ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
Nesneye Dayalı Analiz ve Tasarım Dersi

Proje Adı:
Lokanta Sipariş sistemi
(Restaurant Ordering System : ROSys)

Final Report

Grup No: 06
Grup Adı: Firestone

Grup Üyeleri:

Amer Sawan (130401073)

Abdoul Karim Touré (130401078)

Danışman: Yrd. Doç. Dr. Ali Murat Tiryaki

Çanakkale 2017

Table of Contents

1. ROSys Hakkında Vizyonumuz

1.1. Giriş (Introduction):

Lokanta yönetimi ve yemek siparişi sistemi, batı dünyasındaki çoğu restoranın benimsediği en yeni servislerden biridir. Bu yöntemle restoran yönetimi ve yiyecek siparişleri kolaylaşır. Bu nedenle, bu projede tasarlanan sistem, müşterilerin yiyeceklerini otomatik olarak sipariş etmelerini sağlayacak bir sipariş sistemidir. Lokanta sipariş sistemi mevcut olan lokanta veya Restoran'da müşterilerin siparişleri alabilecek, gönderilen siparişleri yemek hazırlayan kişilere gösterebilecek, çalışan garsonlar yardımıyla hesabı ödetebilecek, yönetimlere raporlar üretebilecek bir sistemdir.

İnternet bilinci ve onunla ilişkili teknolojilerdeki büyük artış nedeniyle, otomasyonda birçok fırsat ortaya çıkıyor. Pek çok restorasyon servis sağlayıcılar şimdi işlerine kolaylıkla giriyor. Teknolojinin getirdiği bu işlerden biri, gıda sipariş sisteminin otomasyonudur.

Bugünkü fast food ve online sipariş çağında , birçok restoran zengin bir yemek deneyimi sunmak yerine, hızlı hazırlama ve hızlı sipariş vermeye odaklanmayı tercih etti. Yakın zamana kadar, bu teslimat emirlerinin çoğu telefon üzerinden yapıldı ve garsonlara dayandı, ancak bu sistemin birçok dezavantajı var. Çalışan kadro sayısı az veya bazı saatlerde kalabalıktan dolayı oluşan yavaş tepki ve yanıtlama müşterileri rahatsız eden bir problemdir. Aynı zamanda garson veya çalışan kişi sözle iletişim kurulduğu için yanlış anlama veya bazı isteklerin unutma imkânı daha çoktur.

Bu projede önerdiğimiz restoran sipariş sistemi ROSys'tir. Başlangıçta restoranlarda kullanılmak üzere tasarlanmış bir otomasyon sistemidir, ancak herhangi bir gıda dağıtım endüstrisinde de geçerlidir.Bu sistemin en büyük avantajı hem müşteriye hem de restorana yönelik sipariş sürecini büyük ölçüde basitleştirmesidir.ROSys, siparişlerin alınması sürecinin tamamı otomatik olduğu için, restoran yönetimi ekibindeki yükü de oldukça hafifletiyor. Bir sipariş, dizayn edilecek bir konsol veya terminal vasıtasıyla sisteme yerleştirildiğinde, veritabanına yerleştirilir ve daha sonra restoran sorumluları üzerindeki bir masaüstü uygulaması tarafından neredeyse gerçek zamanlı olarak alınır. ROSys içinde siparişteki tüm öğeler, ilgili seçenekler ve teslimat bilgileri ile birlikte, özlü ve okunması kolay bir şekilde görüntülenir. Bu, restoran çalışanlarının siparişleri hızla geçmesini ve gerekli öğeleri en az gecikme ve karışıklıkla üretmesini sağlar.ROSys'in en büyük avantajı esnekliğidir.

1.2. Tanıtımlar (Definition Of Terms):

Yemek: İnsanın veya hayvanların yediği ya da içtiği ya da bitkiyi emerek yaşamı ve büyümeyi sürdürmek için besleyici herhangi bir maddedir.

Restoran veya lokanta: (Yeme yeri), yemeklerin ve içeceklerin satıldığı ve müşterilere servis edildiği bir yerdir.

Menu: Bir restoranda bulunan yemeklerin veya bir restoranda veya bir yemekte servis edilen veya sunulacak yiyeceklerin bir listesi.

Kredi Kart: ödeme sistemi olarak kullanıcılara verilen bir ödeme kartıymış. Kart sahibinin, sahibinin kendileri için ödeme yapma sözü üzerine mal ve hizmetler için ödeme yapmasına izin verir.

Müşteri: Bazen müşteri, alıcı veya satın alan kişi olarak bilinir), bir satıcının, satıcının veya tedarikçinin parasal veya değerli bir bedel karşılığında elde ettiği mal, hizmet, ürün veya fikir sahibidir.

Sipariş: Bir şeyin yapılmasını, gönderilmesini, getirilmesini isteme.

Sipariş sistemi: Bu, sipariş verme sürecinin işlenmesinde kullanılan ayrıntılı yöntemler kümesi olarak adlandırılır.

1.3. Sistemden beklenen/Gereksinimleri (Business Requirements)

1.3.1. Müşteriler

Lokantadaki masa üzerindeki konsol üzerinden:

- Girişi yapabilme.
- Menülerin tipleri ve fotoğraflar görebilme ve herhangi birisini sipariş edebilme.
- Yemek yerken siparişe ilave yemek ekleyebilme.
- Yemek bittikten sonra müşteri memnuniyeti form doldurabilme.
- Yemeğin faturası ödeyebilme.
- Fatura öderken müşteri tarafından ayrı bahşiş miktarı seçebilme.
- Sipariş hazırlarken hangi tabaklar hazır olup olmadığını takip edebilme.

Lokantadaki masa üzerindeki konsol üzerinden:

- Yeni hesap oluşturabilme.
- Giriş yapıp önceki siparişler görebilme.
- online olarak seçilen bir zaman içinde bir masayı rezervasyon edebilme.
- Yediği yemeklerin hakkında istatistikler görebilme (Kcal, Karbonhidrat, Protein ve yağ … vb. gibi)
- Lokantanın hakkında bilgi görebilme (açılış saati, menü çeşitleri...vb.).

1.3.2. Çalışan kadro

- Konsol üzerinden müşterilerin verdiği siparişler görebilme, bir masa üzerinde hala müşteri var mu yok mu
- Müşterinin hesabı ödetebilmek için masanın toplam hesap miktarı görebilme ve ödetebilme imkânı (kredi kartı ile veya nakit)
- Ödenen faturaların fişleri kullanıcıya yazdırabilme veya görüntüleme veya müşterinin hesabına bağlı olan e-mail’ine gönderebilme.

1.3.3. Yemek hazırlayan kişiler:

- Müşteriler tarafından verilen tabak siparişleri girebilme.
- Bir tabak bittiğinde sistemde bittiği belirlemek için konsol üzerinde imkânı sağlamak.

1.3.4. Raporlama Yöneticileri

- Günlük veya haftalık veya aylık veya seçilen 2 tarih arasında aşağıdaki raporlardan herhangi birisinden çıkartabilme:
- Satış hakkına raporlar (kullanıcılar, menüler, tabaklar, içecekler…vb.)
- Sipariş gecikme ve sistemin performansı hakkında raporlar.
- Müşterilerin memnuiyeti raporları.

1.3.5. Yöneticiler

- Sistem bulunan tüm lokanta hakkına bilgileri güncelleyebilme (adres, menüler, tabaklae, içeckler…vb.)
- Lokantanın binasında bulunan masalar hakkında bilgiler ekleyebilme (Numara, kişi sayısı, bulunduğu yeri…vb).
- Çalışanlar yönetebilme.

1.4. Sistem görünümü (Solution Overview)

Hemen hemen her alanda bilgisayar teknolojisinin hızlı gelişimi ve bilgi yönetimi ile ilişkili kullanımı göz önüne alındığında, firmaların müşterilerin talepleri doğrultusunda buluşmalarını sağlayan sipariş sisteminin geliştirilmesine bakmak önem kazanmıştır.Bu nedenle, gıda sipariş ve dağıtım sistemi, müşterilerin ve yönetimin şunları yapmasına yardımcı olacaktır: şirketlerinde mevcut yiyecekleri tanıtmak , mevcut sistemdeki iş yükünü azaltmak , veri işlemede boşa harcanan zamanı azaltmak,

çevrimiçi satın alma ve fast food dağıtımı için bir platform oluşturumak, satın alınan sipariş ve teslimat hakkında doğru kayıt tutmak.

1.5. Kapsam ve Sınırlamalar (Scope & Limitations)

Zaman ve mali kısıtlamalar nedeniyle, geliştirilecek yazılım yalnızca Gıda sipariş etme ve ödemeler kapsıyor.

1.6. ROSys işlevsellikleri (Functionalities)

Bu çalışma, geliştirilecek yeni bir sistemin çerçevesini ortaya koyuyor ve müşterilerin ROSys'e giriş yapabilecekleri bir konsoldan bir arayüz oluşturmasına izin veriyor.

İstedikleri mal veya gıdaları seçtikten sonra müşterilerin sistem üzerinden ücretlerini ödeyebilirler.

