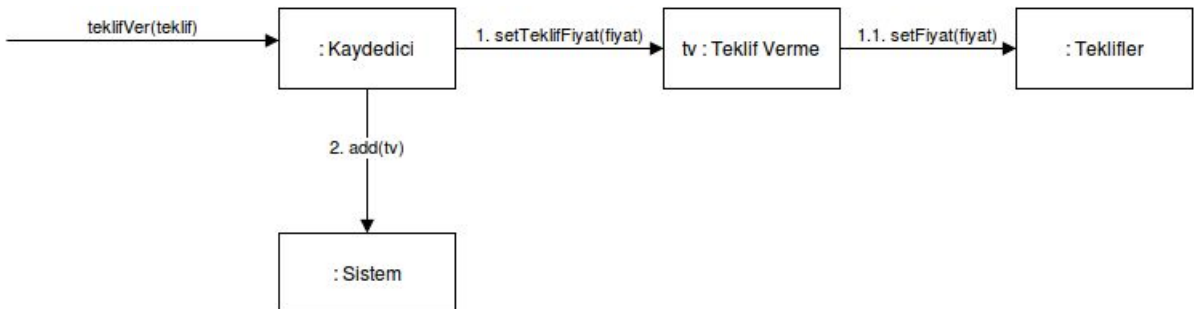
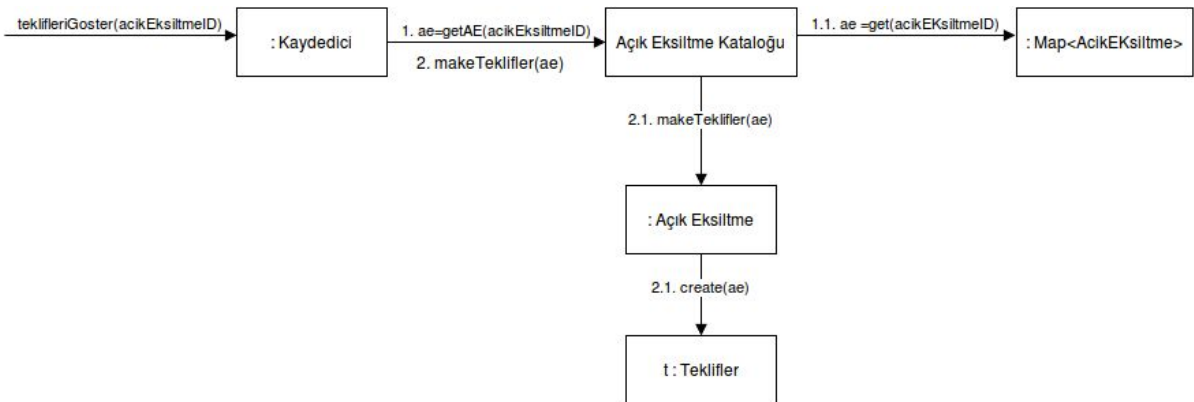
Logical Architecture



