

**COMPUTER ENGINEERING**  
**OBJECT ORIENTED PROGRAMMING**  
**TOUR AND TRIP BOOKİNG-PURCHASİNG**  
**SYSTEM**

**PREPARED BY**

**1030520725– ÇAĞRI KUZULU**

**1030520768 – CAN KARABULUT**



## **İÇİNDEKİLER:**

1. Requirements Phase.....	5
1.1.Uygulama Dönemi.....	5
1.2.Sorular.....	5
1.3.İş Modeli.....	7
1.4.Sözlük.....	8
1.5.Use-Case Diyagramları.....	10
2. Analyse Phase.....	22
2.1.Fonksiyon Modelleme.....	23
2.2.Varlık Sınıfları.....	23
2.3.Temel CRC Kartları.....	27
2.4.Dinamik Modelleme.....	32
2.5.Sequence Diyagramları.....	33
2.6.Yazılım Proje Yönetim Planı.....	40
3. Design Phase.....	46
3.1.Sınıfların iç detaylarını belirlenmesi.....	46
3.2.Pseudo Kodlar.....	56
3.3.Form Arayüzleri.....	62



# Requirements Phase

## **1.1 UYGULAMA DÖNEMİ:**

Günümüzde uzun çalışma saatleri ve yoğun iş hayatının olumsuz etkilerini atmak ve dinlenmek için tatile ihtiyaç duymaktadır. Tatilciler gidecekleri yerleri belirlemede internet, televizyon, broşürler ve reklamları kullanarak karar vermektedir.

Google Türkiye'nin "Yeni Nesil Seyahat Alışkanlıkları" araştırması tatil süreçlerinde internetin artan önemi ortaya koymaktadır. Buna göre otel rezervasyonu yapacakların yüzde 95'i, uçak bileti alacakların ise yüzde 96'sı online araştırma yapmaktadır. Araştırmaya katılanların yüzde 68'i uçak biletini online olarak satın alırken, online otel rezervasyonu yapanların oranı yüzde 45 olarak karşımıza çıkmaktadır.

Turizm acenteleri ve tur şirketleri insanların daha kolay ve hızlı seyahat etmesi için hizmet veren ticari kuruluşlardır . Günümüzde internet ve online alışverişin yaygınlaşması turizm sektörünün de bu yönde gelişmesine , yazılım sektörüyle beraber çalışmasına neden olmuştur.

Bu bağlamda online sistemi bulunmayan bir turizm acentasıyla yaptığımız görüşme sonucu firmanın istekleri doğrultusunda web tabanlı çalışan bir yazılım yapmaya karar verdik .

## **1.2 SORULAR:**

1. Firmanızın interaktif çözümlerle kar marjının artacağını düşünüyor musunuz ?
2. Özel olarak tercih ettiğiniz bir arayüz var mı ?
3. Siteniz mobil uyumlu olmalı mı ?
4. Web tabanlı bir yazılım istemenizin sebebi nedir ?
5. Web sitesini yönetecek teknik elemana sahip misiniz ?
6. Kaç çalışanınız var ? Firma içinde ki görevleri nelerdir ?
7. Web sitesi admininizin ne gibi yetkileri olmasını istersiniz ?
8. Web sitesi adminin kullanıcı silme yetkisi olsun mu ?

9. Web sitesi admini kullanıcıların muhasebe bilgilerini görüntüle yetkisine sahip olsun mu ?
10. Organizatörün düzenleyeceği turları düzenleme yetkisine sahip olsun mu ?
11. Organizatör hangi sorumluluklara sahip ?
12. Oluşturulan tur paketlerini organizatör hangi ölçütlere göre düzenliyor ?
13. Organizatör otel ve ulaşım bilgilerini admine yolluyor mu ?
14. Sitenizi ziyaret eden potansiyel müşterileriniz sitenize üye olabilsin mi ?
15. Kayıt esnasında kullanıcılarınız kişisel bilgilerini sisteme girsin mi ?
16. Bu kişisel bilgileri ne için ve nerede kullanacaksınız ?
17. Kişisel bilgilerin kullanımı ve yapılan satışın müşteri açısından güvencesi için satış sözleşmesi oluşturuyor musunuz ?
18. Ön satış veya rezervasyon opsiyonu konulmasını istiyor musunuz ?
19. Müşterinin ödeme yaptıktan sonra iptal etme isteği için neler yapıyorsunuz ?
20. Müşterinin tura katılamaması durumunda ne gibi işlemler yapıyorsunuz ?
21. İptal sonrası ücreti kaç gün içinde geri ödüyorsunuz ?
22. Ödeme şekilleriniz nelerdir ?
23. Banka dökümlerine hangi elemanlar erişebilir ?
24. Kullanıcı ödeme yaptıktan sonra ödemenin ulaşp ulaşmadığını kim kontrol ediyor ?
25. Faturalandırma işlemini nasıl yapıyorsunuz ?
26. İadeyi nasıl yapıyorsunuz ?

### **1.3 İŞ MODELİ:**

Yapılan soru bazlı ankete göre aşağıdaki iş modeli oluşturulmuştur.

- Firma web tabanlı tur satışı yapabileceği ve mobil uyumlu bir yazılım istiyor.
- Firma da 5 kişi çalışıyor, sistemi 4 kişi kullanacak.
- Yönetici, Kullanıcılar(Müşteriler), Muhasebe ve Organizatör olmak üzere 4 adet ana rol mevcut.
- Tur eklemek için organizatör ön araştırma yapar.
- Organizatör araştırması sonucu turun nereye yapılacağına karar verir, konaklama ve ulaşım gibi konular hakkında turun yapılacağı bölgede araştırma yapar. Tura katılacak kişilerin üst limitini, fiyatını ve tarihi belirler.
- Organizatör web sitesinde kendi profiline giriş yapar ve tur oluştur diyerek edindiği bilgilerle turu oluşturur. Admin site üstünden yollar.
- Admin organizatörün tur ekleme talebini görür. Firma sahibine danıştıktan sonra turu siteye ekler. Daha sonra düzenleyebilir, tarihi geçen turları silebilir.
- Kullanıcılar web sitesine kişisel bilgilerini girerek profillerini oluşturur.
- Kullanıcı adı ve şifrelerini kullanarak sisteme giriş yaparlar.
- Kullanıcı daha önceden site yöneticisi tarafından girilen tur paketlerini seçtiği tarih aralığına göre görüntüler.
- Gitmek istediği turu seçer ve satın al butonuna basar. Ödeme ekranına yönlendirilir.
- Ödeme ekranında kullanıcının kişisel bilgileri profilinden çekilir.
- Kullanıcı kredi kartı bilgilerini girer veya havale-eft yöntemiyle ödemesini gerçekleştirir.
- Muhasebeci, bankadan bilgileri alarak gerekli kontrolü yapar ve ödemeyi onaylar.
- Organizatör müşterinin işlemlerinin tamamlandığı onayını alır ve müşterinin ulaşım ve konaklama işlemlerini ayarlar. Organizatör bu işlemleri program üzerinden yapmaz.

- Eğer müşteri tur tarihi geçmeden iptal etmek isterse ödenen toplam miktara yüzde 10 kesinti uygulanır . Müşteri tur tarihinden 7 gün önceye kadar iptal talebinde bulunabilir.
- Kullanıcılar tura çıkmadan önce acenteye gelir ve servis yolcuyu alır. Seyahatleri başlar.

## **1.4 SÖZLÜK:**

**TURİZM:** Turizm, bir ülkeye ya da bir bölgeye yapılan turistik seyahatlerin ekonomik, sosyal ve kültürel olarak düzenlenmesi, organize edilmesi, önlemlerin alınması ve çalışmaların yapılması anlamına gelmektedir.

