T.C. Çanakkale Onsekiz Mart Üniversitesi

Bilgisayar Mühendisliği Bölümü

****

**Nesneye Yönelik Analiz ve Tasarım Dersi**

**Proje Konusu** *Lokanta Sipariş Sistemi*

**Proje Grubu** *140401029 Berkay DEMİREL*

*120401031 Ali Osman AY*

**Dersin Danışman Hocası** *Yrd. Doç.Dr. Ali Murat TİRYAKİ*

İçindekiler

**Vizyon3**

**Aktor Goal Model6**

**Use Case Model7**

**Use Case Text8**

**Domain Model**13

**SSD Diagrams**1**6**

**Operation Contracts**19

**State Machine Diagrams22**

**Interaction Diagrams25**

**UML Class Diagrams28**

**VİZYON**

**Amaçlar**

* Müşterilere kişileştirilmiş hizmet sunmak
* Müşterilere ve işletme çalışanlarına zaman kazandırmak
* Lokanta sahiplerine daha kolay yönetilebilirlik sağlanmak
* Sağlığına dikkat eden müşteriler için özel hizmet sunmak
* Müşterilere lokanta, ürünler veya özel durumlar(parti,maç yayını vb.) hakkında bilgilendirme yapmak
* Müşterilerin sisteme doğrudan istek,öneri ve şikayetlerini girmesiyle hataların çok kısa sürede düzeltilmesini sağlamak
* Hesap hatalarını ortadan kaldırarak gereksiz masrafın önlenmesini sağlamak

Lokantalarda garson ile müşteri arasındaki iletişimi dijital ortama taşımayı amaçlayan bir sistemdir.Bu sistem temelde, müşterilerin garson ile iletişime geçerek yaptıkları sipariş verme, hesap isteme, menü isteyerek ürünlere bakabilme veya çeşitli isteklerini dile getirme işlemlerini elektronik bir tablet yardımıyla yapmasını sağlamaktır.Ayrıca bu sistem geleneksek garson yönteminden farklı olarak müşteriye özel menüler hazırlayabilme, sürekli lokantaya gelen müşterilere indirim yapabilme, müşterilerin daha önce neler sipariş ettiğini analiz ederek onlara kişileştirilmiş indirimler sunabilme gibi imkanını da lokanta yetkililerine sağlıyor.

Lokantaya giriş yapan bir müşteri masaya oturduktan sonra masada bulunan tablet ile üye girişi yapabiliyor veya daha önce sisteme üye olmamış ise hızlı bir şekilde üye olabiliyor.Ayrıca sisteme üye olmak istemeyen kullanıcılarda tek bir tık ile devam edebiliyor ancak kişileştirilmiş hizmetlerde yararlanamıyorlar.

Sisteme üye olarak giriş yapan bir kullanıcı için sistem daha önce sipariş ettiği ürünleri analiz ederek müşterinin sipariş vermesi muhtemel ürünleri ,daha önceki siparişlerinden dolayı kazandığı indirimleri ve indirimli ürünleri reklam tarzında gösteriyor. Sisteme üye girişi yapmadan devam eden bir kullanıcı bu kişileştirilmiş fırsatlardan yararlanamıyor.

Kullanıcılar arayüzde bulunan menü seçeneğine tıklayarak menü kategorilerinden birini seçerek o kategoriye özel menüleri görüntülüyor. Bu kategoriler lokanta sahibinin oluşturduğu menüleri istediği kategoriye göre düzenlemesiyle oluşuyor.Kategoriler normal menüler, vejetaryen menüler, sporcu menüleri, tek ürünler, sağlık problemleri olan insanlar için oluşturulmuş menüler(Şeker hastaları için olan menüler, kolesterolü olan hastalar için menüler vb.) gibi lokanta sahibinin belirleyeceği şekilde olabilir. Menü kategorilerini üye olmadan devam eden kullanıcılarda görüntüleyebilir ancak üye olan kullanıcılar için kaydettiğim menüler kategorisi olacaktır. Bu kategori üye olan kullanıcıların daha önce oluşturdukları ve kaydettikleri menülerin bulunduğu kategoridir.

Üye olarak veya üye olmadan sisteme giriş yapan kullanıcılar menüler kısmına girdiğinde isterse daha önceden lokanta sahibinin belirlediği menüleri sipariş edebilir isterse de tekli ürünlerden sipariş verebilir. Ayrıca kullanıcılar isterse tekli ürünleri kombine ederek kendi menülerini oluşturabilirler. Bu durumda sistem lokanta sahibinin belirlediği indirimleri (Yemeğin yanında sipariş edilen tatlının tek alım fiyatına göre 1 tl indirimli olması gibi) otomatik olarak uygular ve kullanıcıya gösterir. Bu kısımda üye olan kullanıcılar için kişileştirilmiş hizmet olarak oluşturdukları menüleri kaydedebilme imkanı sunulur. Ayrıca üye olan kullanıcılar için sistem daha önceki siparişlerini analiz ederek ekstra indirim uygulayabilir.

Menüler ekranında ürünleri görüntüleyen kullanıcılar ürünlerin içindeki malzemeleri, kalori değerlerini ve seçtikleri tüm ürünlerin toplam kalori değerlerini görebilecekler. Ayrıca kullanıcılar istemedikleri malzemeleri ürünlerin içerisinden çıkarabilecek veya istedikleri malzemelerden daha fazla olmasını isteyebileceklerdir. Bu sayede sağlığına dikkat eden müşteriler kendilerine özel menüleri daha kolay oluşturabileceklerdir.

Kullanıcılar arayüzde bulunan istek, öneri ve şikayet bölümüne girerek lokanta sahibine direkt olarak mesaj gönderebileceklerdir. Yazılan istek ve öneriler sayesinde lokanta sahipleri istekler doğrultusunda işletmelerini geliştirebileceklerdir. Çalışanlar vasıtasıyla değil de sistem ile doğrudan lokanta sahiplerine ulaştırılan şikayetler ile de lokanta sahipleri müşterilerin sıkıntılarını daha kolay çözebileceklerdir.

Sistem ile doğrudan sorununu çözmeyen veya istediğini yapamayan müşteriler arayüzde bulunan garsonu çağır butonu ile garsonu çağırıp yüz yüze konuşabilme imkanını bulacaklardır.

Bu sistem sayesinde wi-fi şifresi, lokanta hakkında ki bilgiler, ürünlerin açıklaması, varsa diğer şubelerin bilgileri, yeni çıkan ürünlerin tanıtımı, varsa maç yayını ve ne zaman olacağı, maç yayınının ücretleri, işletmede yapılacak özel günlerin tanıtımı ve ücretleri(parti,toplantı vb.), özel günler için(düğün,toplantı vb.) lokantadan verilebilecek toplu siparişler ve ücretleri gibi bilgilendirmeler kolaylıkla yapılabilecektir.

Masasından memnun olmayan değiştirmek isteyen kullanıcılar (içerde oturup dışarıda sigara içilen bölüme geçmek isteyenler veya dışarıda oturup içeride masanın boşalmasını bekleyen müşteriler) tek tık ile bunu bildirebilecekler. Uygun bir masa boşaldığında ise garson müşterilere bunu bildirip sistemde çok kolay bir şekilde hesabı ve kullanıcıları diğer masaya taşıyabilecektir.

Aynı masada oturan birden çok kullanıcı olabileceği için aynı anda en fazla 4 kişi olacak şekilde ekran dörde bölünerek her bir kullanıcının sisteme girişi sağlanabilecektir. Dörtten daha fazla kullanıcı aynı anda sisteme giriş yapmak isterse arayüzden garsonu çağırıp ayrı bir tablet isteyerek sisteme giriş yapabilecektir.Yine sistemde bu kullanıcılar aynı masada gözükecek ve hesap istendiğinde garson her kullanıcıdan ayrı ayrı hesabı alarak masanın toplam hesabını almış olacaktır.