Bunun getireceği işlevler şunlardır:

1. ROSys, müşterilerin restoranın gerçekten ne yaptığı hakkında daha ve güvenilir bilgi toplayabilecekleri bir arayüz oluşturmaktadır.

2. Sunulan gıdalar ve hizmetler, müşterilere seçebilecekleri ve seçebilecekleri mevcut gıdaların farklı kategorilerini sağlayacaktır.

3. Bu, müşteri ve çalışan arasında dostça bir çevre sağlayacak ve böylece gıda sipariş sistemi verimliliğini artıracaktır.

4. Ayrıca müşteriler tarafından yapılan siparişlerin kolayca alınması için yardımcı olacaktır.

2. Ek Spesifikasyon (Supplementary specification)

2.1. Giriş (Introduction)

Use Case’ lerin kullanımı yeterli değildir . Başka gereksinim de vardır:

belgelendirme, ambalajlama, desteklenebilirlik, lisanslama, vb.

Bunlar Ek Spesifikasyonda ele geçirilmektedir. Ve bu ikinci raporumuzun birinci bölümünün amacı, ayrıntılı bir Ek Spesifikasyon sunmak değildir . Temel amacı, use case gereksinimlerini analiz etmek ve nesneye dayalı analizi kullanmak.

Bu nedenle, daha önceki ve gelecekteki çalışmalar arasında bağlantılar kurmak, kayda değer konuları vurgulamak ve çabucak ilerlemek için yalnızca ek spesifikasyonun bazı kısımlarına değinilmiştir. Bu ilk bölüm, use case’lerinde yakalanmayan tüm ROSys gereksinimlerinin deposudur.

2.2. ROSys Nitelikleri (Quality Attributes)

a. İşlevsellik (Functionality)

ROSys’e tüm girişler ve hataları kalıcı depolamaya geçecek.

Birkaç kullanım durumunun çeşitli senaryo noktalarında (tanımlanacak), sistemin işlevselliğini o noktada veya olayda yürütülen keyfi kurallar kümesiyle özelleştirme yeteneğini destekleyecek.

ROSys kullanımı kullanıcı kimlik doğrulaması gerektirecek.

b. Kullanılabilirlik (Usability)

Müşteri, tüm menülerin küçük bir monitör ekranını görebilecektir. Dolayısıyla:

- Metin 1 metreden kolayca görülebilir olmalıdır.
- Monitör ekranı renkleri renk körlüğünün ile ilişkili renkler olmamasıdır.

Müşterilerin hızlı bir şekilde yemek istedikleri için ROSys'in hızlı, kolay ve hatasız işleme tabi tutulması gerekir.

Yoksa sipariş deneyimini daha az olumlu olarak algılarlar.

Kasiyer genellikle müşteriye bakar, bilgisayar ekranını değil. Bu nedenle, sinyaller ve uyarılar yalnızca grafiklerden ziyade sesle iletilmelidir.

c. Performans

ROSys kullanılabilirliği altında belirtildiği gibi, müşteriler siparişleri çok hızlı bir şekilde tamamlamak istiyorlar. Potansiyel bir tıkanıklık, dış ödeme yetkisidir.

Amacımız, zamanın% 90'ından 1 dakikadan az bir sürede yetki vermektir.

d. Güvenilirlik (Reliability)

Harici hizmetler (ödeme yetkilisi, muhasebe sistemi, ...) kullanmada başarısızlık varsa ROSys, yine de bir siparişin tamamlanması için yerel bir çözümle çözmeye çalışacaktır.

e. Desteklenebilirlik (Supportability)

Bir sipariş işlenirken farklı müşterilerin kendine özgü sipariş kuralları ve işleme ihtiyaçları vardır. Bu nedenle, senaryodaki tanımlanmış birkaç noktada (örneğin, yeni bir sipariş başlatıldığında, yeni bir sipariş eklendiğinde) takılabilir iş kuralı etkinleştirilecektir.

Farklı müşteriler, kendi ihtiyaçlarına göre değişen sistem yapılandırmalarını ister. Buna ek olarak, değişen istek ve performans ihtiyaçlarını yansıtmak için bu yapılandırmaları değiştirme kabiliyetlerini arıyorlar. Bu nedenle, ROSys sistemi bu ihtiyaçları yansıtacak şekilde yapılandırılabilir olacaktır. Bu alanda daha fazla analiz yapılması gerekiyor; esneklik derecesini ve bunu gerçekleştirme çabasını kapsar

2.3. ROSys İmplementasyon Kısıtlamaları (Implementation Constraints)

ROSys geliştiricileri ekibi olarak, bir Java teknolojileri çözümü üzerinde ısrar ediyoruz, bunun öngörülmesi, geliştirme kolaylığına ek olarak uzun vadeli taşıma ve desteklenebilirliği de geliştirecektir.

2.4. Open Source Bileşenler (Open Source Components)

Genel olarak, bu proje üzerinde ücretsiz Java teknolojisi açık kaynak bileşenlerinin kullanımını maksimize etmenizi öneririz.

2.5. ROSys Arayüzleri (Interfaces)

a. Dikkat çeken Donanım Arayüzleri

- Dokunmatik ekran monitörü (işletim sistemi tarafından normal bir monitör olarak algılanır ve dokunmatik hareketler fare olayları olarak algılanır)
- Makbuz yazıcısı
- Kredi / banka kartı okuyucu

b. Yazılım Arayüzleri

- Birçok dış işbirliği sistemi (muhasebe, envanter, ...) için, değişen sistemleri ve dolayısıyla değişen arayüzleri takabilmeliyiz.

2.6. Ticari kurallar (Business Rules)

Id	Kural	Değiştirilebilirlik	Kaynak
1	Kredi ödemeleri için imza gerekmektedir.	Müşteri imzası gerekli olmaya devam edecek, ancak birkaç yıl içinde müşterilerin çoğu dijital yakalama cihazında imza yakalamayı istiyor ve birkaç yıl içinde yeni benzersiz dijital destek talebinde bulunmasını bekliyoruz Kod "imza" artık ABD yasası tarafından desteklenmektedir.	Tüm kredi yetkilendirme şirketlerinin politikası.
2	Vergi kuralları. Satışlar, eklenen vergiler gerektirir. Mevcut ayrıntılar için hükümet tüzüklerine bakın.	Yüksek. Vergi kanunları her yıl hükümet düzeyinde değişir.	Hukuk

Id	Kural	Değiştirilebilirlik	Kaynak
3	Kredi ödeme ters kayıtları sadece nakit olarak değil, alıcının kredi hesabına kredi olarak ödenebilir.	Düşük	Tüm kredi yetkilendirme şirketlerinin politikası.
4	Müşteri indirim kuralları. Örnek: • Müşteri -%10 indirim. • Kıdemli -%15 indirim.	Yüksek. Her perakendeci farklı kurallar kullanır.	Perakende politikası

2.7. Yasal Sorunlar

Açık kaynak yazılımları içeren ürünlerin satışına izin vermek için lisans kısıtlamaları çözülebilecek olursa, bazı açık kaynak bileşenlerini önermekteyiz.
Tüm vergi kuralları satış sırasında yasal olarak uygulanmalıdır. Bunların sık sık değişebileceğini unutmayalım.

2.8. Fiyatlandırma ve Ödeme İşletmesi (Pricing And Payment Handling)

İş kuralları bölümünde açıklanan fiyatlandırma kurallarına ek olarak, siparişin ve menülerin orijinal bir maliyeti ve isteğe bağlı olarak daimi bir maliyeti olduğunu unutmayalım.
Bir siparişin maliyeti (daha fazla indirime girmeden önce) kalıcı maliyettir. Kuruluşlar, muhasebe ve vergi nedenleriyle kalıcı bir maliyet olsa dahi orijinal maliyetlerini korurlar.