**TUR:** Türkiye'nin tarihi, doğal, kültürel, turistik değerlerinden en az birini tanıtmaya ve ulaştırmayı birlikte kapsayan, bu hizmetlerin dâhil olduğu tek bir fiyatla satılan veya satış taahhüdü yapılan ve hizmeti yirmi dört saatten kısa bir süreyi kapsayan veya konaklama içermeyen ticari faaliyet.

**ACENTA:** Bir kuruluşun mali veya ticari işlerini kazanç karşılığında yürüten ticarethanedir.

**BAYİİ:** Yetkili Satıcı İktisadi bir terim olarak, bir imalatçı ya da üreticinin malının satış yetkisini tek başına elinde bulunduran gerçek veya tüzel kişilik şeklinde tanımlanmaktadır.

**ORGANİZASYON:** Düzenlemek eylemidir.

**ORGANİZATÖR:** Organizatör; konser, sergi, seminer, konferans, lansman, festival gibi çeşitli etkinlikleri planlayan ve organize eden kişidir. Kamu ve özel sektör için veya bireysel hizmet verir. Kısıtlı zaman içinde organizasyonun, dekor, catering, eğlence, ulaşım, lokasyon, davetli listesi, ekipman gibi ayrıntılarını yönetir. Organizasyonun basılı yayın ve dijital ortamda tanıtımının yapılmasında rol alır.



**MUHASEBE** : Tüm kuruluşların mali nitelikteki işlemleri ve olayları para ile ifade edilmiş şekilde kaydeden, sınıflandıran, özetleyerek rapor eden ve sonuçlarını yorumlayan ve analiz eden bir bilim dalıdır.

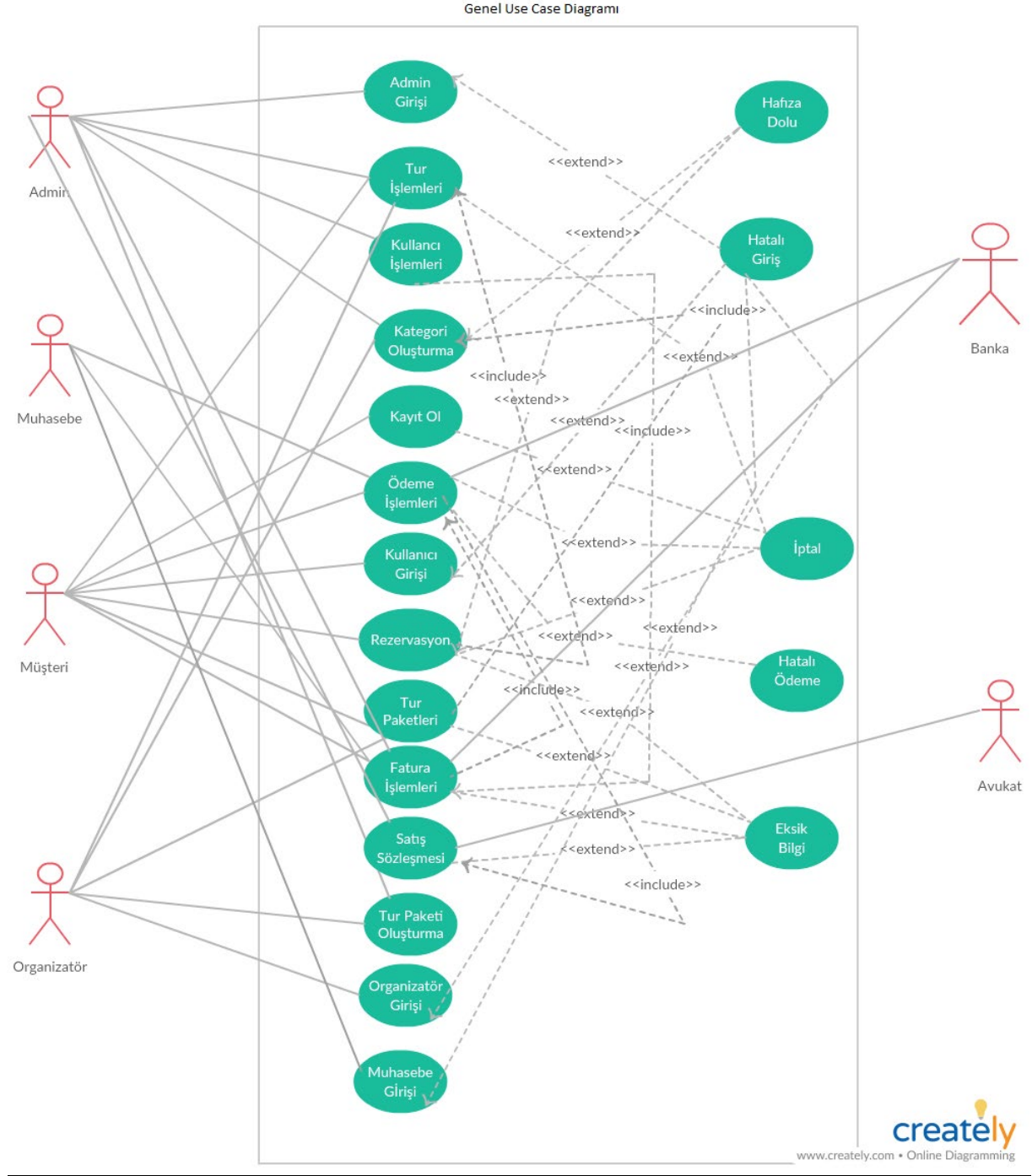
**HAVALE**: Havale yurt içinde veya yurt dışındaki bir alıcıya para gönderilmesi işlemidir.