Müşteri siparişini verdikten sonra arayüz sayesinde siparişinin durumunu anlık takip edebilecektir. Ayrıca müşteriye sisteme sipariş hazırlanmaya başladı durumu girilmeden siparişini iptal edebilme veya değiştirebilme hakkı da tanınacaktır.

Müşteri hesabı istediğinde sistem kişileştirilmiş hesabı garsona bildirecek ve garson hesabı masaya götürecektir. Müşteri masadan kalktıktan sonra garsonlar sistemden masayı sıfırlayıp yeni müşteriler için hazır hale getirebileceklerdir.

Sistemin yöneticileri lokanta sahipleri olacaktır. Onlar için ayrı tanımlanan hesaplar ile sisteme giriş yapabileceklerdir. Lokanta sahipleri sisteme giriş yaparak hasılatı ve hangi üründen ne kadar sipariş edildiğini günlük, haftalık, aylık olarak görüntüleyebileceklerdir. Lokantadaki masa durumlarını ve kullanıcılarını da görüntüleyebileceklerdir. Sistemdeki ürünleri ,ürün bilgilerini, menüleri ve menü kategorilerini lokanta sahipleri sisteme girebileceklerdir. Ayrıca lokanta sahipleri müşterilerin gönderdikleri istek, öneri ve şikayetleri görüntüleyebilecekler ve sistemde garson ve şefler için hesap oluşturabileceklerdir.

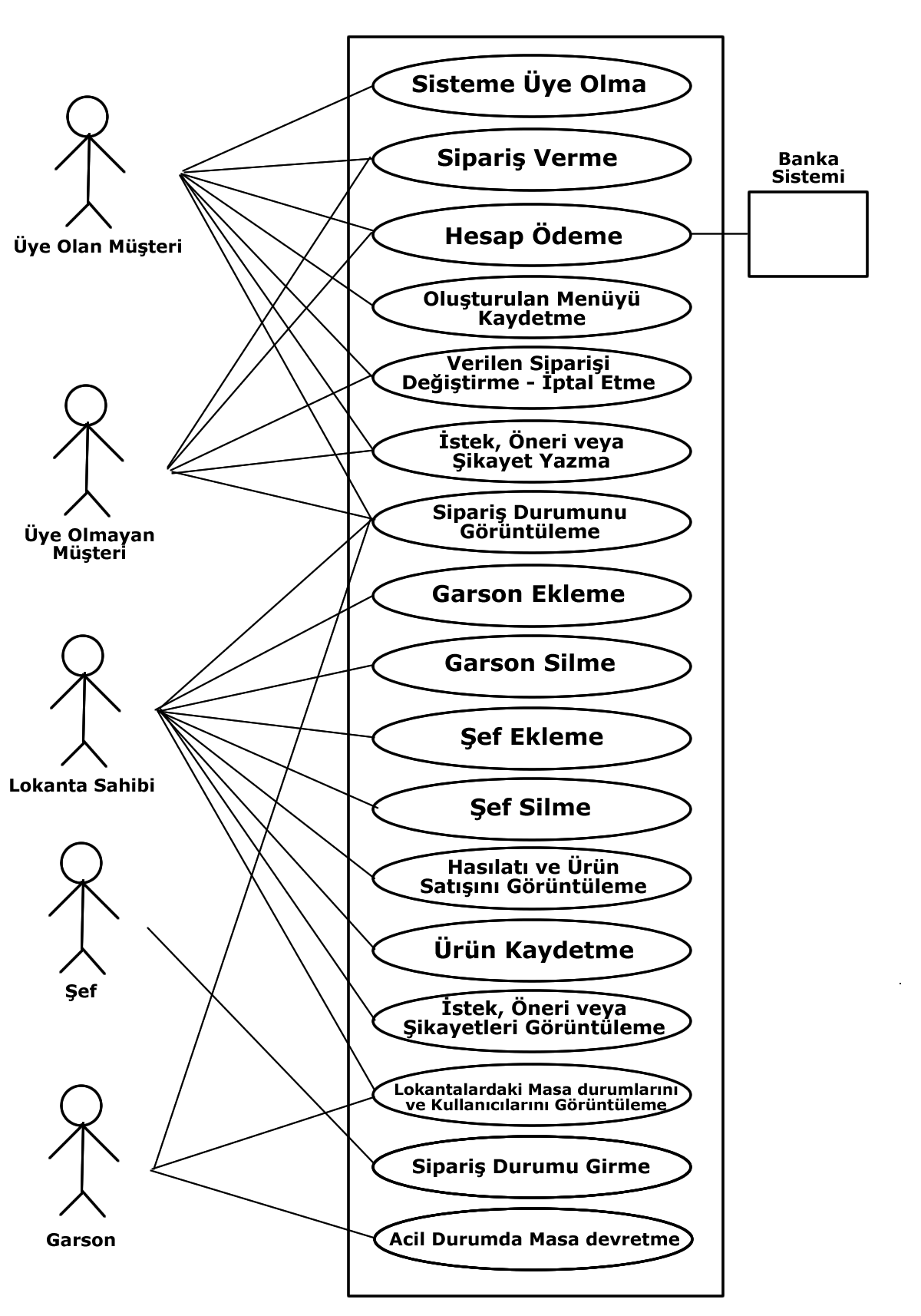
Sistemde garsonlar için ayrı tanımlanmış hesaplar ile garsonlar sisteme giriş yapabileceklerdir. Garsonlar sisteme giriş yaptıklarında lokantada bulunan masaları ve masalardaki kullanıcıları görüntüleyebileceklerdir. Sistem her masaya otomatik olarak aktif olan bir garsonu atayacaktır. Masadan bir istek geldiğinde masaya bakan garson sistem tarafından uyarılacaktır. Ayrıca şefler siparişi hazırladıklarında sipariş hangi masaya ait ise sistem o garsonu uyaracak ve siparişi teslim etmesini söyleyecektir. Bir müşteri hesabı istediği zaman sistem o masaya ait olan garsonu uyaracaktır ve garson hesabı aldıktan sonra sistem masayı otomatik olarak sıfırlayacaktır. Sistemden acil çıkış yapması gereken garsonlar masasını başka bir garsona devredebilecektir.

Sistemde şefler için ayrı tanımlanmış hesaplar ile şefler sisteme giriş yapabilecektir. Şefler ekranlarında gelen siparişleri ve ürünlerin içeriklerini (İçerisinde ekstra istenen malzemeler ve istenmeyen malzemeler) görüntüleyeceklerdir. Ayrıca şefler hazırladıkları siparişlerin durumunu sisteme gireceklerdir.

**ACTOR GOAL MODEL**

| **Actor** | **Goal** |
| --- | --- |
| Üye Olan Müşteri | Sisteme Kayıt Olma  Sipariş Verme  Hesap Ödeme  Oluşturulan Menüyü Kaydetme  Verilen Siparişi Değiştirme - İptal Etme  İstek, Öneri veya Şikayet Yazma  Sipariş Durumunu Görüntüleme |
| Üye Olmayan Müşteri | Sipariş Verme  Hesap Ödeme  Verilen Siparişi Değiştirme – İptal Etme  İstek, Öneri veya Şikayet Yazma  Sipariş Durumunu Görüntüleme |
| Lokanta Sahibi | Garson Ekleme  Garson Silme  Şef Ekleme  Şef Silme  Hasılatı ve Ürün Satışını Görüntüleme  Ürün Kaydetme  İstek, Öneri veya Şikayetleri Görüntüleme  Lokantalardaki Masa durumlarını ve Kullanıcılarını Görüntüleme  Sipariş Durumunu Görüntüleme |
| Şef | Sipariş Durumu Girme |
| Garson | Acil Durumda Masa devretme  Lokantalardaki Masa durumlarını ve Kullanıcılarını Görüntüleme  Sipariş Durumunu Görüntüleme |