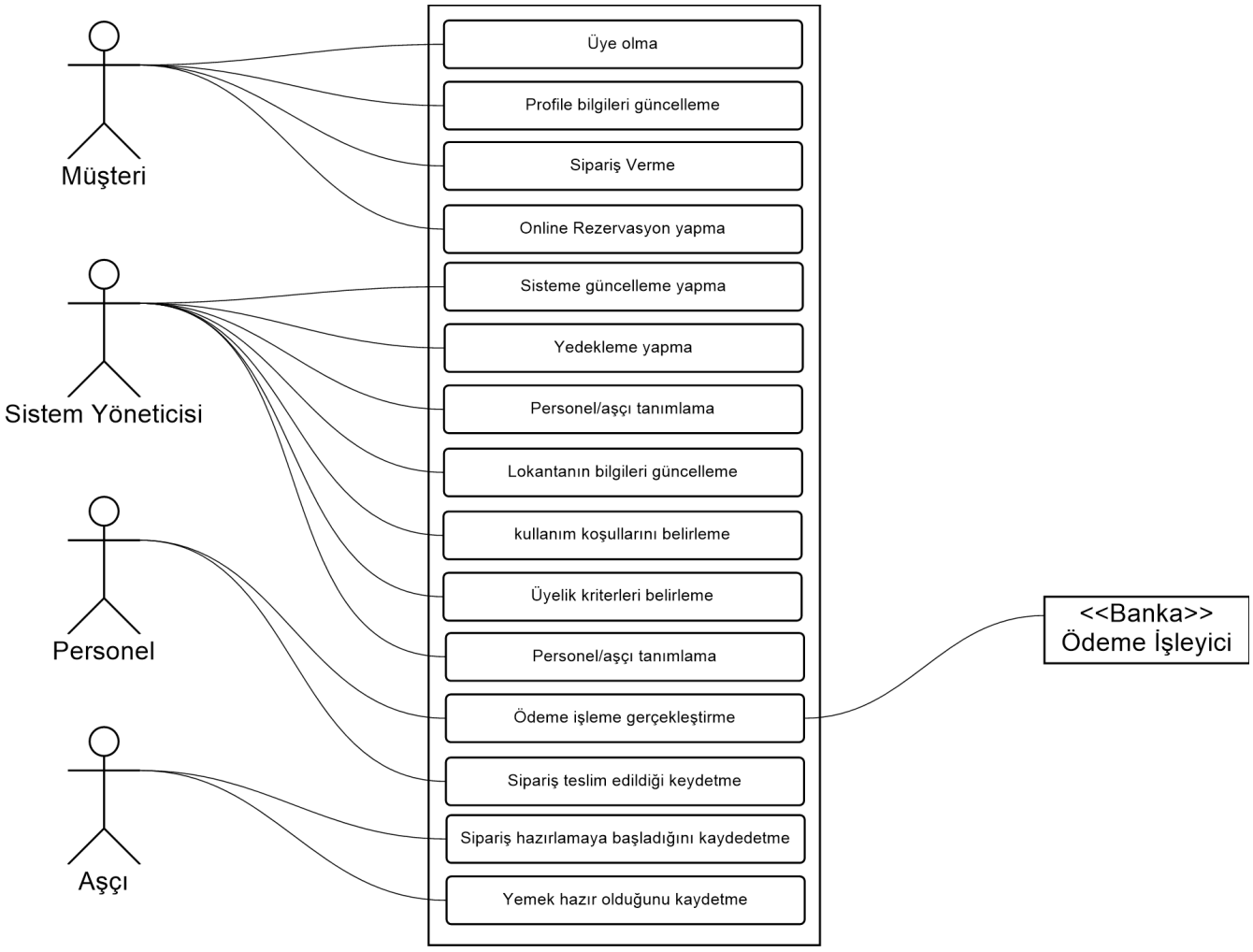
Elektronik kredi veya banka havalesi bir ödeme yetkilendirme servisince onaylandığında, alıcıyı değil satıcıyı ödemekten sorumludurlar.
Sonuç olarak, her bir ödeme için, satıcı, alacak hesaplarından dolayı yetkilendirme servisinden para yatırmaya ihtiyaç duyar.
Genellikle, yetkilendirme hizmeti, günlük toplam borç ödemeleri için satıcının hesabına elektronik para transferi yapar, hizmet ücreti başına işlem ücreti düşer.

2.9. Satiş Vergisi (Sales Tax)

Satış vergisi hesaplamaları çok karmaşık olabilir ve hükümetin her kademesinde mevzuata yanıt olarak düzenli olarak değişir.
Bu nedenle, vergi hesaplamalarını üçüncü parti hesap makinesi yazılımına devretmek (bunun için birkaç tane mevcut) tavsiye edilmektedir.
Vergi, şehir, bölge, eyalet ve ulusal organlara bağlı olabilir.
Bazı öğeler, alıcı veya hedef alıcıya (örneğin, bir çiftçi veya bir çocuk) bağlı olarak niteliksiz vergi muafiyeti veya muaf olabilir.

3. Aktor-Gol Model (Actor – Goal Model):

Müşteri	• Sisteme üye olur
	• Profile bilgileri günceller
	• **Sipariş verir**
	• Sisteme online rezervasyon yapar
Sistem Yöneticisi	• Sistemde güncellemeleri yapar
	• Sistemde yedekleme yapar
	• Sistemde personel veya aşçı tanımlar
	• Lokantanın bilgileri günceller
	• Sistem kullanım koşullarını belirler.
	• Üyelik konusunda kriterleri belirler.
Personel	• **Ödeme işlemi gerçekleştirir**
	• Sistemde sipariş teslim edildiği kaydeder
Aşçı	• Yemek hazırlamaya başladığını kaydeder
	• **Yemek bittiğini sistemde kaydeder**