**EFT**: Havale yurt içinde veya yurt dışındaki bir alıcıya para gönderilmesi işlemidir.

**ADMIN**: Admin; Yönetim, idare gibi anlamlara gelmektedir.

## 1.5 USE CASE DİAGRAMLARI

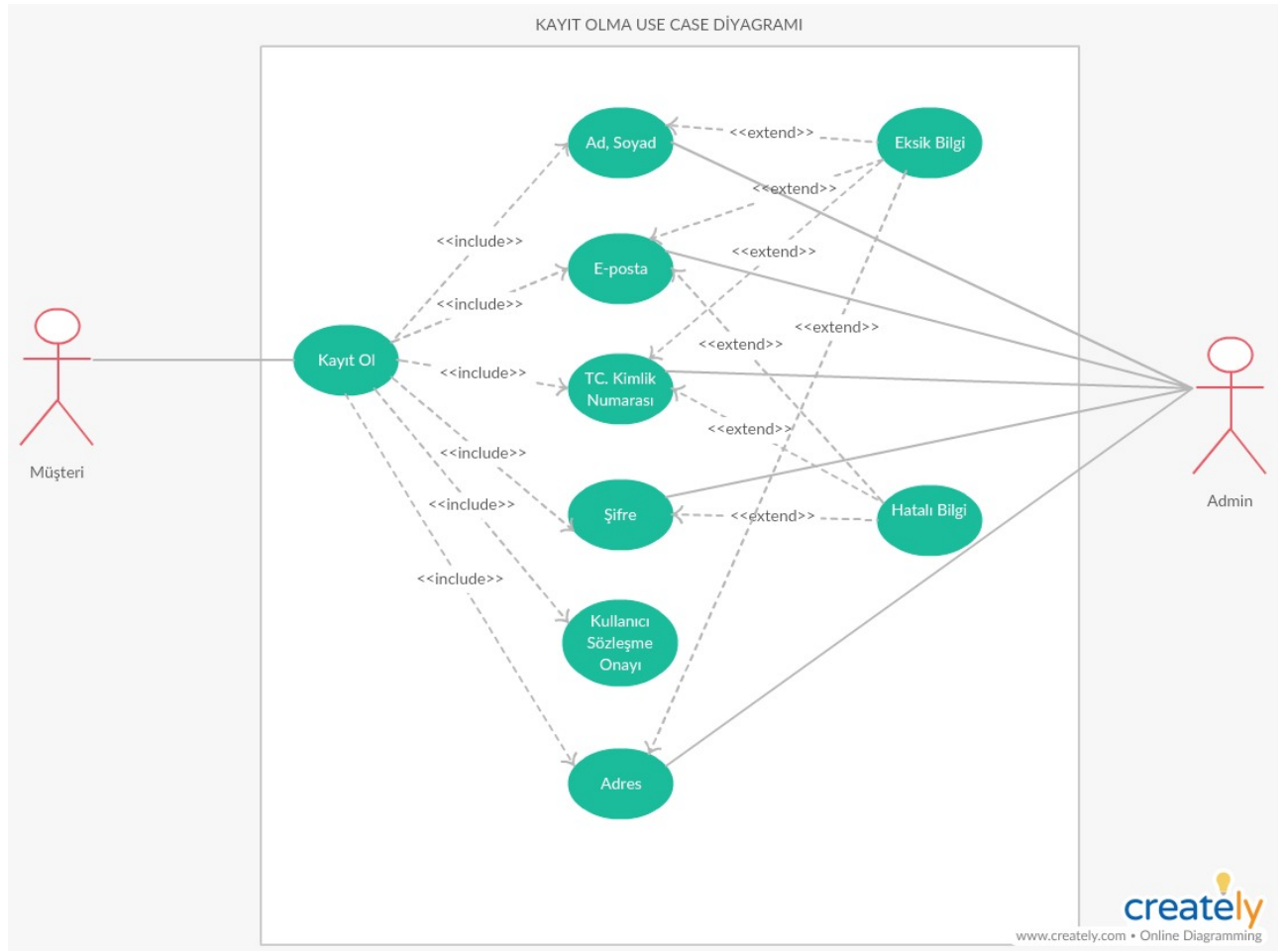
### Genel Use Case Diagramı:



Tüm aşamaların use case diagramları çizilmiş ve altlarına actors, entry conditions, exit conditions , flow of events ,exceptional cases bilgileri yazılmıştır. Üstteki diagramda genel çizim yapılmıştır.



## Kayıt Use Case Diagramı:



Kısa Tanım: Kullanıcı kaydında gerekli olan kişisel bilgiler alınmıştır .

Actors: Kullanıcı(Müşteri) , Admin.

Entry Condition: Kayıt ol butonuna basılması . Kayıt isteği oluşturulması.

Exit Condition: “Kayıt başarılı.” Mesajının alınması.

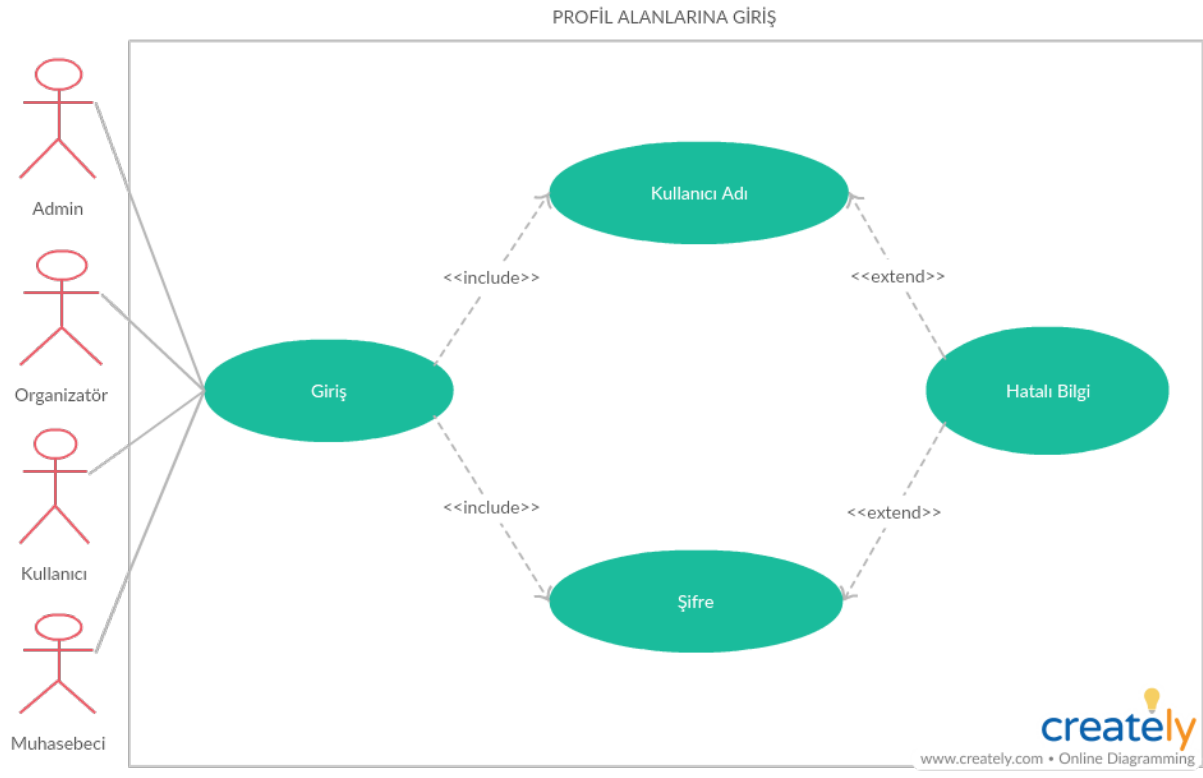
Flow of Events :

1. Ad, Soyad, e-posta, TC kimlik no , şifre, adres kayıt ekranına girilir.
2. Kullanıcı Sözleşmesi onaylanır.
3. Sisteme eklenir.
4. “Kayıt başarılı” Mesajı döner .

Exceptional Cases:

1. Hatalı Bilgi.
2. Eksik Bilgi.
3. İptal.

## Giriş Use Case Diagramı:



Kısa Tanım: Sessionlara göre kullanıcı girişi için tasarlanmıştır.

Actors: Kullanıcı(Müşteri),Admin, Organizatör , Muhasebeci.

Entry Condition: Giriş yap.

Exit Condition: “Giriş başarılı.” Mesajının alınması.