**USE CASE MODEL**



**USE CASE TEXT**

**Use Case UC1**

|  |  |
| --- | --- |
| **Use Case UC1:** | Ürün Kaydetme |
| **Scope:** | Lokanta Sipariş Sistemi |
| **Level:** | User Goal |
| **Primary Actor:** | Lokanta Sahibi |
| **Stakeholders and**  **Interests:** | **Lokanta Sahibi:** Gireceği bilgileri nereye gireceğini görmek ister. Sistemin hızlı olmasını ister. |
| **Preconditions:** | Lokanta sahibi sisteme kayıt edilmiş olmalı ve sisteme giriş yapmış olmalıdır. |
| **Success Guarantee:** | Lokanta sahibi bilgileri sisteme başarıyla girebilmiş olmalıdır.  Müşteri ürünleri görüntüleyebilmelidir. |
| **Main Success Scenario:** | 1. Lokanta Sahibi sisteme bilgi girişi isteği yapar.  2. Sistem ürün bilgisi giriş sayfasını görüntüler.  3. Lokanta Sahibi ürün bilgilerini sisteme girer.  4. Sistem kaydedilecek ürünleri ve girilmiş bilgilerini görüntüler.  5. Lokanta sahibi bilgi girişi işlemini sonlandırır.  6. Sistem ürün bilgilerini kaydeder. |
| **Extansions:** | \*a. Sistem Çökerse:  1. Lokanta sahibi sistemi yeniden başlatır.  1a. Bilgi girişi isteği başlatılamazsa:  1. Lokanta sahibi bilgi girişi isteğini yeniden yapar.  6a. Sistem ürün bilgilerini kaydetmezse:  1. Lokanta sahibi aynı bilgileri 1. adımdan başlayarak tekrar sisteme girer. |
| **Special Requirements:** | Sistemin kullanabileceği bilgisayar.  Aktörlerin sistemi kullanabilmesi için tabletler. |
| **Frequency of** | Az sıklıkla gerçekleşir. |
| **Open Issues:** | -Lokanta sahibi sistemi kullanabilir mi?  -Girilecek ürünlerin tüm bilgileri öğrenilmiş mi? |

**Use Case UC2**

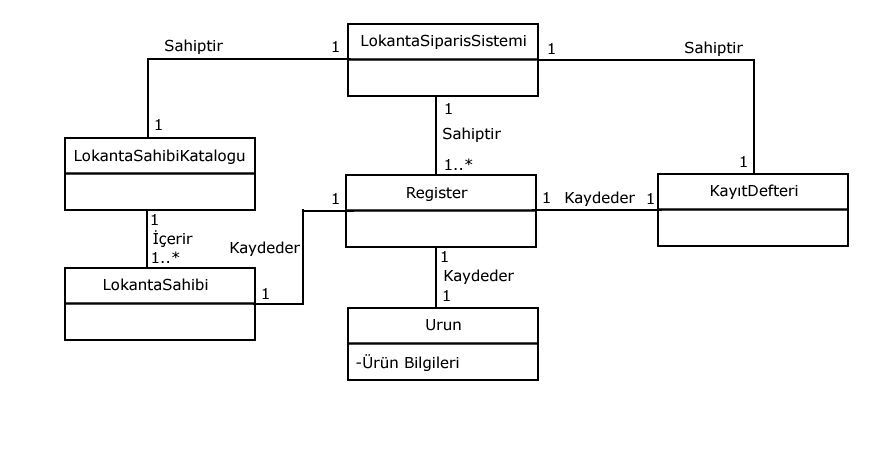
|  |  |
| --- | --- |
| **Use Case UC2:** | Sipariş Verme |
| **Scope:** | Lokanta Sipariş Sistemi |
| **Level:** | User Goal |
| **Primary Actor:** | Üye Olan Müşteri, Üye Olmayan Müşteri |
| **Stakeholders and**  **Interests:** | **Müşteri:** Sipariş vermek istediği ürünlerin ve bilgilerinin sisteme giriş yapılmış olmasını ister.  **Şef:** Müşterinin verdiği siparişin ekranında görüntülenmesini ister.  **Garson:** Müşterinin verdiği siparişin ekranında görüntülenmesini ister.  **Lokanta Sahibi:** Müşterinin verdiği siparişin ekranında görüntülenmesini ister. |
| **Preconditions:** | Müşteri sisteme giriş yapmış olmalıdır.  Lokanta sahibi sisteme ürün girişi yapmış olmalıdır.  Sisteme giriş yapmış en az bir garson olmalıdır.  Sisteme giriş yapmış en az bir şef olmalıdır. |
| **Success Guarantee:** | Müşteri istediği ürünleri seçebilmiş ve başarıyla siparişini verebilmiş olmalıdır.  Sisteme giriş yapmış lokanta sahipleri, garsonlar ve şefler siparişi görüntüleyebilmelidir. |
| **Main Success Scenario:** | 1. Müşteri sisteme sipariş verme isteği yapar.  2. Sistem ürünleri listeler.  3. Müşteri istediği ürünü seçer.  2. ve 3. adımlar müşterinin almak istediği ürün sayısı kadar tekrar eder.  4. Müşteri ürün seçimi bitirme isteği yapar.  5. Sistem seçilen ürün bilgilerini ve toplam fiyatı gösterir.  6. Müşteri siparişi tamamlama isteği yapar.  7. Sistem sipariş bilgilerini kaydeder. |
| **Extansions:** | \*a. Sistem Çökerse:  1. Lokanta sahibi sistemi yeniden başlatır.  \*b. Sistemden masaya ulaşılamazsa:  1. Garsonlar masayı sıfırlar.  1a. Sipariş verme isteği başlatılamazsa:  1. Müşteri sipariş verme isteğini yeniden yapar.  6a. Sipariş tamamlama isteği yapılamazsa:  1. Müşteri sipariş verme isteğini yeniden yapar.  7a. Sistem sipariş bilgilerini kaydedemezse:  1. Lokanta sahibi aynı bilgileri 1. adımdan başlayarak sisteme girer. |
| **Special Requirements:** | Sistemin kullanabileceği bilgisayar.  Aktörlerin sistemi kullanabilmesi için tabletler. |
| **Technology and Data**  **Variations List:** | - |
| **Frequency of**  **Occurrence:** | Sürekli gerçekleşir. |
| **Open Issues:** | -Sipariş hazır olduğunda garson sistemin başında olacak mı?  -Şefler siparişi müşterinin istediği gibi hazırlayabilecek mi?  -Müşteriler sistemde olmayan özel bir ürün isterse ne yapılacak? |