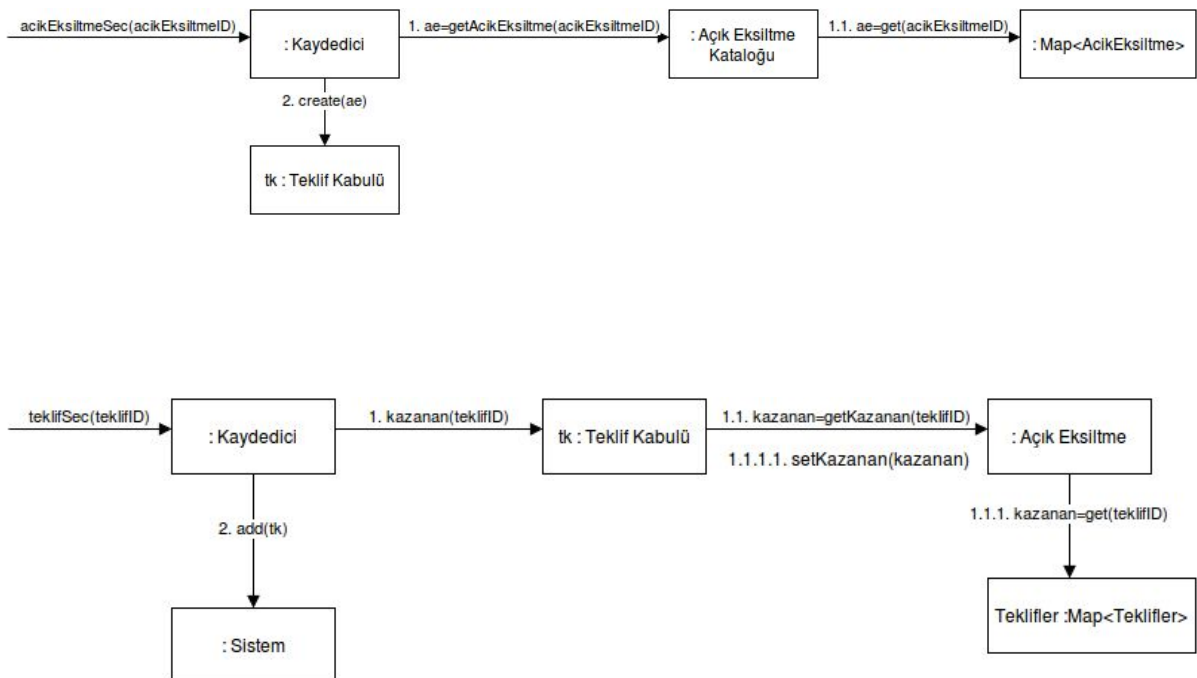
Use Case 1 Interaction Diagram



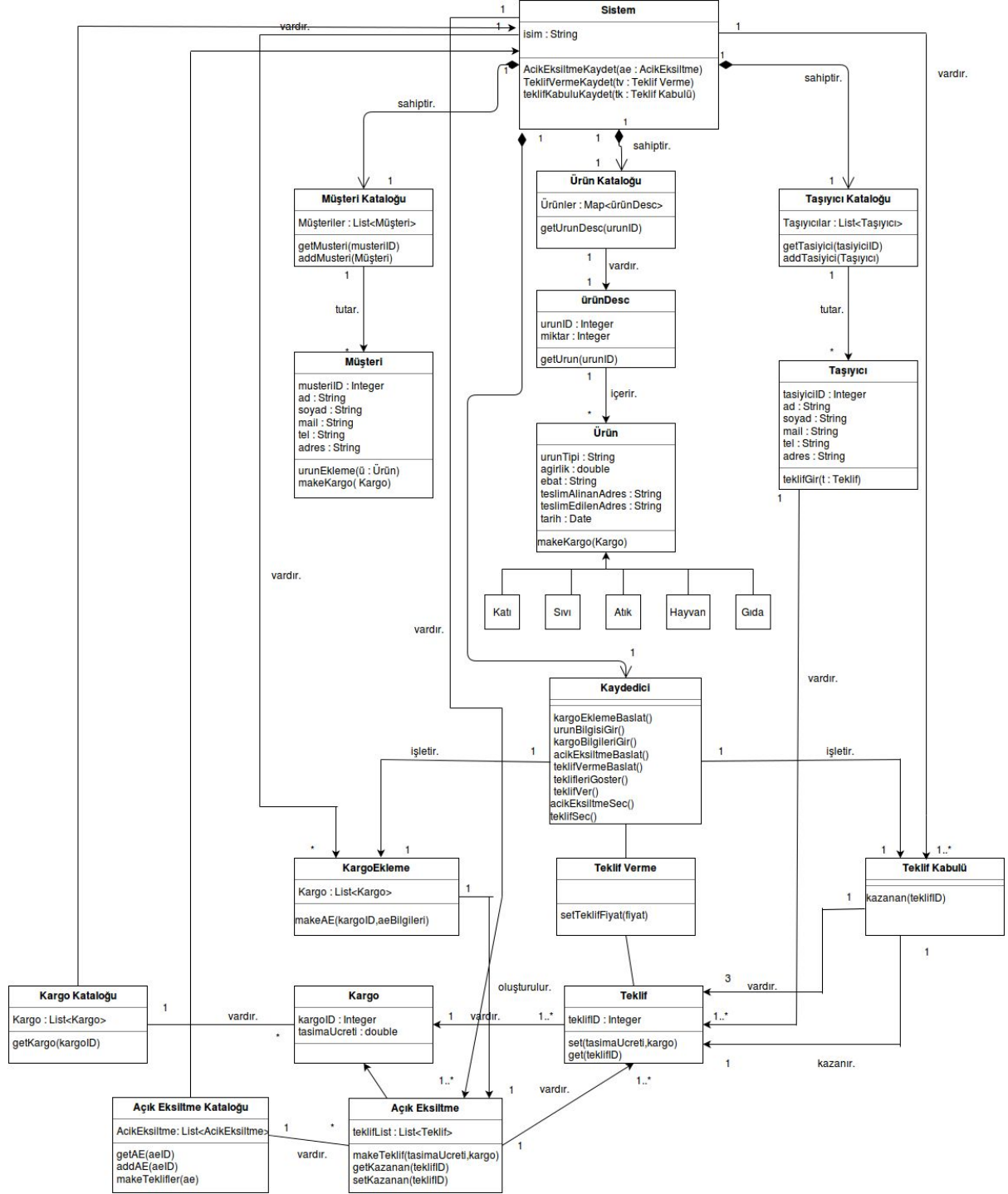
Use Case 2 Interaction Diagram