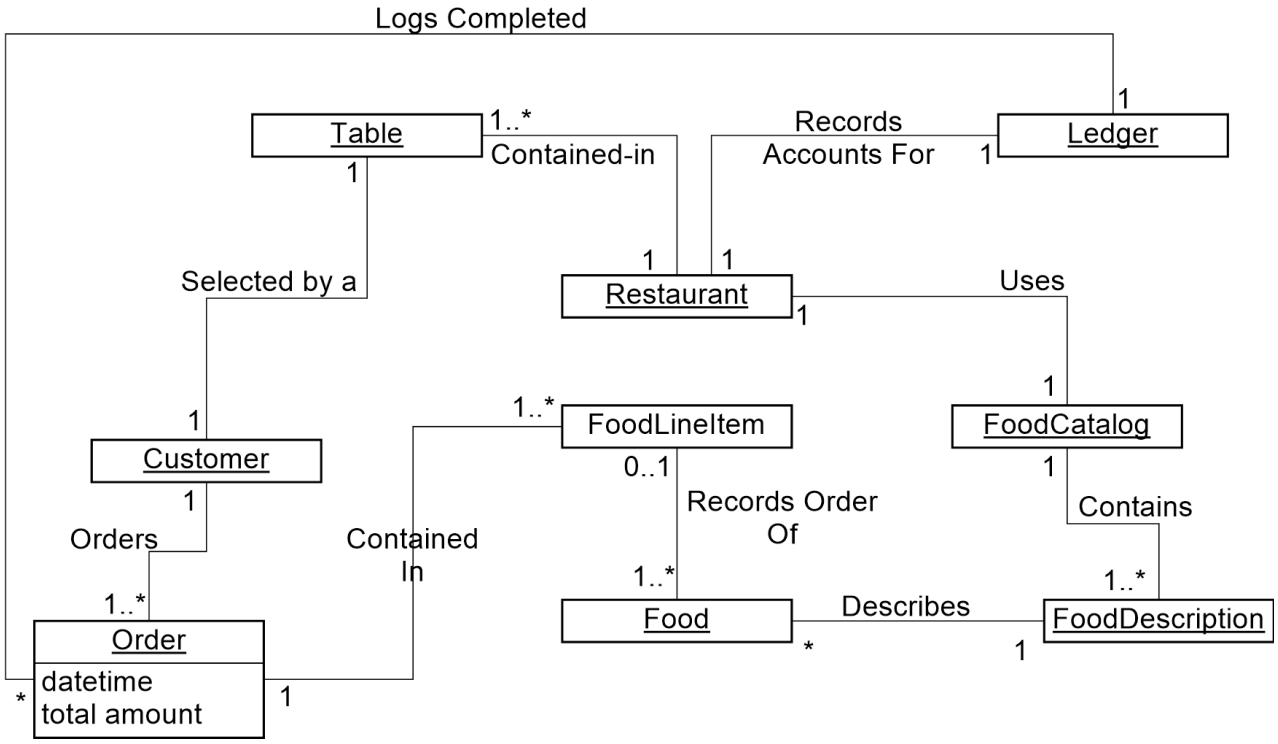
4. Use Cases

4.1 UC1 :Placing Order

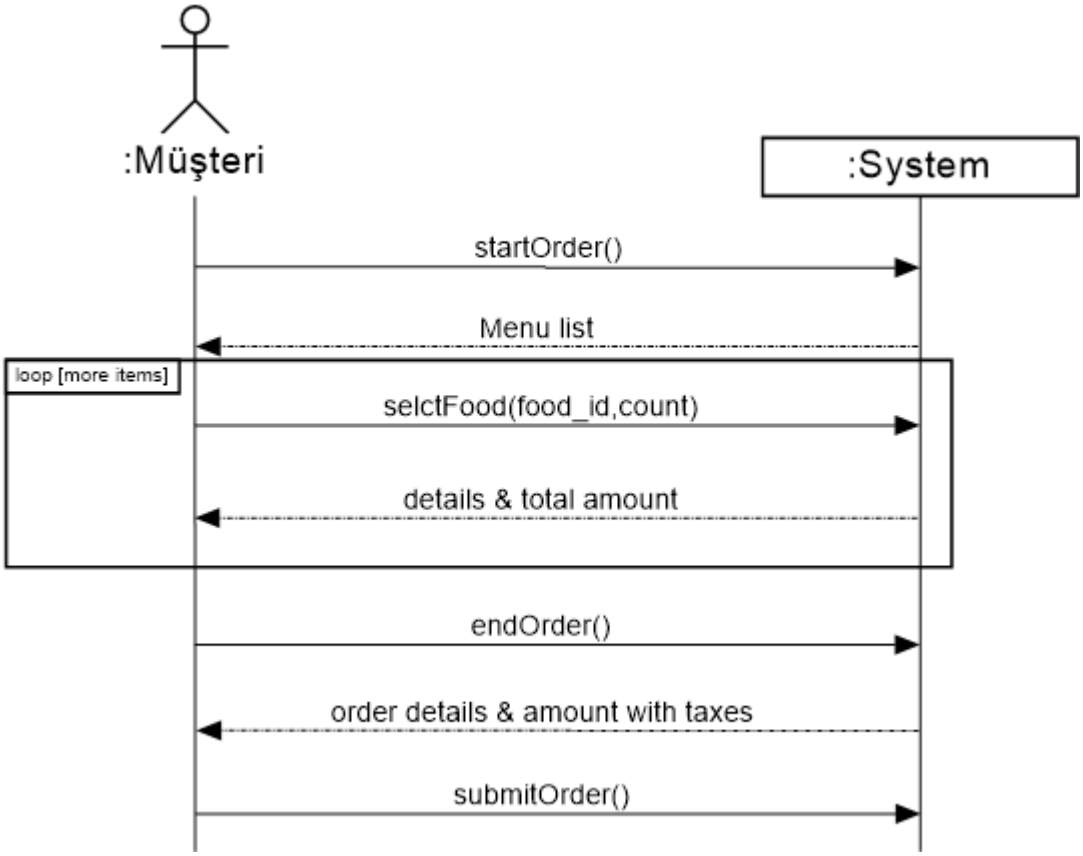
4.1.1 Use Case Senaryo

Scope	ROSys sistemi
Level	User Scope
Primary Actor	Müşteri
Stakeholders and Interests	• Müşteri: hızlı ve güvenli şekilde sipariş verebilmek. • Yemek Hazırlayan kişi: müşterilerin verdiği siparişlerin görüntülemek. • Sistem Yöneticisi: müşterilerin verdiği siparişler takip edebilmek ve bunların hakkında raporlar çıkartabilmek.
Preconditions	• Müşterinin kim olduğunu belli (Authenticated) olması lazım.
Triggers	• Müşteri lokantaya geldiğinde
Success Guarantee (Postconditions)	• Sipariş kaydedildi. • Vergiler düzgün bir şekilde hesaplandı.
Main Success Scenario	1- Müşteri lokantanın bir masa üzerinde oturur 2- Müşteri yeni sipariş başlar 3- Sistem, yemek çeşitlerin fiyatları ve fotoğrafları gösterir. 4- Müşteri bir yemek çeşitti ve miktarı seçer 5- Sistem, Seçilen yemeğin ismi ve hesaplanmış fiyatını ve şimdiye kadar toplam tutarı gösterir **Müşteri 3,4 adımları bittiğini söyleyene kadar tekrarlar** 6- Sistem seçilen yemeklerin listesi ve fiyatları ve Toplam tutarı hesaplanmış vergileri ile gösterir. 7- Müşteri, sipariş onaylar. 8- Sistem, sipariş onayladığını kaydeder. 9- Müşteri, verdiği sipariş gelmeye bekler (varsa)
Extensions (Alternative Scenarios)	4-6a. Müşteri siparişi iptal etmesi ister: 1. Müşteri siparişi iptal eder. 2. Sistem, kaydedilen siparişi öğeleri yok eder. 4-6b. Müşteri sipariş listesinden bir yemek çeşitti çıkartmak ister: 1. Müşteri seçilen yemekler listesinden birisi silmeye ister 2. Sistem silmek istenilen eleman siler, toplam tutarı yeniden hesaplar, güncel hesaplanan fiyatları gösterir 3. Müşteri kaldığı adımdan devam eder. 4-6c. Müşteri bir yemeğin miktarı değiştirmek ister: 1. Müşteri miktarını değiştirmek istediği yemeği seçer. 2. Müşteri yeni miktarı girer ve onaylar. 4. Sistem, seçilen yemeğin miktarı günceller, toplam tutarı yeniden hesaplar, güncel hesaplanan fiyatları gösterir 3. Müşteri kaldığı adımdan itibaren devam eder. *a. Sistem herhangi bir anda hataya düşer: 1. Sistem, yeniden başlatılır. 2. Sistem, müşteriden yeni sipariş alabilme haline döner
Frequency of Occurrence	Sürekli bir şekilde olabilecek şey
Miscellaneous (Open issues)	• Sistem gün boyunca aynı menüler mı gösterecek?

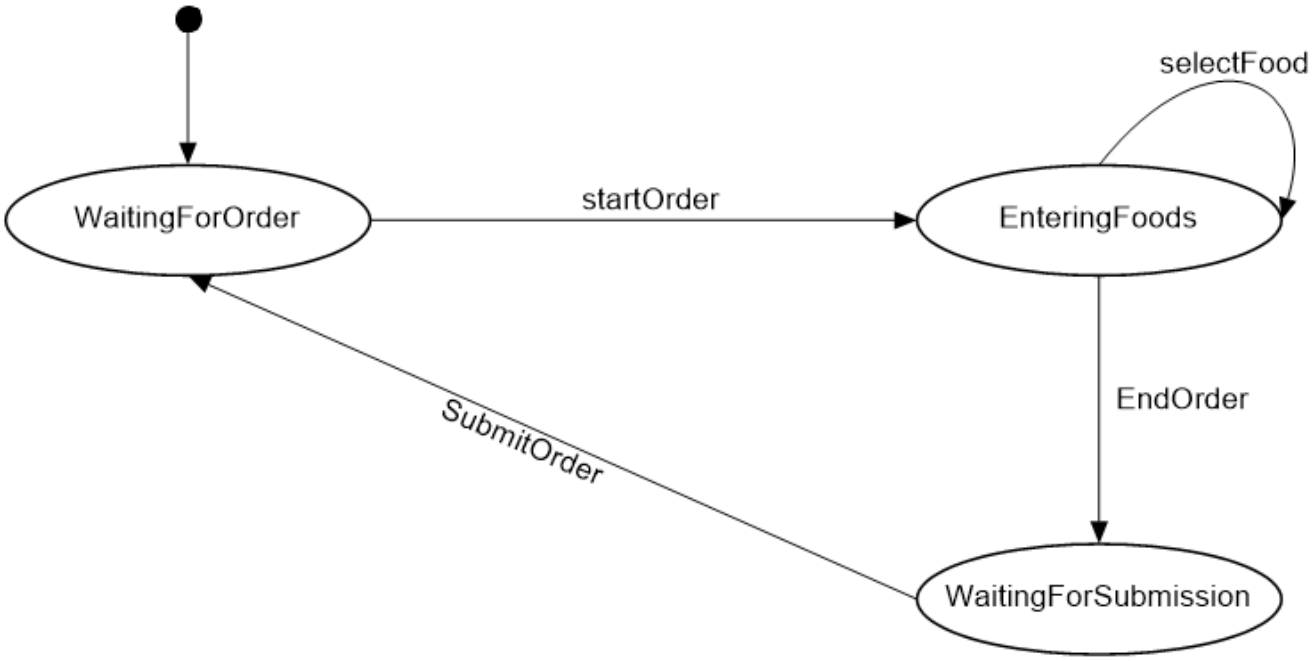
4.1.2 Domain Model



4.1.3 System Sequence Diagram



4.1.4 State Machine Diagram



4.1.5 Operation Contracts

Contract CO1: startOrder

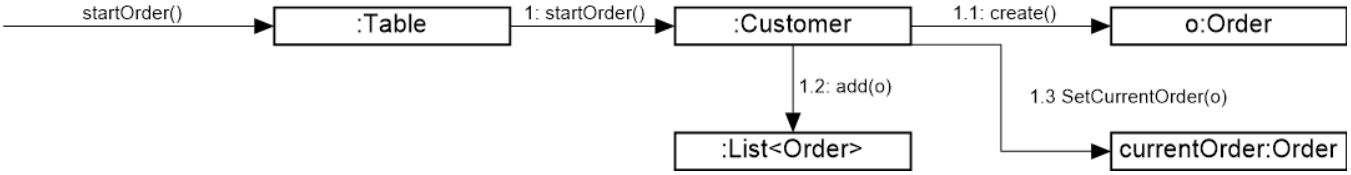
Operation	startOrder()
Cross References	Use Case : Placing order
Preconditions	none
Postconditions	- An Order instance was created with name (o). - (o) was associated with the currently logged in customer instance.