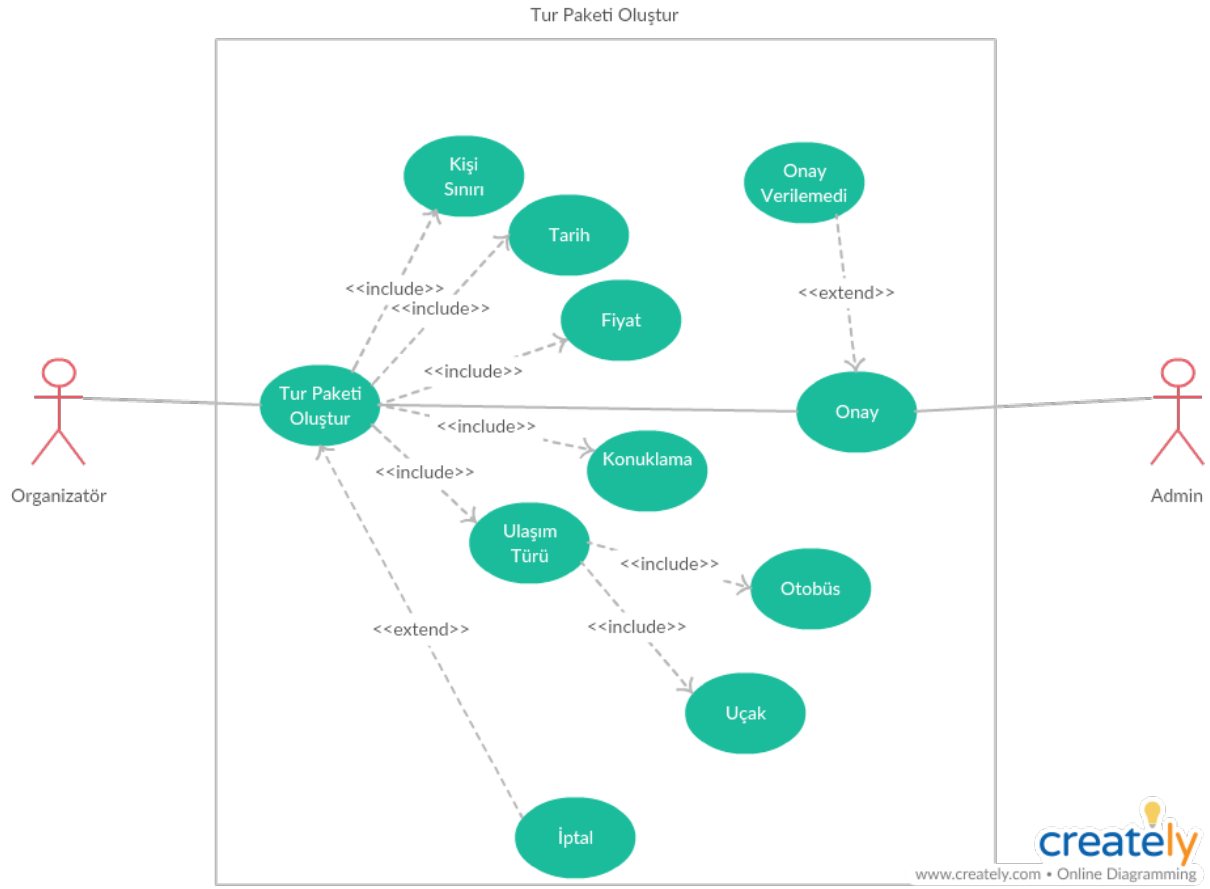
Flow of Events :

1. Kullanıcı Adı(email adresi) girilir.
2. Şifre Girilir.
3. Giriş butonuna basılır.

Exceptional Cases:

1-“Kullanıcı adı veya Şifre hatalı” mesajı döndürür.

## Tur Paketi Oluşturma Use Case Diagramı:



**Kısa Tanım:** Organizatör tarafından oluşturulan tur paketinin admin'e yollamasını ve onay beklemesini ifade eder.

**Actors:** Organizatör, Admin.

**Entry Condition:** Tur paketi oluştur.

**Exit Condition:** "Tur paketi isteği gönderildi." Mesajının alınması.

**Flow of Events:**

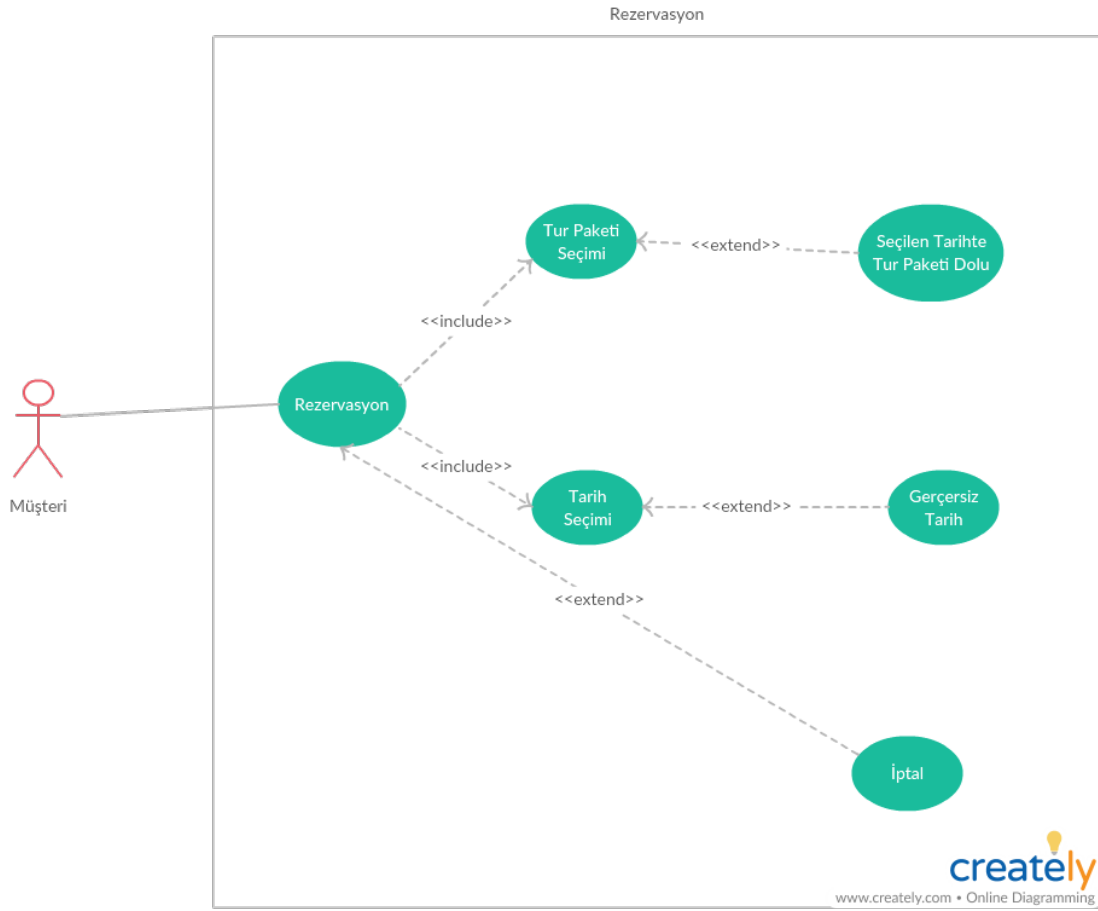
1. Tura katılacak kişi sayısını girilir.
2. Tur tarih aralığı girilir.
3. Fiyat girilir.
4. Konaklama yeri girilir.
5. Ulaşım türü girilir. (Uçak veya otobüs)
6. Admin onayına yollanır.

**Exceptional Cases:**

1-İptal etme.

2-Admin onay vermedi.

## Rezervasyon Yapma Use Case Diagramı:



Kısa Tanım : Rezervasyon işleminin basamaklarını gösterir.

Actors: Müşteri

Entry Condition: Rezervasyon yap.

Exit Condition: "Rezervasyon tamamlandı." Mesajı döndür.