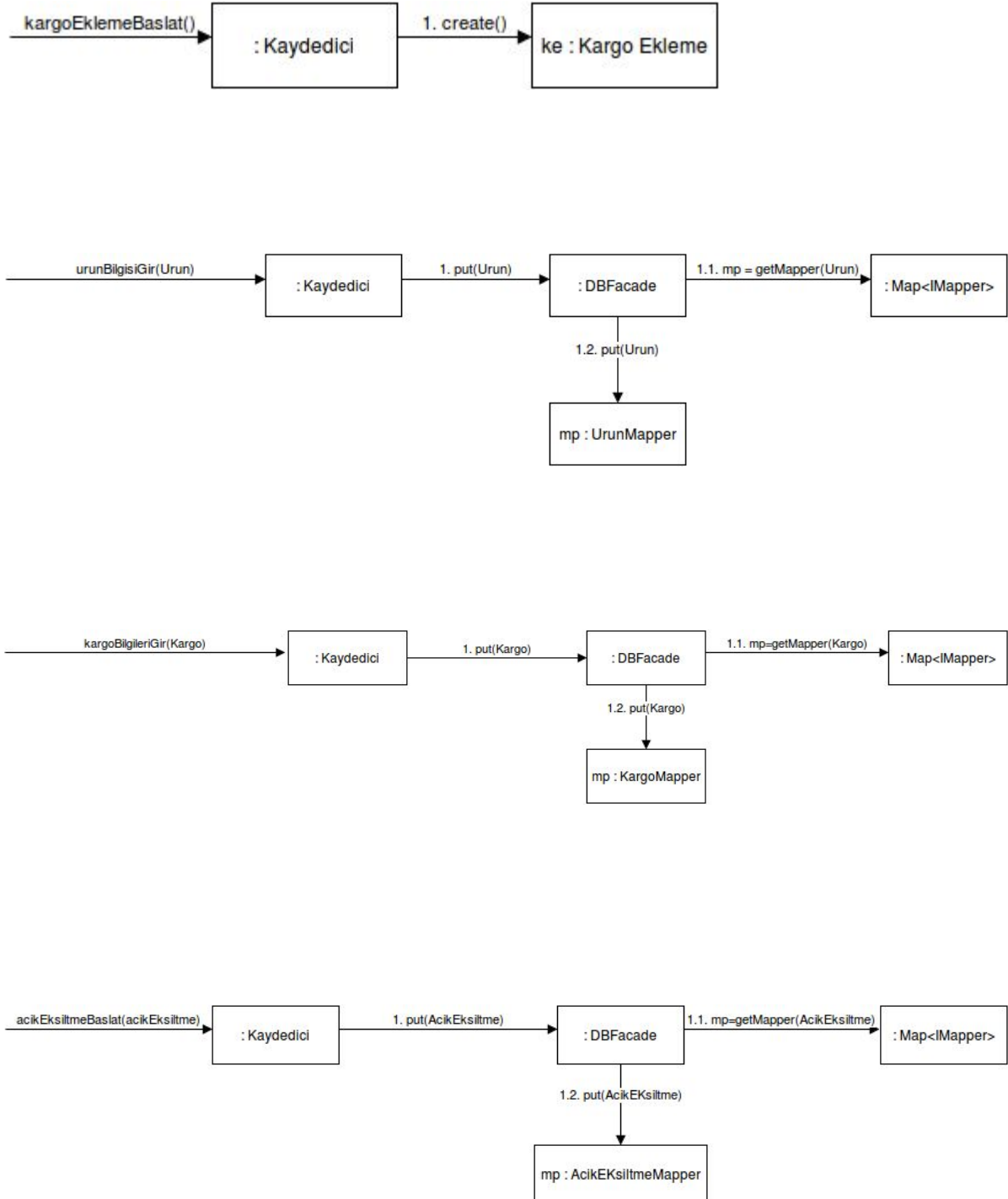
Use Case 3 Interaction Diagram



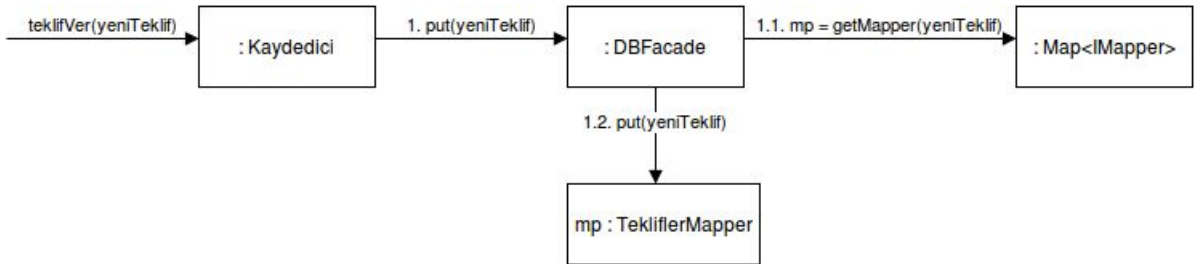
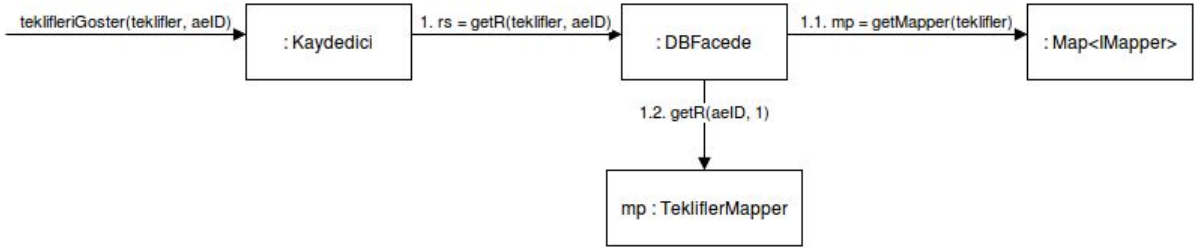