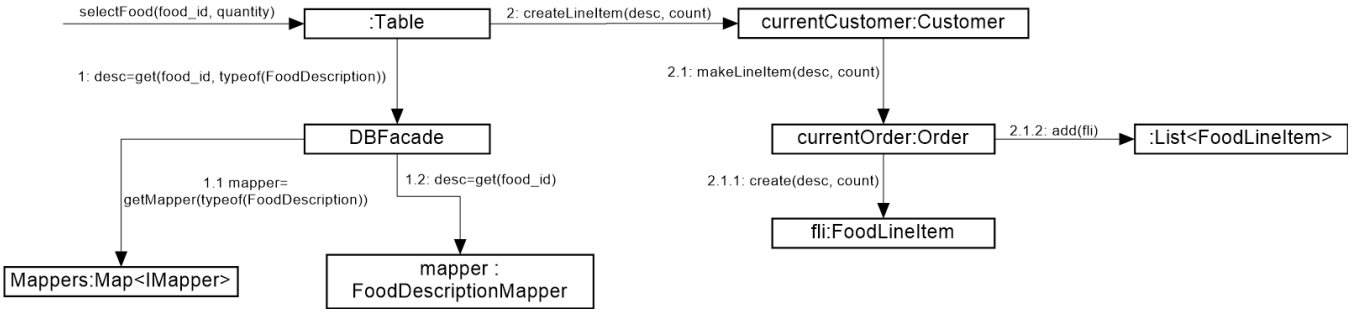
Contract CO2: selectFood	
Operation	selctFood(food_id, count)
Cross References	Use Case : Placing order
Preconditions	There is an order underway.
Postconditions	- A FoodLineItem instance was created with name (fli). - (fli) was associated with the current order. - (fli) was associated with FoodDescription, based on food_id match. - (fli.count) became count (which came from parameters)

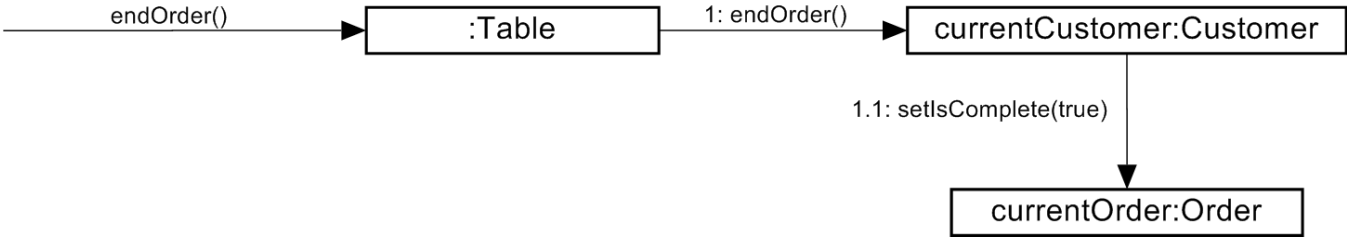
Contract CO3: endOrder	
Operation	endOrder()
Cross References	Use Case : Placing order
Preconditions	There is an order underway.
Postconditions	- The current underway order item’s isCompleted attribute became true.

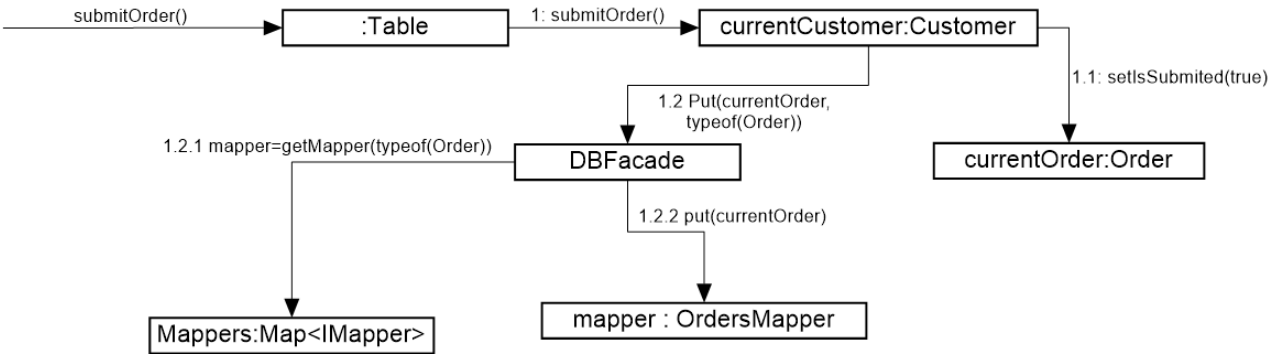
Contract CO4: submitOrder	
Operation	submitOrder ()
Cross References	Use Case : Placing order
Preconditions	There is an order underway.
Postconditions	- The current underway order item’s isSubmitted attribute became true.