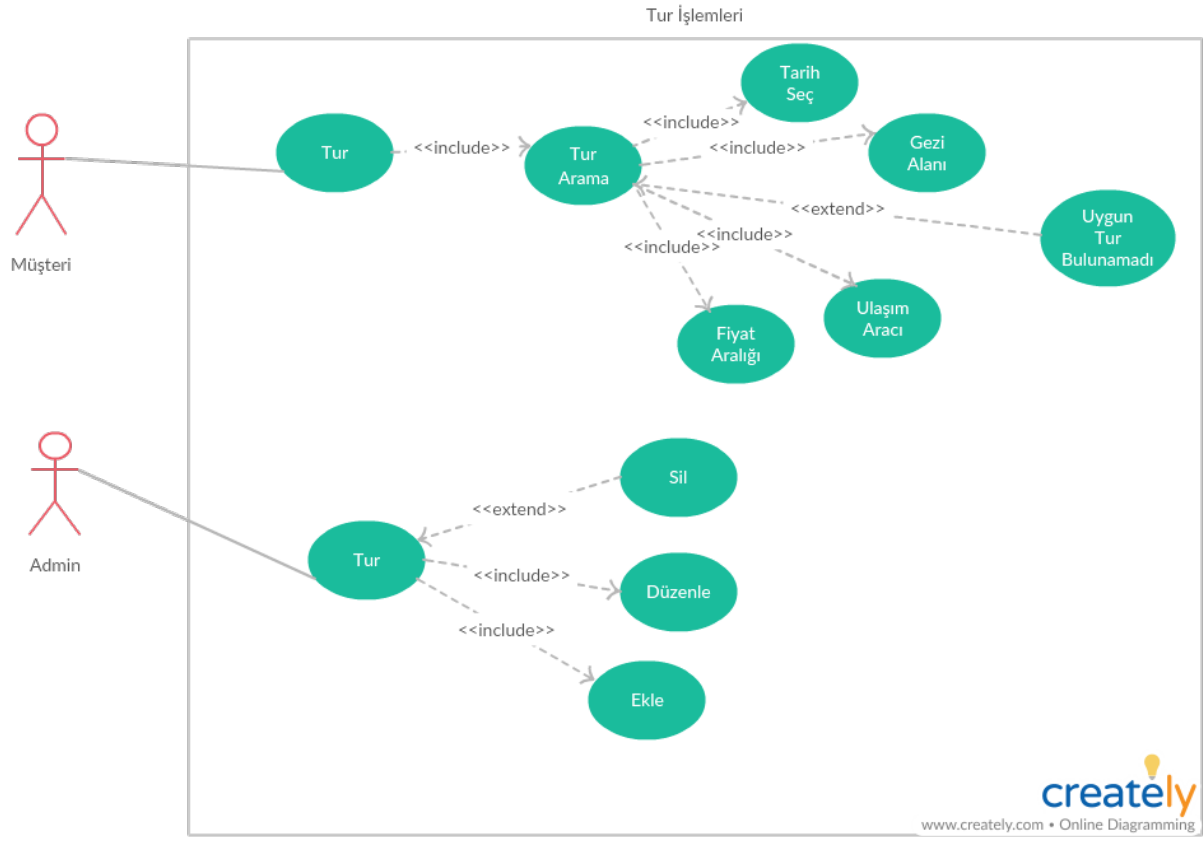
Flow of Events:

1. Rezervasyon sekmesine tıklanır.
2. Tarih aralığı seçilir.
3. Tur paketi seçilir.
4. "Rezervasyon tamamlandı." Mesajı dönderilir.

Exceptional Cases:

- 1-Seçilen tarihte tur paketi dolu.
- 2-Geçersiz tarih girildi.
- 3-İptal.

## Tur İşlemleri Use Case Diagramları :



**Kısa Tanım:** Müşterinin seçeneklere göre tur aramasını ve adminin tur işlemlerini yapmasını gösterir.

**Actors:** Müşteri, Admin.

**Entry Condition:** Müşterinin tur araması .

**Exit Condition:** “Tur bulundu . Listelendi “ Mesajının alınması .

**Flow of Events:**

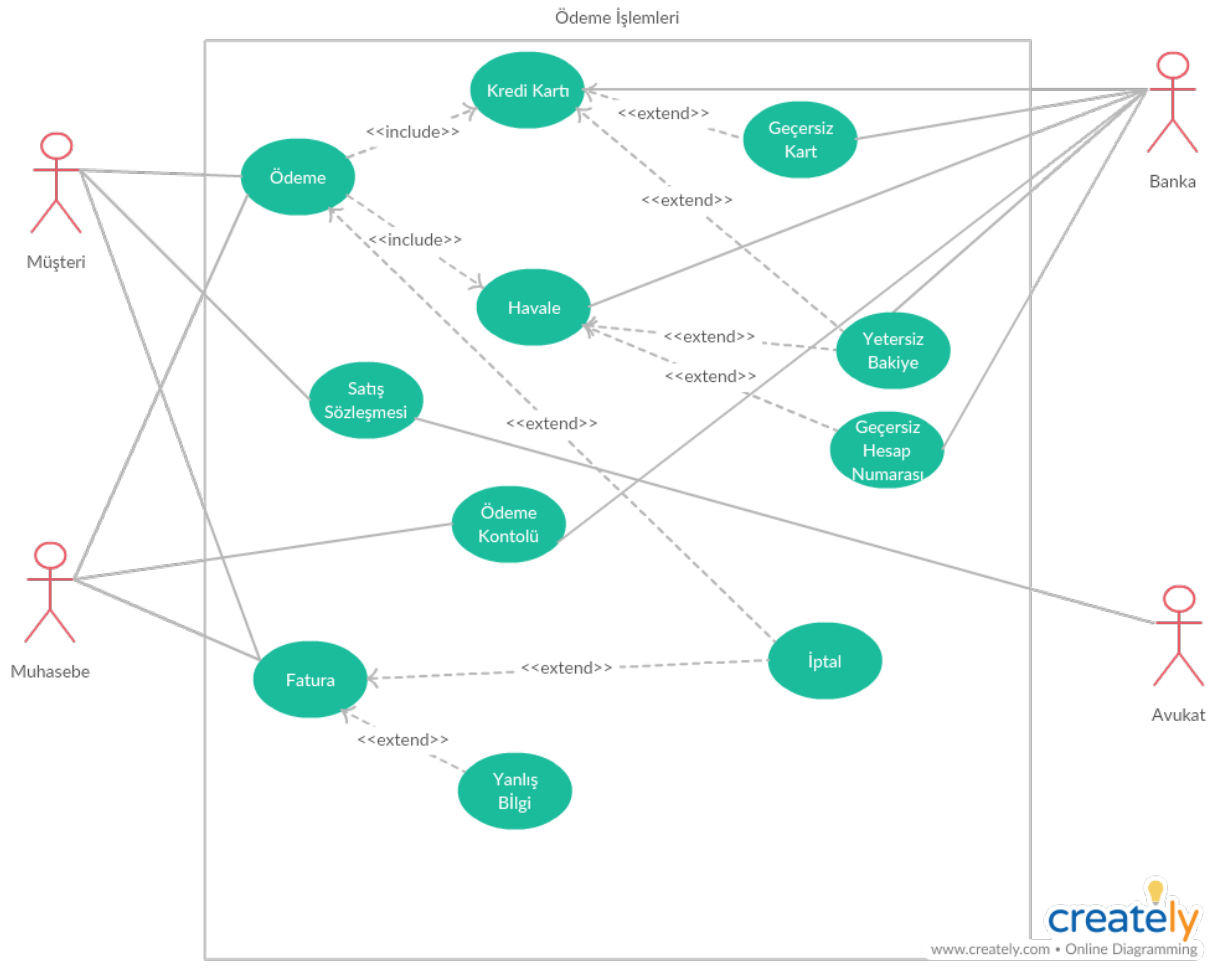
1. Tarih seçimi yapıldı.
2. Gidilecek yer seçildi.
3. Ulaşım çeşidi seçildi.
4. Fiyat aralığı belirlendi.
5. “Tur bulundu” Mesajı döner .
6. Ödeme sayfasına yönlendi.

**Exceptional Cases:**

1. “Uygun tur bulunamadı . “Mesajı döner.



## Ödeme Bölümü Use Case Diagramları :



Actors: Müşteri, Muhasebe, Banka, Avukat.

Entry Conditions: Ödeme yapılır ve fatura kesilir.

Exit Conditions: "Fatura kesildi , Ödeme başarılı." Mesajı Döner.