**Use Case UC3**

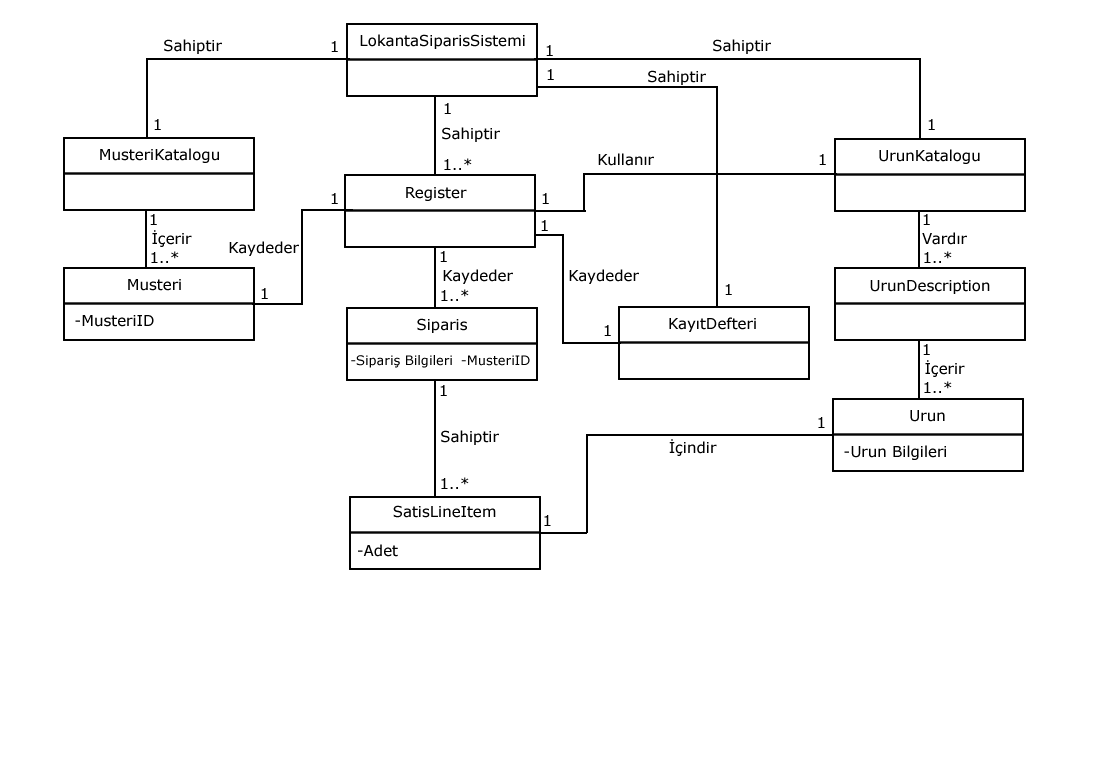
|  |  |
| --- | --- |
| **Use Case UC3:** | Hesap Ödeme |
| **Scope:** | Lokanta Sipariş Sistemi |
| **Level:** | User Goal |
| **Primary Actor:** | Üye Olan Müşteri, Üye Olmayan Müşteri |
| **Stakeholders and**  **Interests:** | **Müşteri:** Hesabı görüntüleme ve masadan kalkacağı zaman hesabı ödeme yetkisini ister.  **Garson:** Müşteri hesabı istediğinde sistem tarafından uyarılarak hesabın ne kadar olduğunu görüntülemek ister.  **Lokanta Sahibi:** Müşterinin sipariş durumunu ve ödeyeceği hesabı görüntülemek ister. |
| **Preconditions:** | Müşteri sisteme giriş yapmış olmalıdır.  Müşteri sipariş vermiş olmalıdır.  Müşterinin en az bir siparişi teslim edilmiş olmalıdır.  Müşterinin teslim edilmemiş siparişi olmamalıdır.  Sisteme giriş yapmış en az bir garson olmalıdır. |
| **Success Guarantee:** | Müşteri hesabını ödeyebilmiş olmalıdır.  Hesap bilgileri sisteme kaydedilmiş olmalıdır. |
| **Main Success Scenario:** | 1. Müşteri sisteme hesap ödeme isteği yapar.  2. Sistem müşteriye hesabını görüntüler.  3. Sistem ilgili garsonu uyararak hesabı görüntüler.  4. Garson ödeme işlemini gerçekleştirir.  5. Sistem işlemi onaylar.  6. Sistem fatura çıktısını verir ve işlemi kaydeder. |
| **Extansions:** | 1a. Hesap ödeme isteği başlatılamazsa:  1. Müşteri hesap ödeme isteğini yeniden yapar.  2a. Sistem müşteriye hesabı görüntülemezse:  1. Müşteri hesap ödeme isteğini yeniden yapar.  2. Sistem müşteriye hesabı görüntüler.  3a. Sistem ilgili garsonu uyararak hesabı görüntülemezse:  1. Müşteri hesap ödeme isteğini başlattıktan belli bir süre sonra garson ödeme işlemini gerçekleştirmediyse sistem tekrardan garsonu uyarır.  4a. Müşteri kredi kartı ile ödeme isterse:  1. Garson sisteme "kredi kartı ile ödeme" olarak girer.  2. Garson müşterinin kredi kartını alır ve ödeme işlemini gerçekleştirir.  3. Ana senaryo 5. adımdan itibaren devam eder.  5a. Sistem işlemi onaylamazsa:  1. Müşteri hesap ödeme isteğini tekrar yapar.  6a. Sistem işlemi kaydetmezse:  1. Lokanta sahibi aynı bilgileri 1. adımdan başlayarak sisteme girer. |
| **Special Requirements:** | Sistemin kullanabileceği bilgisayar.  Aktörlerin sistemi kullanabilmesi için tabletler. |
| **Technology and Data**  **Variations List:** | - |
| **Frequency of**  **Occurrence:** | Sürekli gerçekleşir. |
| **Open Issues:** | -Garson post cihazını kullanabilir mi?  -Nakit ödeme yapıldığında para güvenliği için ne yapılmalı? |