4.1.6 Interaction Diagrams









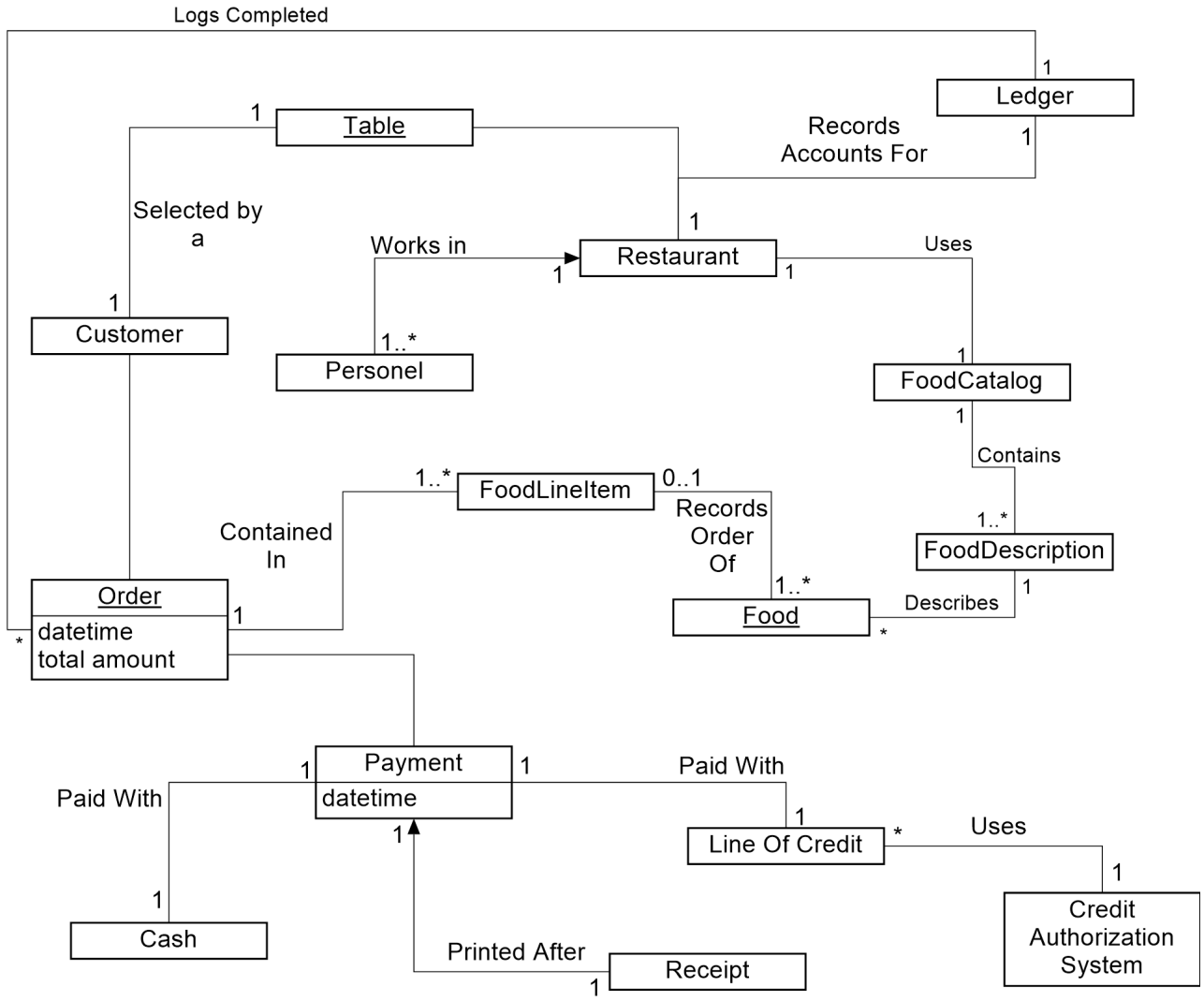
4.2 UC2 : Process Payment

4.2.1 Use Case Senaryo

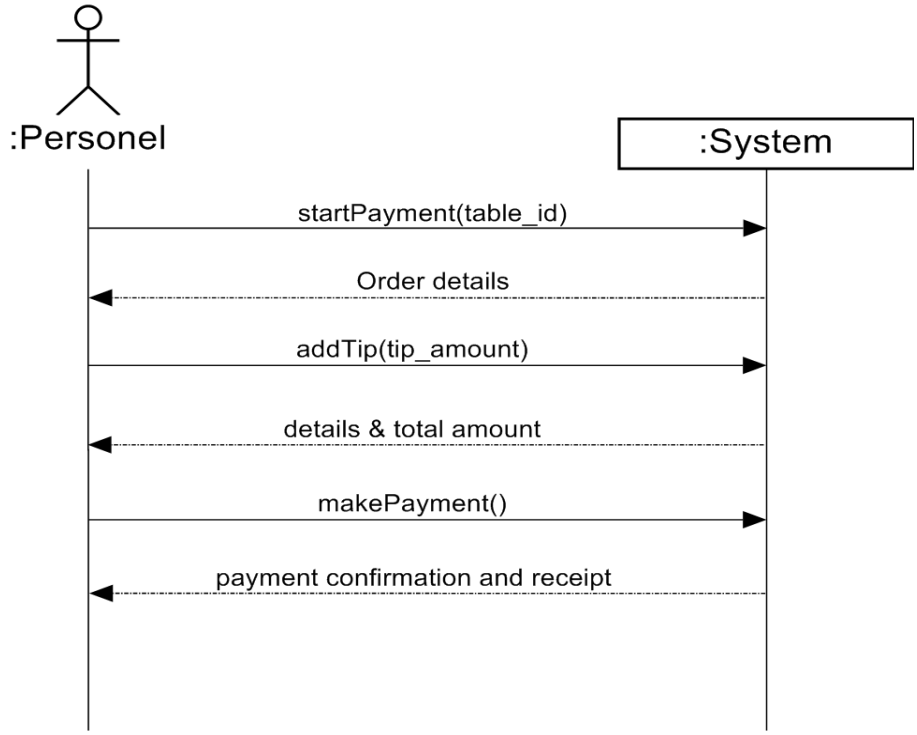
Scope	ROSys sistemi
Level	User Goal
Primary Actor	Personel
Stakeholders and Interests	• Müşteri: verdiği siparişlerin toplam tutarı en kolay, hızlı ve güvenli şekilde ödeyebilmektir. • Personel: Müşterinin ödeyeceğin miktarı fazla sorular sormadan ve hızlı bir şekilde tahsil edebilmek.
Preconditions	• Personel kim olduğunu belli (Authenticated) olması lazım.
Triggers	• Müşteri hesap ödeme istek. • Personel Müşteri lokantadan ayrılmak istediği görmek.
Success Guarantee (Postconditions)	• Ödeme işleyici ödemeyi kabul ettiğini kaydedildi. • Siparişin durumu ödendi durumda olması. • Vergiler düzgün bir şekilde hesaplandı. • Faturayı yazdırıldı.
Main Success Scenario	1. Personel, hangi masa için ödeme gerçekleştireceğini seçer 2. Sistem siparişlerin detaylar göstererek toplam tutarı vergileri ve detaylarıyla gösterir. 3. Personel müşteriye toplam tutarı söyler 4. Personel, müşteri kaç bahşiş vereceğini uygun bir şekilde sorar. 5. Personel bahşiş miktarı girer. 6. Sistem, toplam tutarı bahşiş miktarı ile gösterir 7. Personel toplam tutarı söyler ve ödeme ister. 8. Müşteri toplam tutarığı öder ve sistem onu ele alır. 9. Sistem, yapılan ödeme işlemi kaydeder. 10. Sistem, faturayı yazdırır. 11. Personel faturayı Müşteriye verir ve Müşteri lokantadan ayrılabilir.
Extensions (Alternative Scenarios)	1a. Seçilen masa için sipariş yoktur: 1. Sistem bu masa için sipariş olmadığını gösterir 1-7a. Müşteri fatura ödemeyi ertelemeyi karar alır: 1. Personel işlemi iptal eder. 8a. Müşteri ödemeyi kredi kart ile geçekleştirmek ister: 1. Müşteri kredi kart bilgilerini girer. 2. Sistem ödenecek tutarı ve kredi kartın bilgileri gösterir 3. Müşteri bilgileri doğru olduğunu onaylar. 3a. Müşteri bilgilerini düzenletmek ister. 1. Müşteri bilgilerin güncellemesi seçer 2. Sistem yeni kredi kartın bilgileri ister 3. Müşteri yeni bilgileri girer 4. Sistem, ödeme doğrulamak için ödeme işleyici 3rd-party bir servise gönderir ve ödeme onayı ister. 5. Sistem ödeme onayı alır 5a.Sistem, ödeme işleme başarısız sonuç alır: 1. Sistem, Personele işlem başarısız olduğunu gösterir. 2. Personel müşteriyi yeniden hangi ödeme şekli ile ödemeyi gerçekleştirmek ister 5b.Sistem çok üzün süre cevap almadan başarısız oldu (Timeout):

	1. Sistem, işlem (Timeout) olduğunu gösterir. 2. Personel, işlemi yeniden başlar, veya müşteriyi farklı ödeme şekli ile ödetebilir. 6. Sistem, Personele ödeme onayı gösterir. 7. Sistem, kredi kartın ödeme imzası alınması gerekiyor gösterir. 8. Personel, Müşterinin kredi kart ödeme imzasını alır. 9. Personel, kredi kart ödemenin imzasını alındı sisteme girer. 10. Sistem, Ana başarı senaryosunda 9. Adımdan itibaren devam eder. 8b. Müşteri ödemeyi nakit ile gerçekleştirmek ister: 1. Müşteri toplam tutarı kadar veya fazla öder 2. Personel, kalan para miktarı varsa müşteriye geri verir 3. Sistem, Ana başarı senaryosunda 9. Adımdan itibaren devam eder
Frequency of Occurrence	Sürekli bir şekilde olabilecek şey