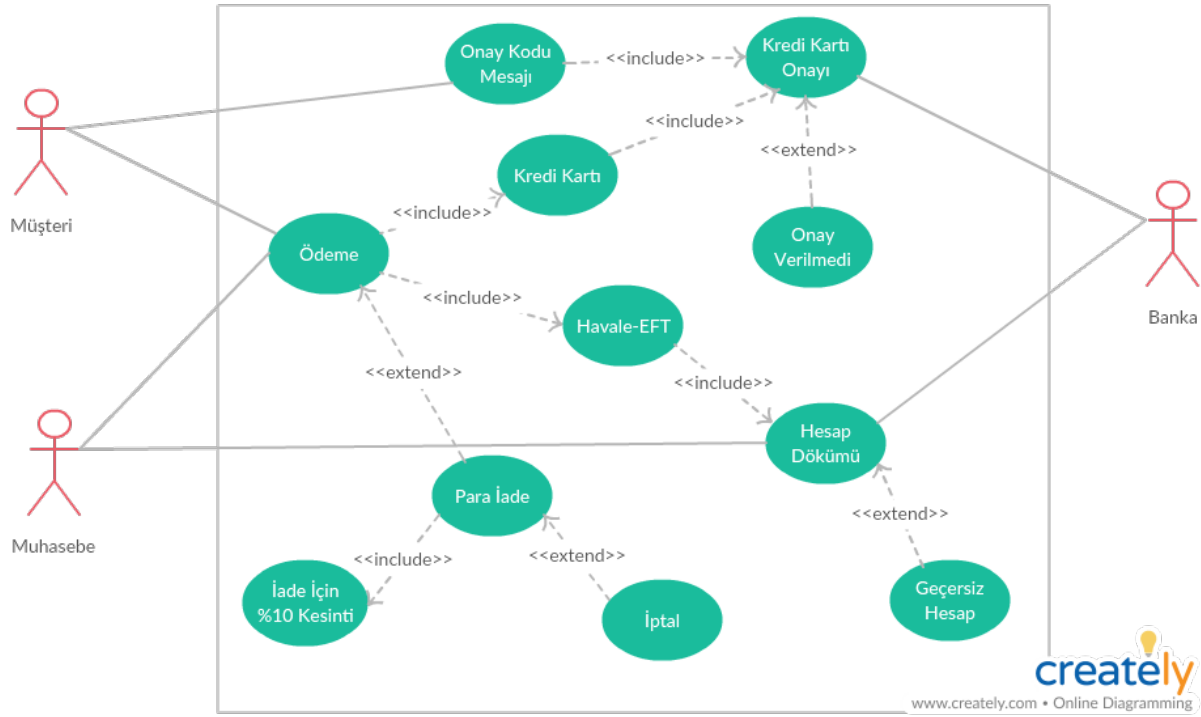
Flow of Events:

1. Ödeme sayfasına geçildi.
2. Ödeme yöntemi seçildi.
3. Avukat tarafından hazırlanan satış sözleşmesi görüntülendi.
4. Ödeme kontrol edildi.
5. Fatura kesildi.
6. "Fatura Kesildi, Ödeme Başarılı." Mesajı Döndü.

Exceptional Cases:

- 1-Geçersiz Kart Bilgisi.
- 2- Yetersiz Bakiye.
- 3-Geçersiz hesap numarası .
- 4-İptal.

## İade İşlemleri Use Case Diagramları:



**Kısa Tanım:**İade işlemlerinin basamaklarını gösterir . Yüzde 10 kesinti para iade balonuna bağlıdır.

**Actors:** Müşteri,Muhasebe, Banka.

**Entry Conditions:** Müşteriye geri ödeme yapılması.

**Exit Conditions:**"İade tamamlandı." Mesajı dönmesi.

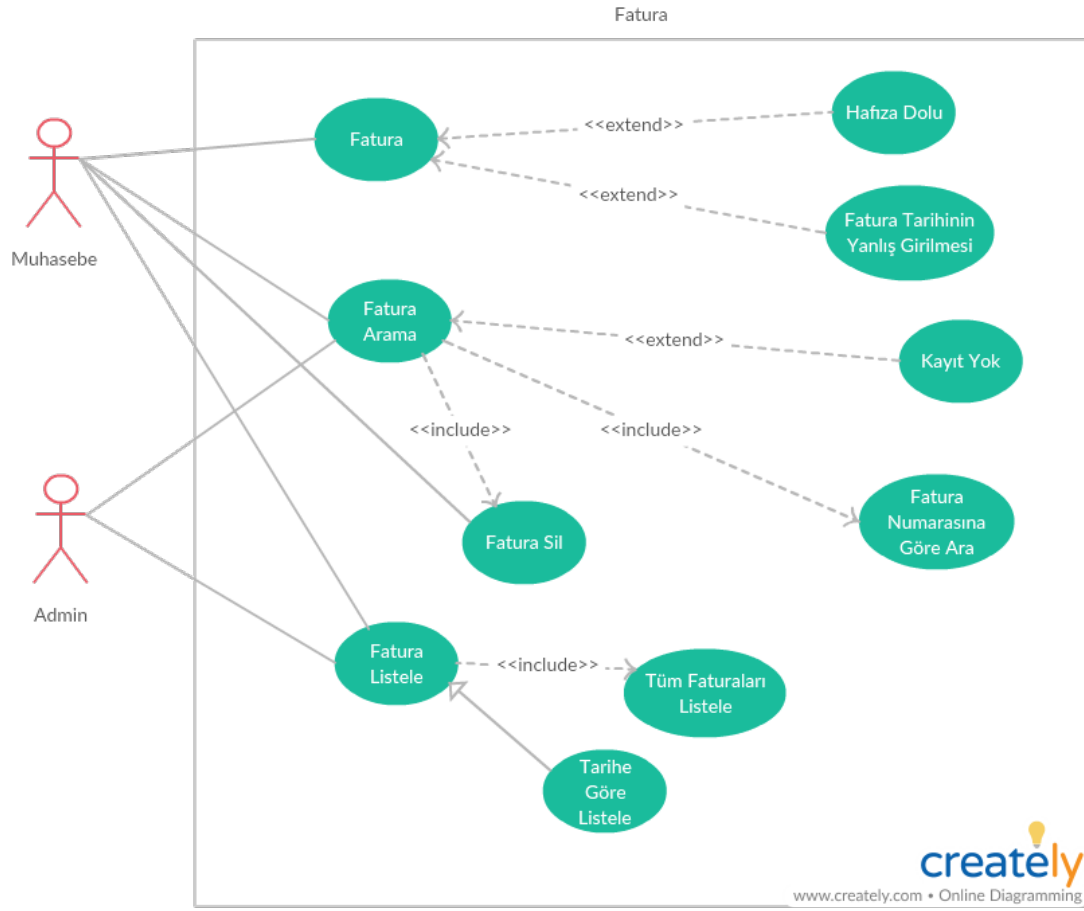
**Flow of Events :**

1. Ödeme sayfası açıldı .
2. Muhasebeciye para iade talebi yollar .
3. Muhasebeci hesap dökümüne bakar.
4. Para iadeye gidilir .
5. İptalde yüzde 10 kesinti uygulanır .
6. İade iptal.
7. "İade tamamlandı " Mesajı alınır .

**Exceptional Cases :**

1. İptal.
2. Geçersiz hesap dökümü.

## Fatura ve Muhasebe Bölümü Use Case Diagramları:



Kısa Tanım: Fatura işlemlerini gösteren bölümdür . Filtreleme seçenekleri bulunur . Fatura listeleme için kullanılır.

Actors: Muhasebe, Admin.

Entry Conditions: Fatura işlemlerini görüntüle.

Exit Conditions: "Fatura ile ilgili işlem tamamlandı" Mesajı dönder.