**DOMAIN MODEL**

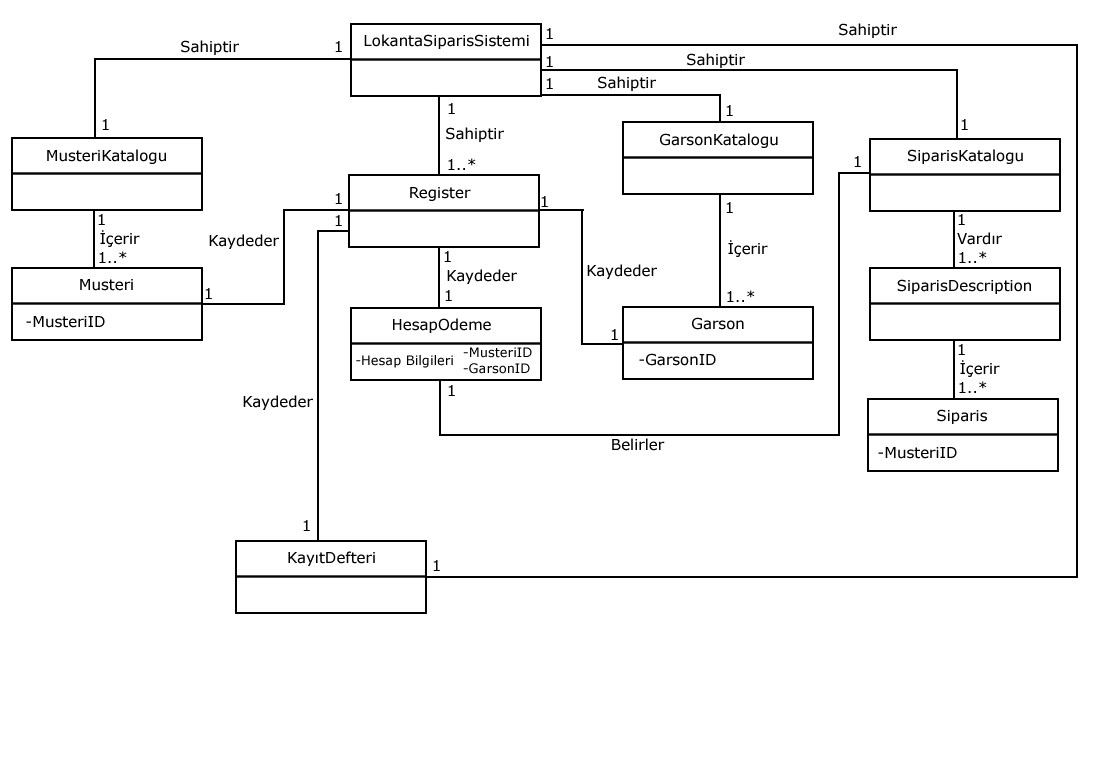
**Use Case 1 Domain Model**

****

**Use Case 2 Domain Model**

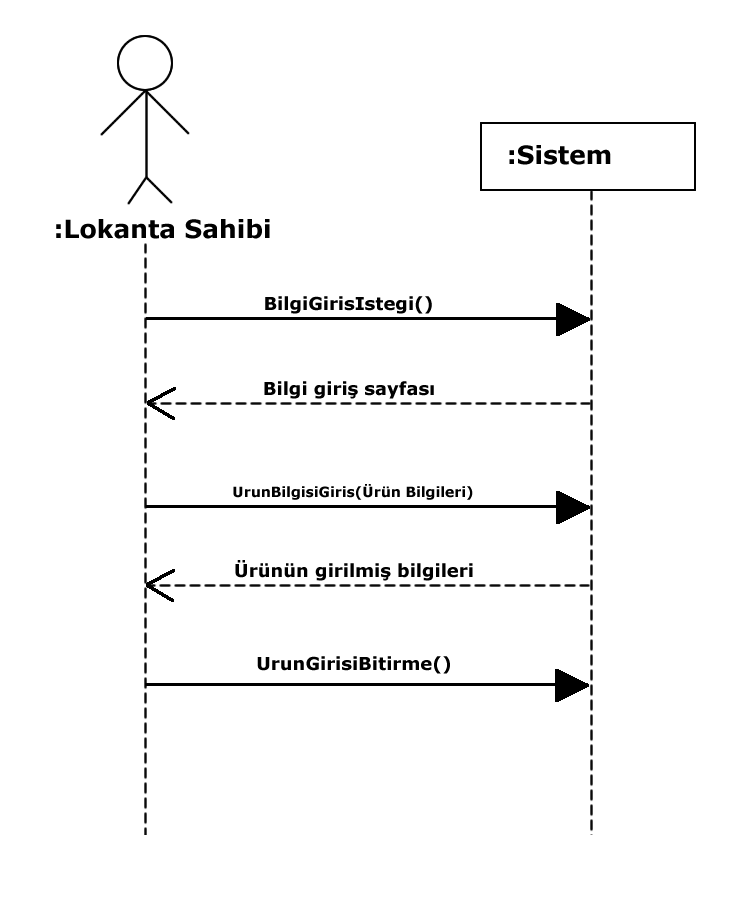
****

**Use Case 3 Domain Model**

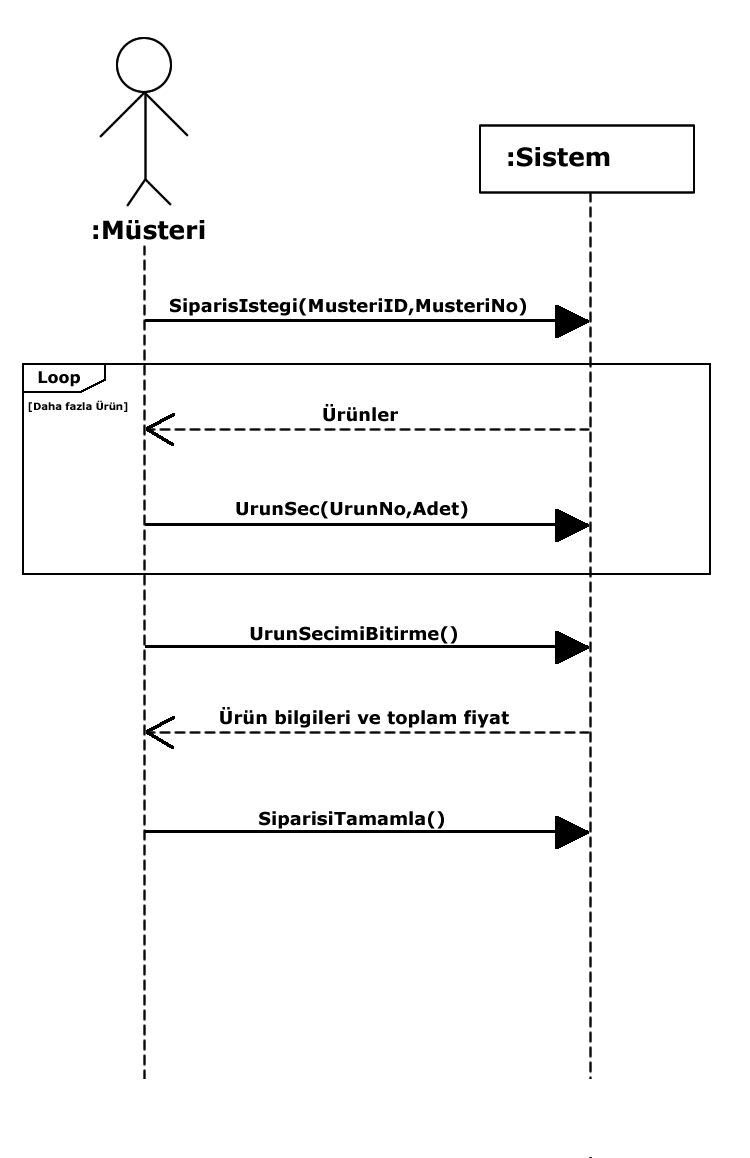
****

**SSD DIAGRAMS**

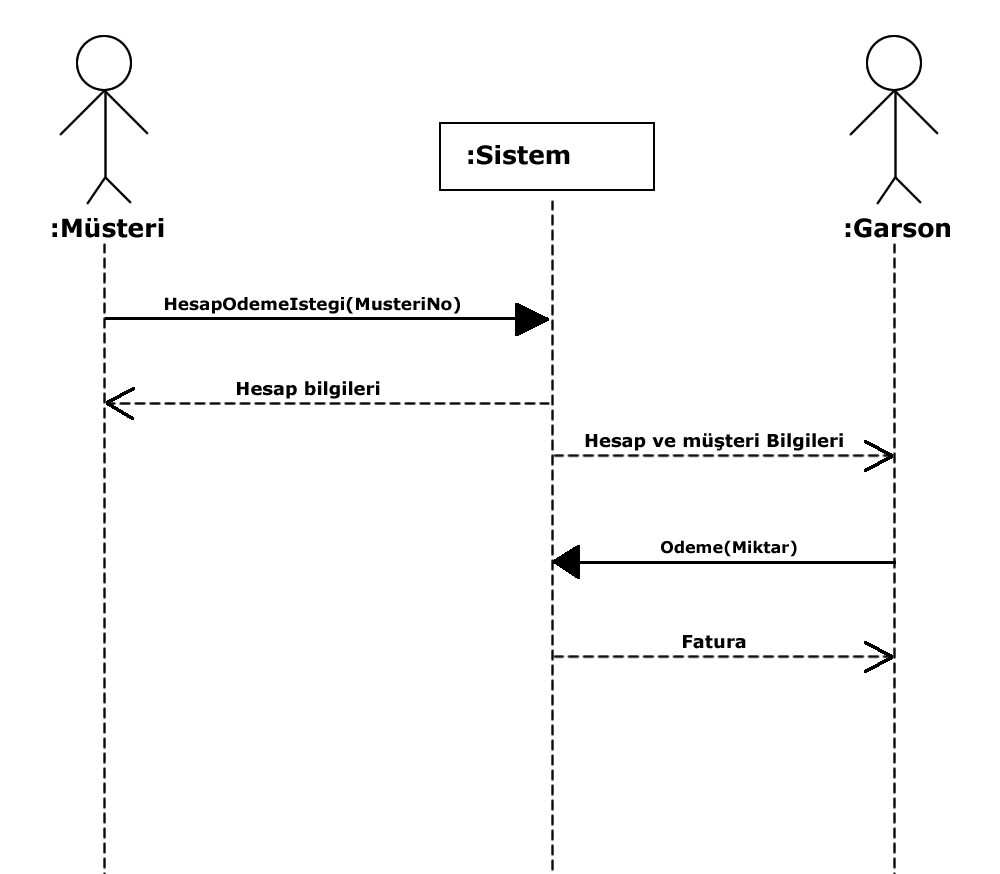
**Use Case 1 SSD Diagram**

****

**Use Case 2 SSD Diagram**

****

**Use Case 3 SSD Diagram**

****

**OPERATİON CONTRACTS**

**Ürün Kaydetme**

**Contract OC1**

**Name:** BilgiGirisIstegi()

**Cross References:** Use Cases: Ürün Kaydetme

**Precondition:** none

**Postcondition:**

* Urun sınıfın YeniUrun isminde bir nesnesi oluşturulmuş olmalıdır.
* YeniUrun nesnesi Register nesnesi ile ilişkilendirilmiş olmalıdır.