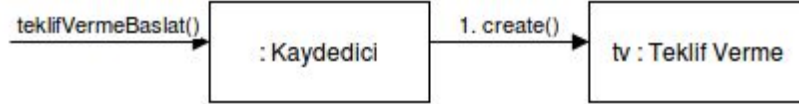
Class Diagram



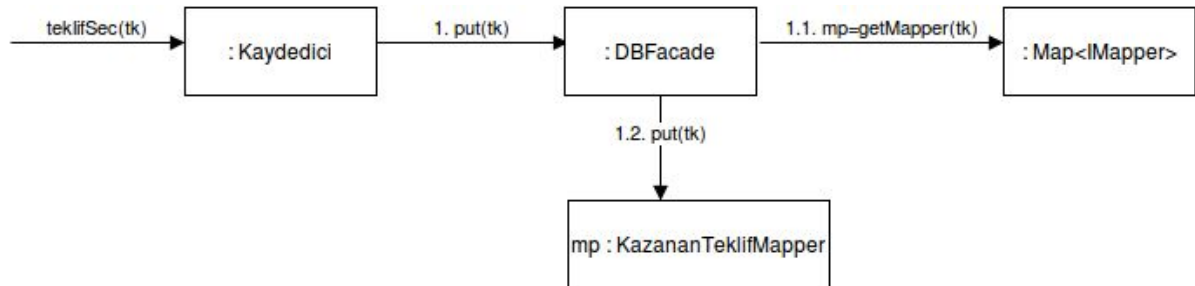
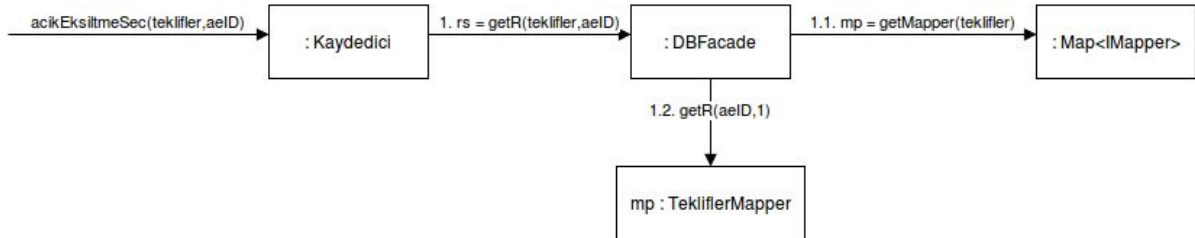
Use Case 1 Interaction Diagram