Flow of Events:

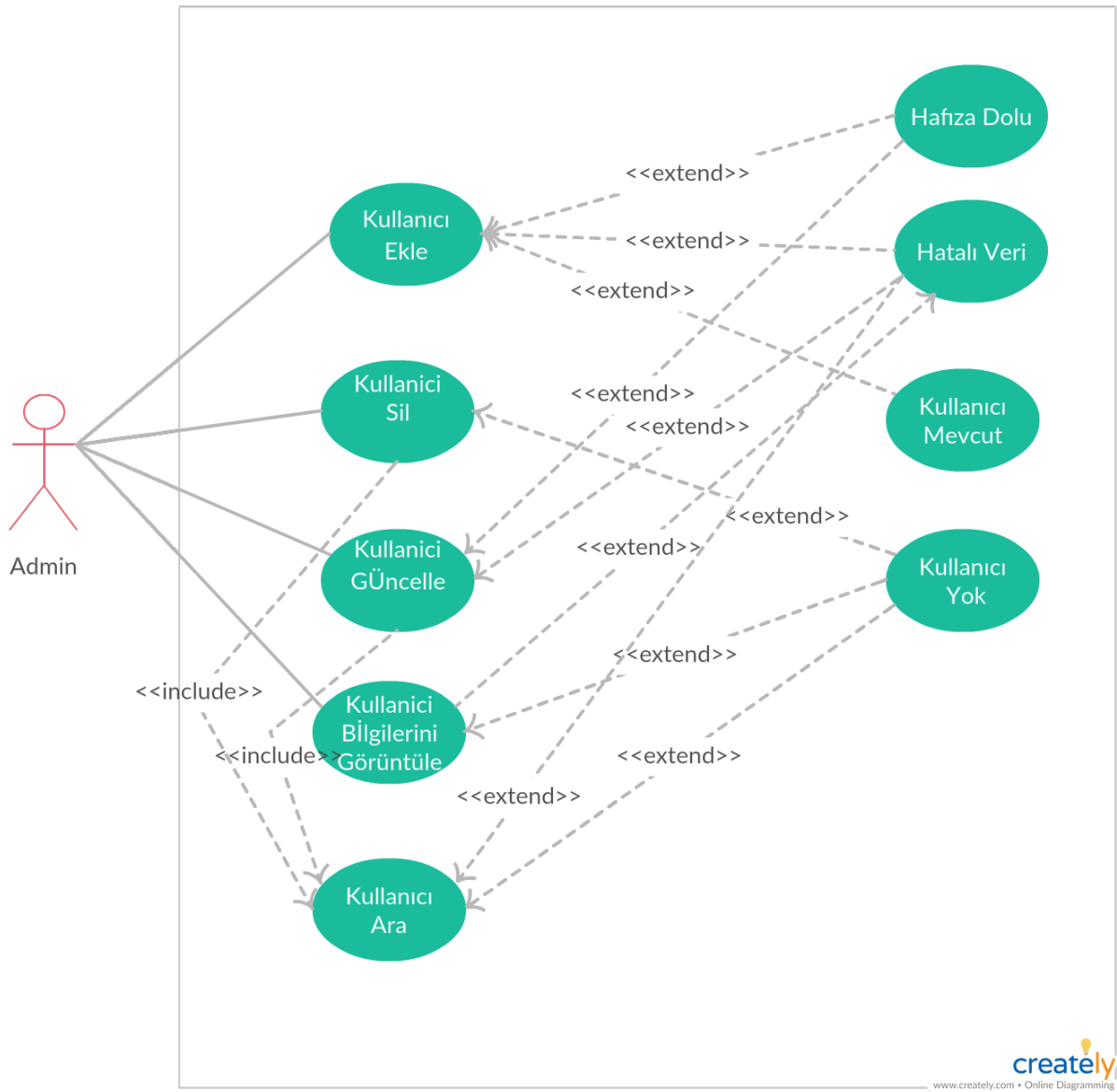
1. Fatura oluştur.
2. Fatura No'ya göre Fatura ara.
3. Faturaları filtrele ve listele.
4. Fatura Sil.
5. "Fatura Silindi " Mesajı dönder.

Exceptional Cases:

1. Hafıza Dolu.
2. Fatura tarihi yanlış
3. Kayıt yok.
4. İptal.

## Admin Kullanici İşlemleri Use Case Diagramı :

Admin Kullanici İşlemleri



Kısa Tanım: Adminin Kullanıcılar sınıfında yaptığı işlemleri gösterir.

Actors: Admin

Entry Conditions: Kullanıcı işlemlerine başla.

Exit Conditions: Kullanıcı işlemi başarıyla tamamlandı.

Flow of Events:

- 1- Kişi Ekle.
- 2- Kişi Sil.

- 3- Kullanıcı bilgi güncelle.
- 4- Kullanıcı bilgi görüntüle.
- 5- Kullanıcı ara.

Exceptional Cases:

- 1- Hafıza dolu.
- 2- Kullanıcı mevcut.
- 3- Hatalı veri.
- 4- Kullanıcı yok.

# Analyse Phase

### 2.1.1 Varlık Sınıflarının ve Özelliklerinin Belirlenmesi

#### VARLIK SINIFLARI



Kullanıcılar Sınıfı	Tur Paketi Yarat Sınıfı	Rezervasyon Sınıfı
KullanıcıBilgileriGetir	MaxKişiSayısı	TarihSec
KullanıcıEkle	TarihSec	TurPaketiSec
KullanıcıSil	Fiyat	RezervasyonTamamla
KullanıcıGuncelle	KonaklamaTürü	
KayıtOl	UlaşımTürü	
KullanıcılarListesi		

Admin Sınıfı	Tur İşlemleri	Fatura Sınıfı
TurEkle	TarihSec	FaturaID
TurSil	GeziYeriniGörüntüle	FaturaNo
TurDüzenle	Fiyat	TarihSec
TurAra	KonaklamaTürü	HesapAdi
TurOnayla	UlaşımTürü	HesapNo
	TurAra	
	TurListele	
	TurSatınAl	

Ödeme Sınıfı	Muhasebe Sınıfı	Giriş Sınıfı
KrediKartı Havale EFT SatışSoz.	Paralade  FaturaID  Fatura Ekle  Fatura Arama  Fatura Listele	KullanıcıAdı  Sifre

## KONTROL SINIFLARI

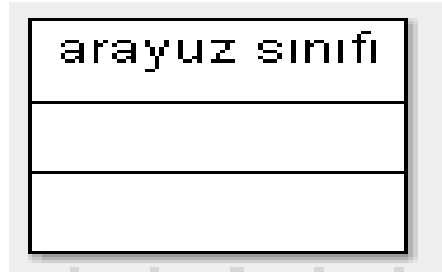
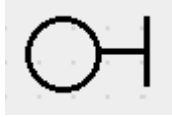


Kayıt Controller	Banka Controller	Satış Söz. Controller
Ad Soyad E-posta(kullanici adı) TC Tel NO Adres Şifre	KrediKartOnay  KrediKartıKontrolu  HesapKontorulu	Satış Söz.Onay (Checkbox)



Muhasebe Controller	Admin Controller	Giriş Controller
ÖdemeKontrol ParaiadeKontrol HesapDokumuKontrol FaturaKontrol	TurPaketiOnayKontrol  RezervasyonKontrol	KullaniciAdi  Sifre

**Sınır Sınıfı**



## **2.1.2 CRC KARTLARI:**

Kullanıcılar Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Kullanıcı bilgileri alınır. 2- Kullanıcı ekler. 3- Kullanıcı siler. 4- Kullanıcı günceller. 5- Kullanıcılar bu classta listelenir.
<u>İş Birliği</u> KayitController GirisController