**Contract OC2**

**Name:** UrunBilgisiGiris(Ürün Bilgileri)

**Cross References:** Use Cases: Ürün Kaydetme

**Precondition:** Devam eden bir Urun nesnesi var olmalıdır.

**Postcondition:**

* Devam etmekte olan Urun nesnesinin Ürün Bilgileri özelliği parametre olarak gelen Ürün Bilgileri ile set edilmiş olmalıdır.

**Contract OC3**

**Name:** UrunGirisiBitirme()

**Cross References:** Use Cases: Ürün Kaydetme

**Precondition:**

* Devam eden bir Urun nesnesi var olmalıdır.

**Postcondition:**

* Devam etmekte olan Urun nesnesi Kayıt Defteri nesnesi ile ilişkilendirilmiş olmalıdır.

**Sipariş Verme**

**Contract OC1**

**Name:** SiparisIstegi(MüsteriID,MusteriNo)

**Cross References:** Use Cases: Sipariş Verme

**Precondition:** none

**Postcondition:**

* Siparis sınıfının YeniSiparis isminde bir nesnesi oluşturulmuş olmalıdır.
* YeniSiparis nesnesi Register nesnesi ile ilişkilendirilmiş olmalıdır.
* YeniSiparis nesnesinin özellikleri başlatılmış olmalıdır.
* YeniSiparis nesnesinin MüsteriID özelliği parametre olarak gelen MüsteriID ile set edilmiş olmalıdır.
* YeniSiparis nesnesinin MüsteriNo özelliği parametre olarak gelen MüsteriNo ile set edilmiş olmalıdır.

**Contract OC2**

**Name:** UrunSec(UrunNo,Adet)

**Cross References:** Use Cases: Sipariş Verme

**Precondition:** Devam eden bir Siparis nesnesi var olmalıdır.

**Postcondition:**

* SatisLineItem sınıfının sli isminde bir nesnesi oluşturulmuş olmalıdır.
* sli nesnesi devam etmekte olan Siparis nesnesi ile ilişkilendirilmiş olmalıdır.
* sli nesnesi parametre olarak gelen UrunNo'nun belirttiği Urun nesnesi ile ilişkilendirilmiş olmalıdır.
* sli nesnesinin Adet özelliği parametre olarak gelen Adet ile set edilmiş olmalıdır.

**Contract OC3**

**Name:** UrunSecimiBitirme()

**Cross References:** Use Cases: Sipariş Verme

**Precondition:**

* Devam eden bir Siparis nesnesi var olmalıdır.
* Devam eden bir SatisLineItem nesnesi var olmalıdır.

**Postcondition:** none

**Contract OC4**

**Name:** SiparisiTamamla()

**Cross References:** Use Cases: Sipariş Verme

**Precondition:**

* Devam eden bir Siparis nesnesi var olmalıdır.
* Devam eden bir SatisLineItem nesnesi var olmalıdır.

**Postcondition:**

* Siparis nesnesi Kayıt Defteri nesnesi ile ilişkilendirilmiş olmalıdır.

**Hesap Ödeme**

**Contract OC1**

**Name:** HesapOdemeIstegi(MusteriNo)

**Cross References:** Use Cases: Hesap Ödeme

**Precondition:** none

**Postcondition:**

* HesapOde sınıfının hesapOde isminde bir nesnesi oluşturulmuş olmalıdır.
* hesapOde nesnesi Register nesnesi ile ilişkilendirilmiş olmalıdır.
* hesapOde nesnesinin MusteriNo özelliği parametre olarak gelen MusteriNo ile set edilmiş olmalıdır.
* hesapOde nesnesi parametre olarak gelen MusteriNo 'nin belirttiği Siparis nesnesi ile ilişkilendirilmiş olmalıdır.
* hesapOde nesnesinin özellikleri başlatılmış olmalıdır.

**Contract OC2**

**Name:** Odeme(Miktar)

**Cross References:** Use Cases: Hesap Ödeme

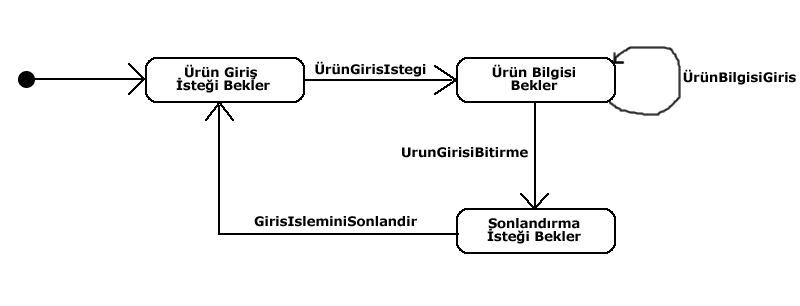
**Precondition:** Devam eden bir HesapOde nesnesi var olmalıdır.

**Postcondition:**

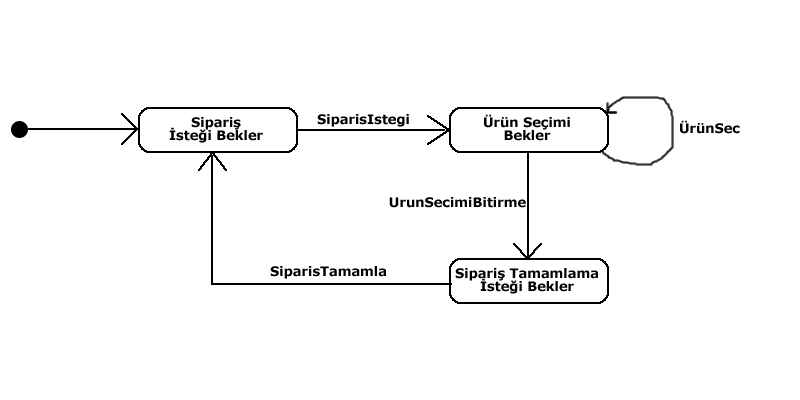
* HesapOde nesnesinin Miktar özelliği parametre olarak gelen Miktar ile set edilmiş olmalıdır.
* HesapOde nesnesi kayıt defteri nesnesi ile ilişkilendirilmiş olmalıdır.