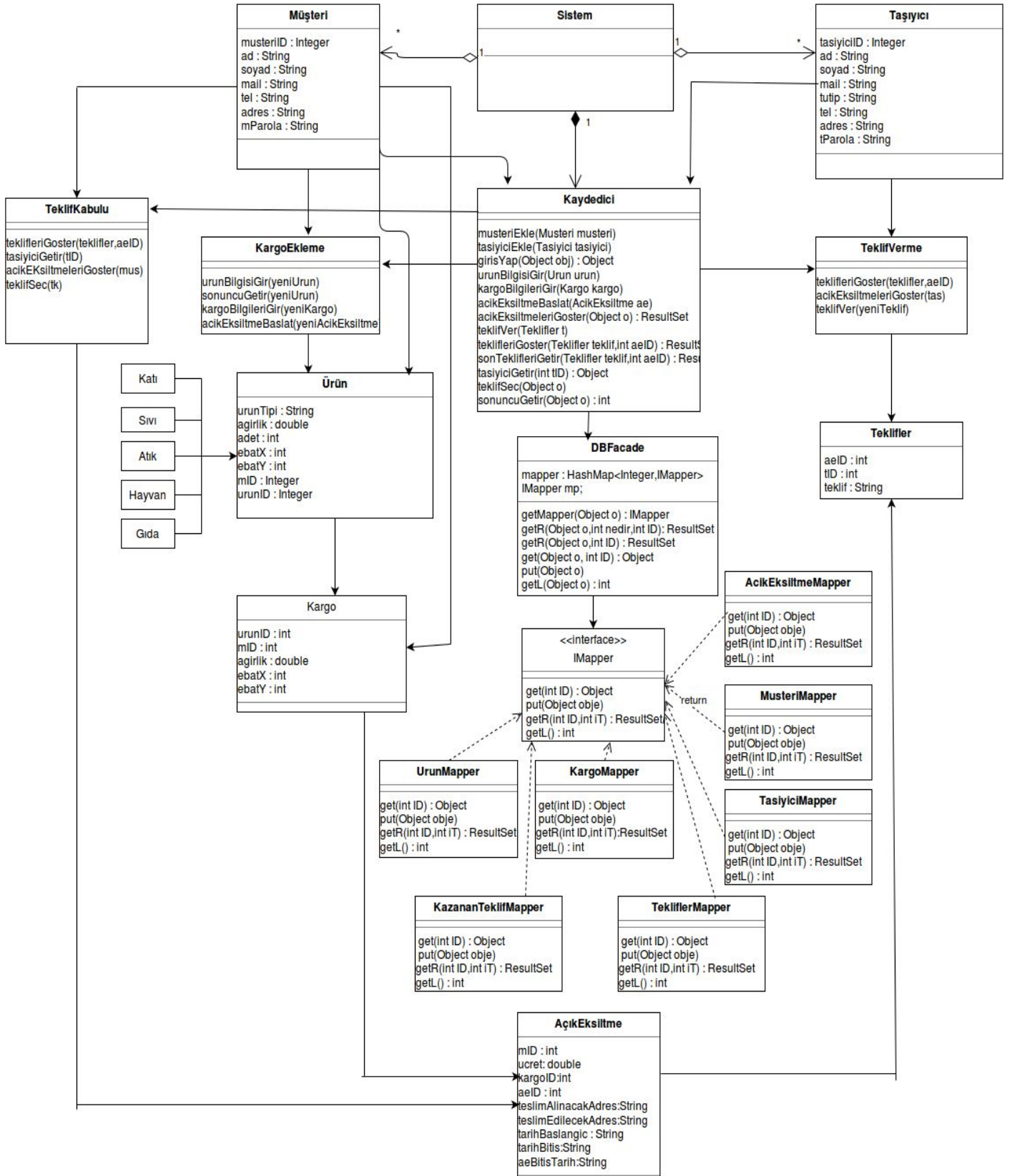
Use Case 2 Interaction Diagram



Use Case 3 Interaction Diagram



Class Diagram



TEST

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

import kargosistemi.AcikEksiltme;
import kargosistemi.Kargo;
import kargosistemi.Kaydedici;
import kargosistemi.Musteri;
import kargosistemi.Tasiyici;
import kargosistemi.Teklifler;
import kargosistemi.Urun;
import kargosistemi.teklifKabulu;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

/**
 *
 * @author hatice
 */
public class test {

    Kaydedici kaydedici = new Kaydedici();
    public test() {
    }

    @Test
    public void musterikaydetTest(){

        Musteri yeniMusteri = new Musteri("Hatice", "ERTÜRK",
"haticeerturk27@gmail.com", "05411111111", "Bursa", "1");
        kaydedici.musteriEkle(yeniMusteri);

        int mID = kaydedici.sonuncuGetir(yeniMusteri);
        assertEquals(21, mID);
    }

    @Test
    public void musterigiris(){
        Musteri musteris = new Musteri(21, "1");
        musteris = (Musteri)kaydedici.girisYap(musteris);
        int id = musteris.mID;
        String pass = musteris.pass;
        assertEquals(21, id);
        assertEquals("1", pass);
    }
}
```

```

@Test
public void tasiyiciKaydet(){
    Tasiyici yeniTasiyici = new Tasiyici("Mert", "TAŞÇI",
"merttasçi2@gmail.com", "05411111111", "İstanbul", "1", "Katı");
    kaydedici.tasiyiciEkle(yeniTasiyici);
    int tID = kaydedici.sonuncuGetir(yeniTasiyici);
    assertEquals(9, tID);
}

@Test
public void tasiyiciGiris(){
    Tasiyici tasiyici = new Tasiyici(9, "1");