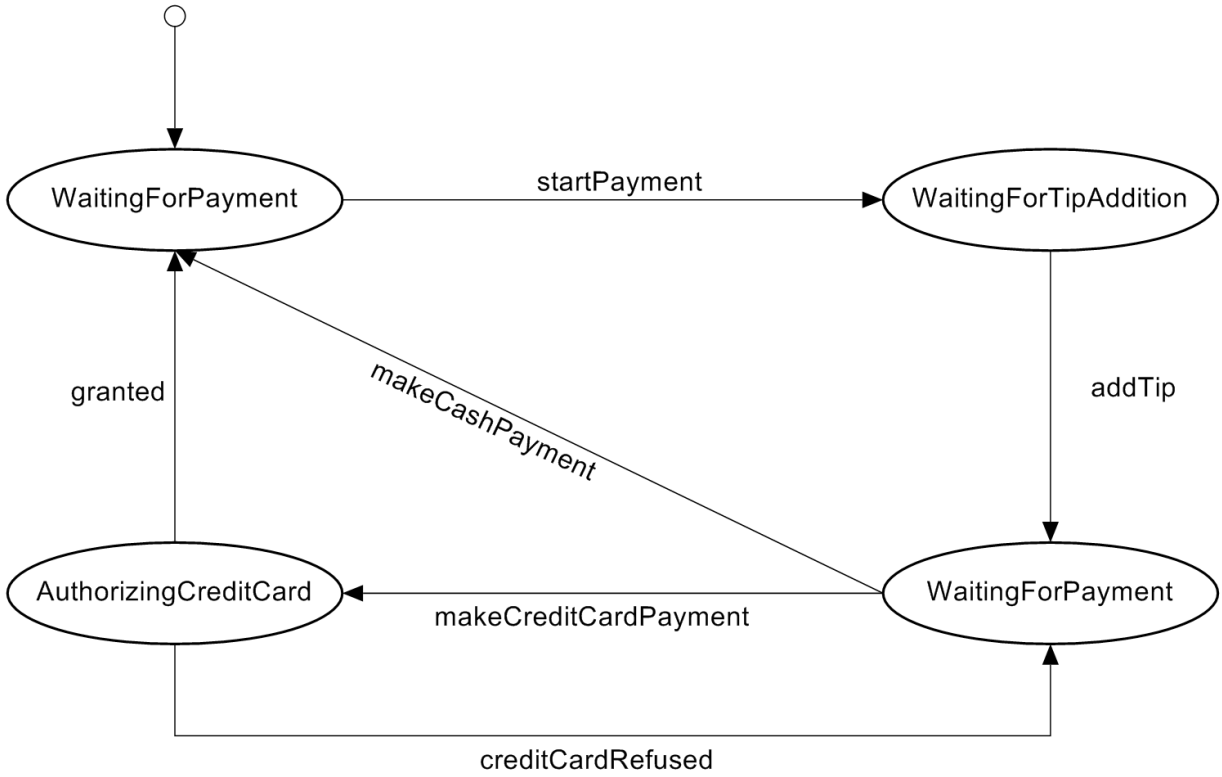
4.2.2 Domain Model



4.2.3 System Sequence Diagram



4.2.4 State Machine Diagram



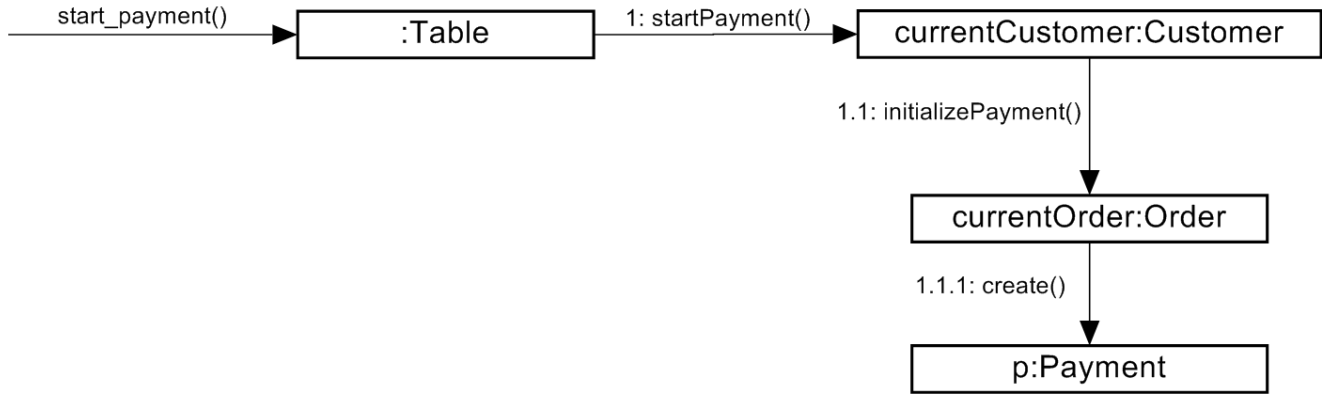
4.2.5 Operation Contracts

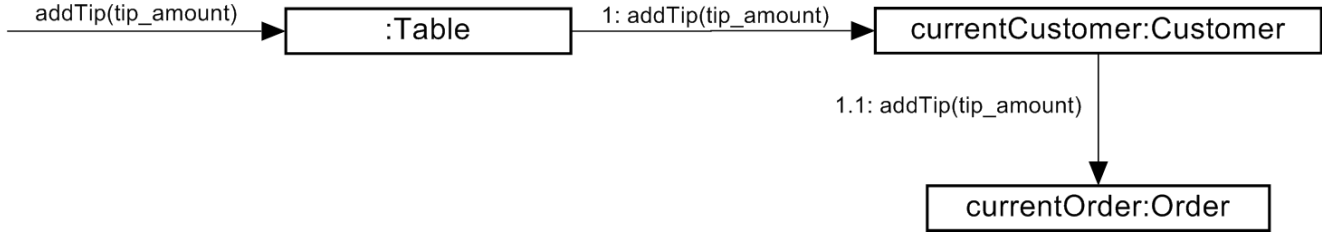
Contract CO1: startPayment	
Operation	startPayment()
Cross References	Use Case : Process Payment
Preconditions	none
Postconditions	- A Payment instance was created with name (p).

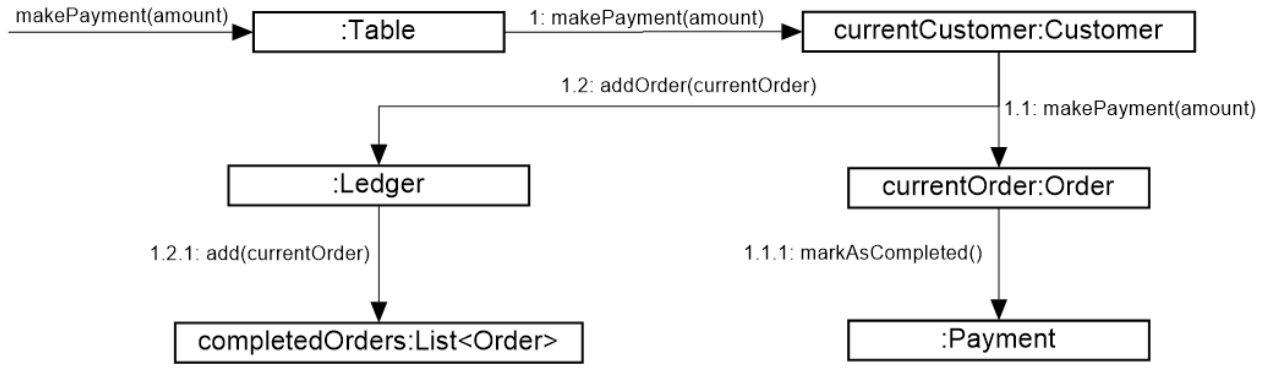
Contract CO2: selectOrder	
Operation	selectOrder(order_id, tip_amount)
Cross References	Use Case : Process Payment
Preconditions	There is payment underway with name (p)
Postconditions	- The current payment was associated with order instance, based on order_id - p.tip_amount attribute became equals to tip_amount (parameter)

Contract CO3: makePayment	
Operation	makePayment()
Cross References	Use Case : Process Payment
Preconditions	There is payment underway with name (p)
Postconditions	- p.isCompleted became true - the current order attached to the current customer became associated with the Ledger instance.