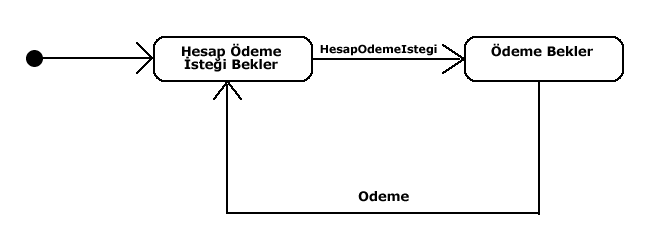
**STATE MACHINE DIAGRAM**

**Use Case 1 :**

****

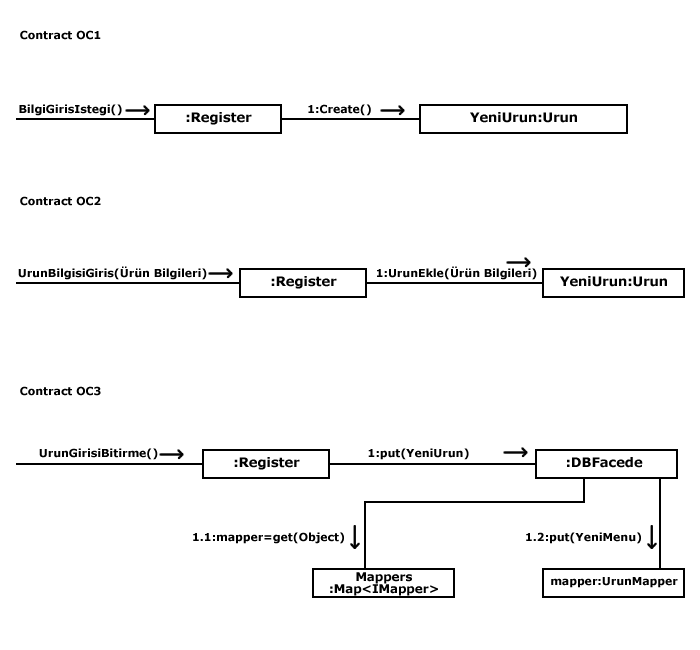
**Use Case 2:**

****

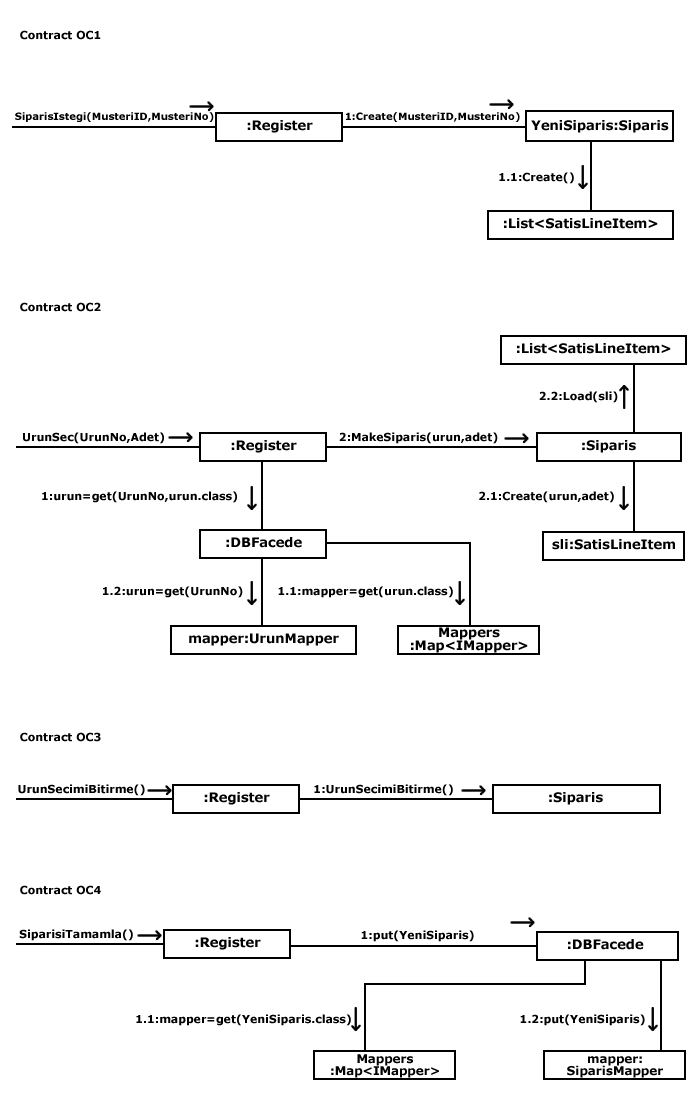
**Use Case 3 :**

**INTERANCTION DIAGRAMS**

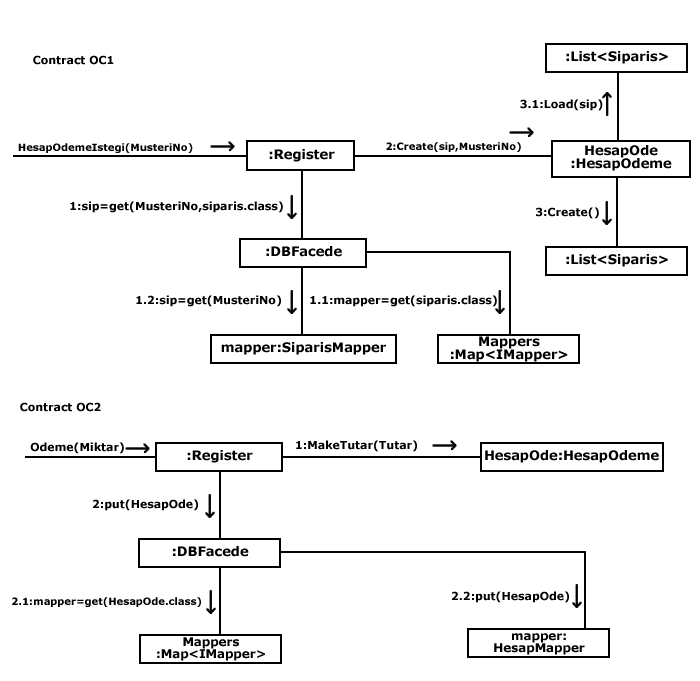
**Operation Contract : Ürün Kaydetme**

****

**Operation Contract : Sipariş Verme**

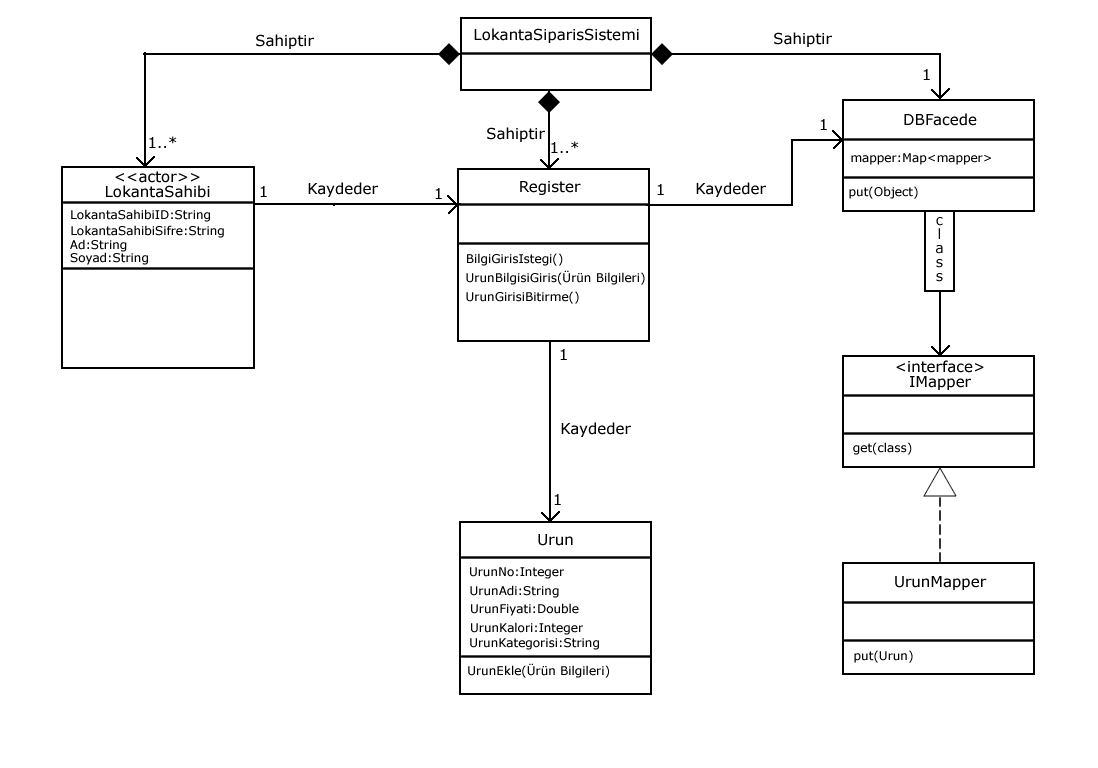
****

**Operation Contract : Hesap Ödeme**

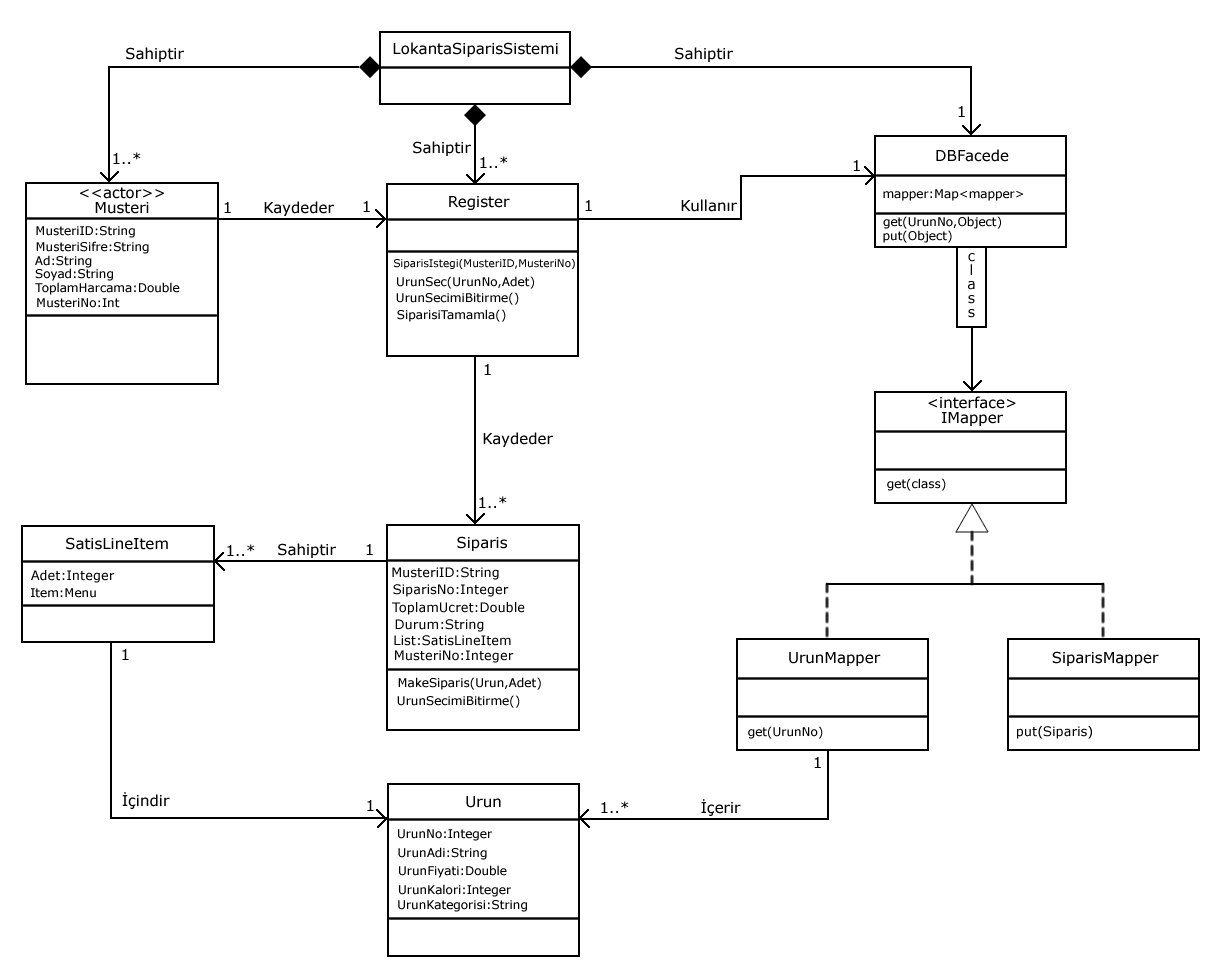
****

**UML CLASS DIAGRAMS**

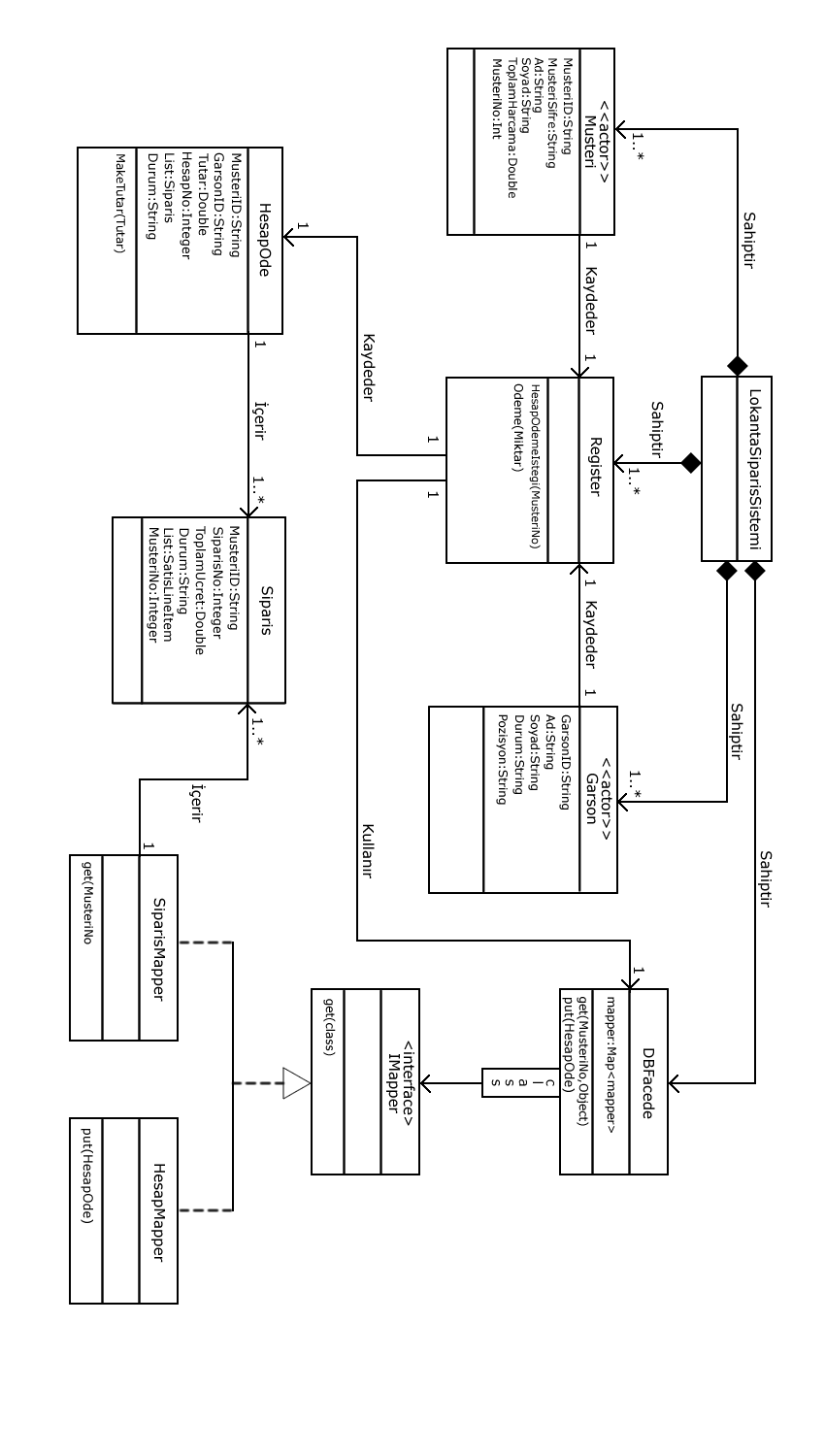
**UML Class Diagram : Ürün Kaydetme**

****

**UML Class Diagram : Sipariş Verme**

****

**UML Class Diagram : Hesap Ödeme**

****