    tasiyici = (Tasiyici)kaydedici.girisYap(tasiyici);
    int id = tasiyici.tID;
    String pass = tasiyici.pass;
    assertEquals(9, id);
    assertEquals("1", pass);
}

@Test
public void kargoEklemeOlayi(){
    Urun yeniUrun = new Urun(21, "Katı", 50.0, 5, 10, 5);
    kaydedici.urunBilgisiGir(yeniUrun);
    int uID = kaydedici.sonuncuGetir(yeniUrun);
    assertEquals(38, uID);

    Kargo yeniKargo = new Kargo(21, uID, 250.0, 5, 50);
    kaydedici.kargiBilgileriGir(yeniKargo);
    int kID = kaydedici.sonuncuGetir(yeniKargo);
    assertEquals(21, kID);

    AcikEksiltme yeniAcikEksiltme = new AcikEksiltme(21, kID, 5000, "Bursa",
"Ankara", "26-05-2016", "30-05-2016", "24-05-2016");
    kaydedici.acikEksiltmeBaslat(yeniAcikEksiltme);
    int aeID = kaydedici.sonuncuGetir(yeniAcikEksiltme);
    assertEquals(18, aeID);
}

@Test
public void teklifVerme(){
    Teklifler yeniTeklif = new Teklifler(18, 9, 4500.0);
    kaydedici.teklifVer(yeniTeklif);
    int ID = kaydedici.sonuncuGetir(yeniTeklif);
    assertEquals(16, ID);
}

@Test
public void teklifKabulu(){
    teklifKabulu teklif = new teklifKabulu(18, 16);
    kaydedici.teklifSec(teklif);
}

```

```

    int ID = kaydedici.sonuncuGetir(teklif);
    assertEquals(5, ID);
}

@BeforeClass
public static void setUpClass() {
}

@AfterClass
public static void tearDownClass() {
}

@Before
public void setUp() {
}

@After
public void tearDown() {
}

// TODO add test methods here.
// The methods must be annotated with annotation @Test. For example:
//
// @Test
// public void hello() {}
}

```