4.2.6 Interaction Diagrams





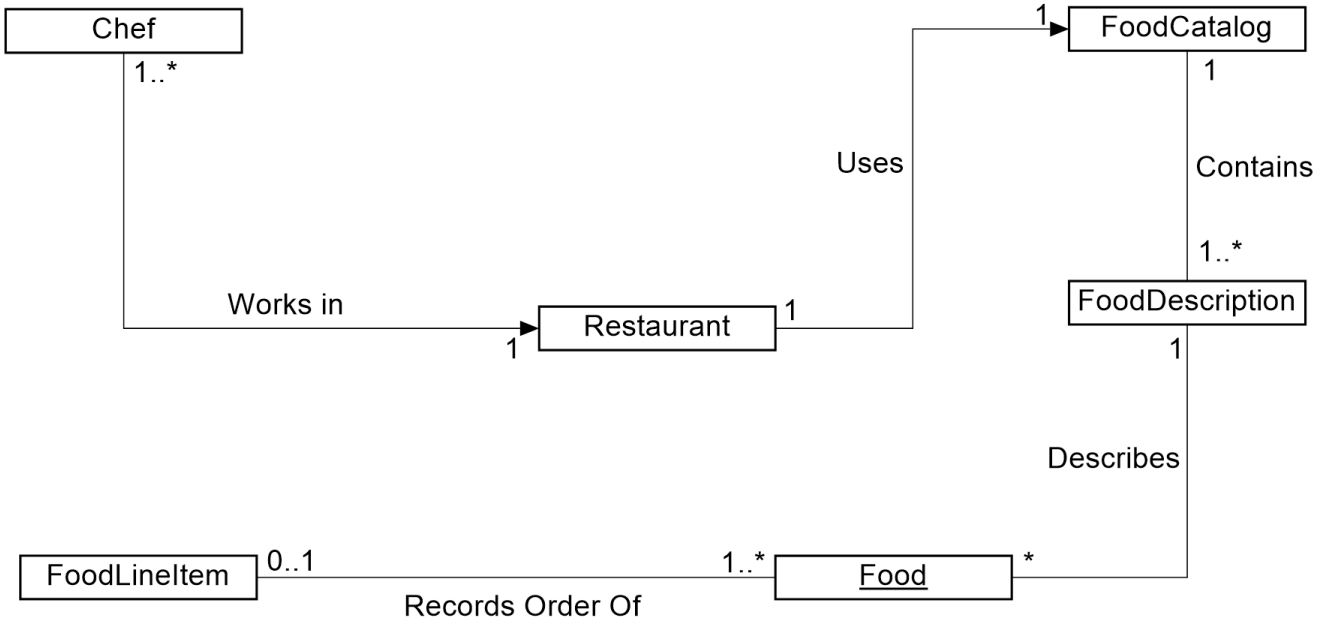


4.3 UC3 : Update Order Status to Ready

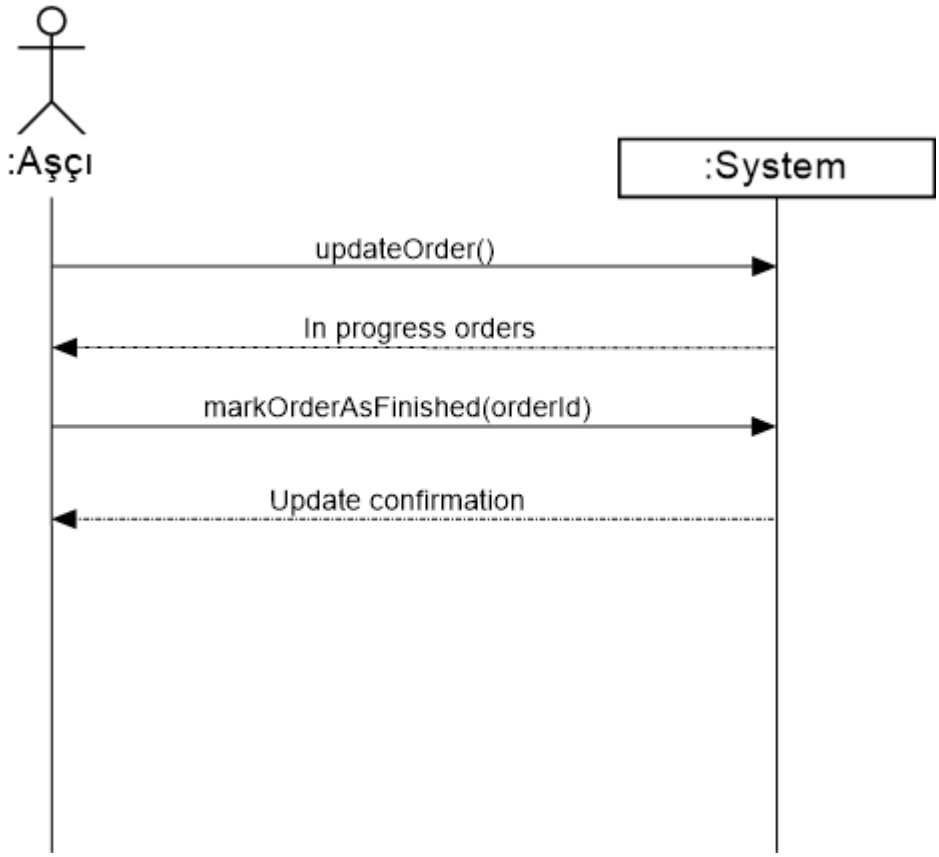
4.3.1 Use Case Senaryo

Scope	ROSys sistemi
Level	User Scope
Primary Actor	Aşçı
Stakeholders and Interests	• Müşteri: Müşterinin verdiği sipariş gerçek zamanda takip edebilmek. • Yemek Hazırlayan kişi: Sistemde kaydedilen yapılan ve hala yapılmayan siparişler ayrılabilmek. • Sistem Yöneticisi: Aşçının performansı hakkında raporlar çıkartabilmek.
Preconditions	• Aşçı sisteme giriş yapmış olması lazım
Triggers	• Aşçı çalıştığı yemek üzerinde hazır olduğunda
(Postconditions)	• Seçilen siparişin durumu “Hazır” durumunda olması
Main Success Scenario	1. Sistem, giriş yapan aşçının “In-Progress” olan siparişlerini gösterir 2. Aşçı, gösterilen siparişlerden bir tanesi seçer 3. Aşçı, sipariş bittiğini girer 4. Sistem, sipariş bittiğini kaydeder. 5. Sistem sipariş hazır olduğunu bütün personellere bir bildirim gönderir. 6. Personel, siparişi alıp müşteriye teslim etmek için işlemleri başlar.
Extensions (Alternative Scenarios)	6a. Aşçı sipariş eksik olduğunu farkeder: 1. Aşçı, son bitirmiş siparişlerden eksik olan siparişi seçer. 2. Aşçı, siparişin durumu “In-Progress” olarak seçer. 3. Sistem, seçilen siparişin durumu günceller. 3a. Personel sipariş aldığını ve teslimatı başladıysa: 1. Sistem sipariş alan sipariş eksik olduğunu personele acil bir bildirim gönderir. 2. Personel siparişi lokantanın mutfak kısmına geri bırakır. 2a. Sipariş teslim edildi: 1. Sistem, Aşçıya sipariş teslim olduğunu söyler. 2. Sistem, Personeller bu müşteriyi siparişini eksik olduğunu haber verebilmek için acil bir bildirim gönderir. 3. Aşçı siparişin eksik tabakları tamamlar ve ayrı bir sipariş gibi hazırlar. 4. Aşçı, eksikler tamamlandığında siparişin bittiğini seçer. 5. Sistem, ana başarı senaryosunda 4. adımdan devam eder
Frequency of Occurrence	Sürekli bir şekilde olabilecek şey
Miscellaneous (Open issues)	• Sipariş yanlış teslim edilirse ve personel/aşçı tarafında fark edemediler ise, nasıl bir senaryo olacak?

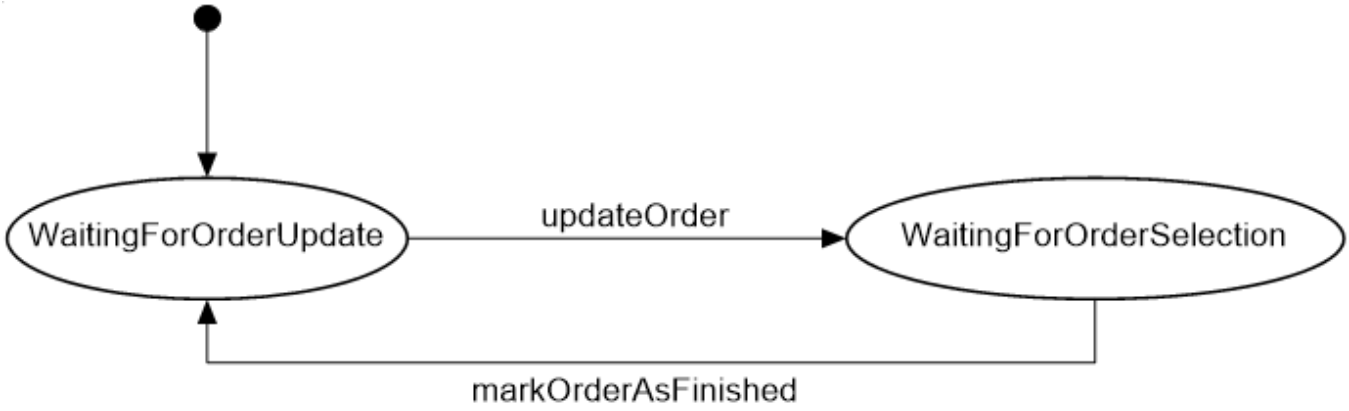
4.3.2 Domain Model



4.3.3 System Sequence Diagram



4.3.4 State Machine Diagram



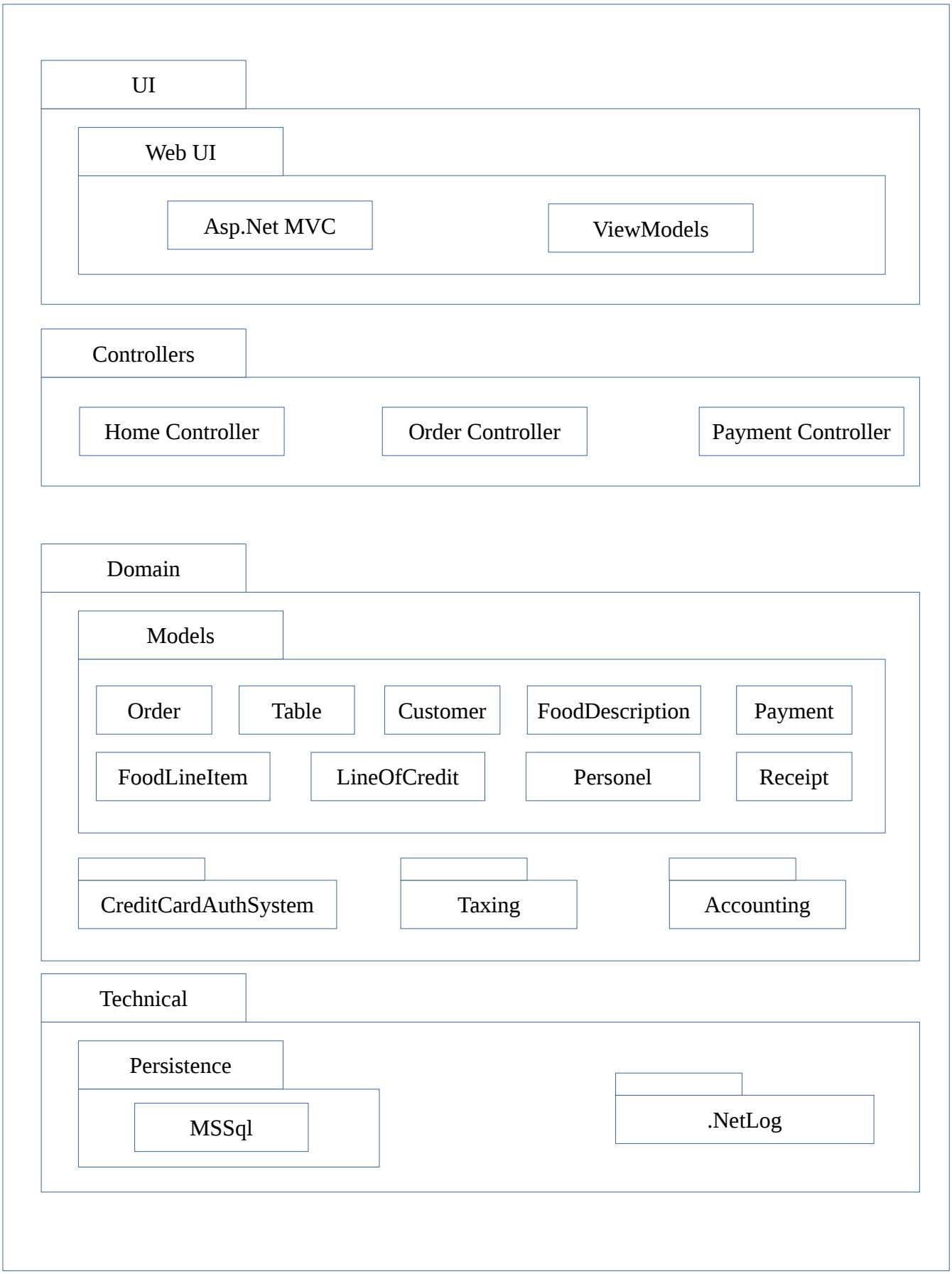
4.3.5 Operation Contracts

Contract CO1: updateOrder	
Operation	makePayment()
Cross References	Use Case : Process Payment
Preconditions	None
Postconditions	None

Contract CO2: markOrderAsFinished	
Operation	markOrderAsFinished(orderId)
Cross References	Use Case : Process Payment
Preconditions	None
Postconditions	- The order instance (o) that has o.id equals to oderId is updated so o.Status became “Finished”

4.3.6 Interaction Diagrams

5. System Architecture



6. UML Class Diagram