Giris Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Kullanıcı adı ve şifresi alınır.
<u>İş Birliği</u> <u>Kullanıcı</u> KayitController GirisController

TurPaketi Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Tarih belirlenir. 2- Mekân belirlenir. 3- Kişi sayısı belirlenir. 4- Ulaşım aracı seçilir. 5- Fiyat belirlenir. 6- Onay alındıktan sonra tur paketi oluşturulur.
<u>İş Birliği</u> TurIslemleri Admin

Rezervasyon Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Tarih belirlenir. 2- Tur paketi seçimi yapılır. 3- Rezervasyon tamamlanır.
<u>İş Birliği</u> TurPaketi Turİslemleri

Admin Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Tur ekleme/silme/düzenleme/arama işlemleri yapılır. 2- Kullanıcı bilgilerini getiririr. 3-Kullanıcı ekler. 4-Kullanıcı siler 5-Kullanıcı günceller. 6- Tur onaylaması yapar.
<u>İş Birliği</u> Kullanıcılar Turİslemleri TurPaketi

Odeme Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Kullanıcının seçtiği seçenek doğrultusunda ödeme seçimi yapılır. 2- Ödeme sözleşmesi onaylanır. 3- Ödeme yapılır. 4- Fatura oluşturulur.
<u>İş Birliği</u> Fatura Muhasebe Kullanıcı Banka

Muhasebe Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Kullanıcı seçimine göre para iadesi yapar. 2- Fatura ekler. 3- Fatura ID sine göre ya da tarihe göre faturaları arar. 4- Faturaları listeler.
<u>İş Birliği</u> Fatura Kullanıcı Banka

Banka Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Kredi kartı onayı yapar. 2- Para havalesi yapar.
<u>İş Birliği</u> Fatura Muhasebe Odeme BankaController

KayıtController Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Kullanıcının girdiği adı/soyadı/şifresi/e-posta/TC. kimlik numarası/telefon numarası/adresi alınır. 2- Eksik ve hatalı bilgi kontrol edilir. 3- Kullanıcı sisteme kayıt edilir.
<u>İş Birliği</u> Kullanici GirisController

BankaController Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Kredi kartı kontrolü yapılır. 2- Kredi kartı onaylanır. 3- Hesap dökümü kontrol edilir.
<u>İş Birliği</u> Kullanici Odeme Banka

SatisSozlesmesiController Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Satış sözleşmesinin onaylanıp onaylanmadığını kontrol eder.
<u>İş Birliği</u> Kullanici Odeme Banka

GirisController Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Girilen kullanıcı adı ve şifresi sistemde kontrol edilir.
<u>İş Birliği</u> KayitController Giris

AdminController Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Tur paketinin onayını kontrol eder. 2- Rezervasyon kayıtlarını kontrol eder.
<u>İş Birliği</u> Admin Rezervasyon TurPaketi

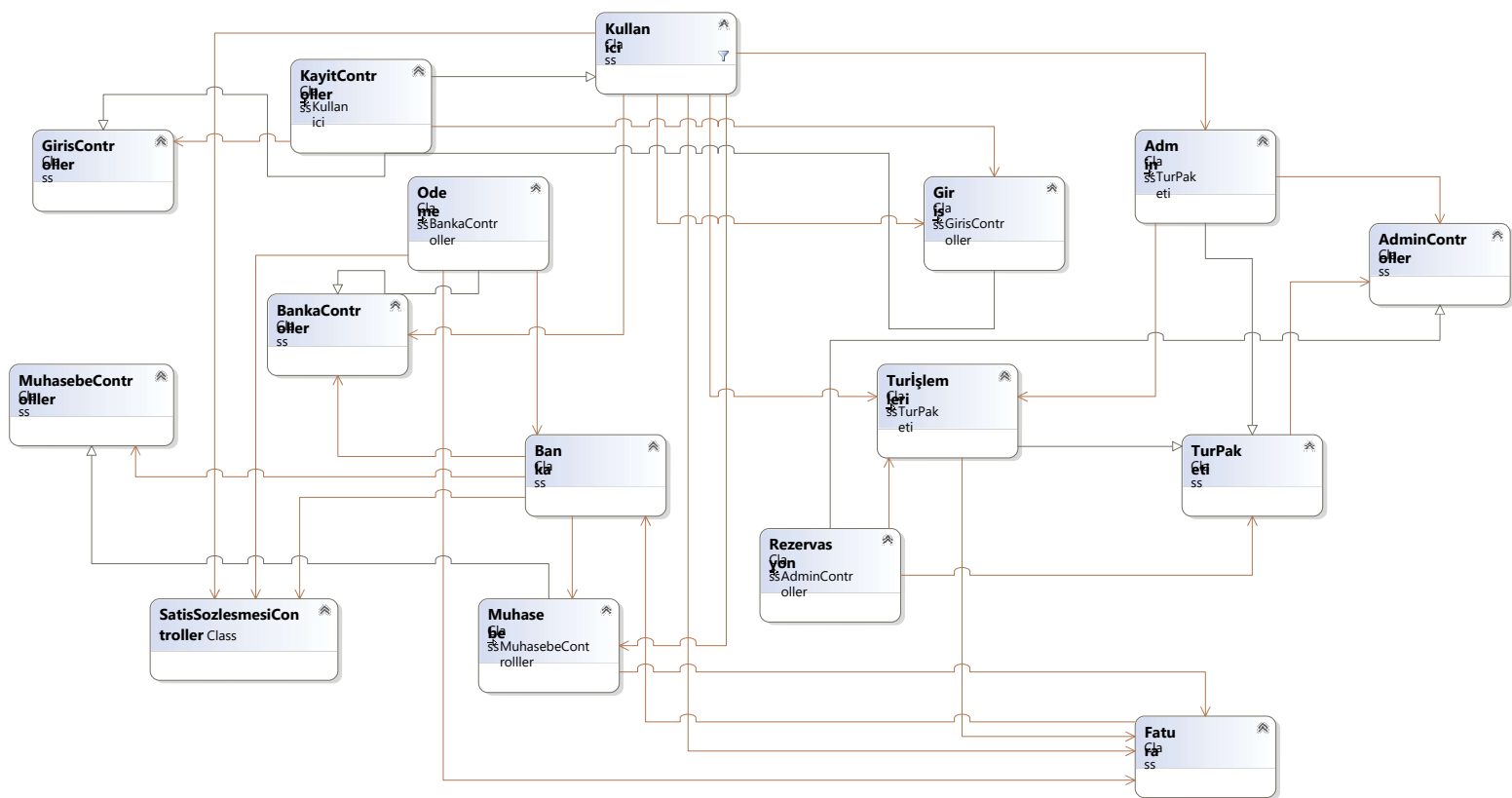
MuhasebeController Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Ödeme kontrolü yapar. 2- Para iadesinin kontrolünü yapar.
<u>İş Birliği</u> Muhasebe Banka

Turlislemleri CRC Kartı
<u>Sorumlulukları</u> 1- Tur aranır ve seçilir. 2- Tarih seçilir. 3- Gezi alanı ve ulaşım aracı seçilir. 4- Fiyat aralığı seçilir. 5- Ödeme sayfasına yönlendirilir.
<u>İş Birliği</u> Kullanici TurPaketi

Fatura Sınıfı CRC Kartı
<u>Sorumlulukları</u> 1- Ödeme yapılır. 2- Yeni fatura oluşturulur. 3- ID numarası ve fatura numarası veri tabanında atanır. 4- Kullanıcı bilgileri çekilir. 5- Tarih, ödeme sınıfından çekilir. 6- Fatura kesilir.
<u>İş Birliği</u> Turlislemleri Odeme Kullanici

### 2.1.3 CLASS

### DIYAGRAMI :

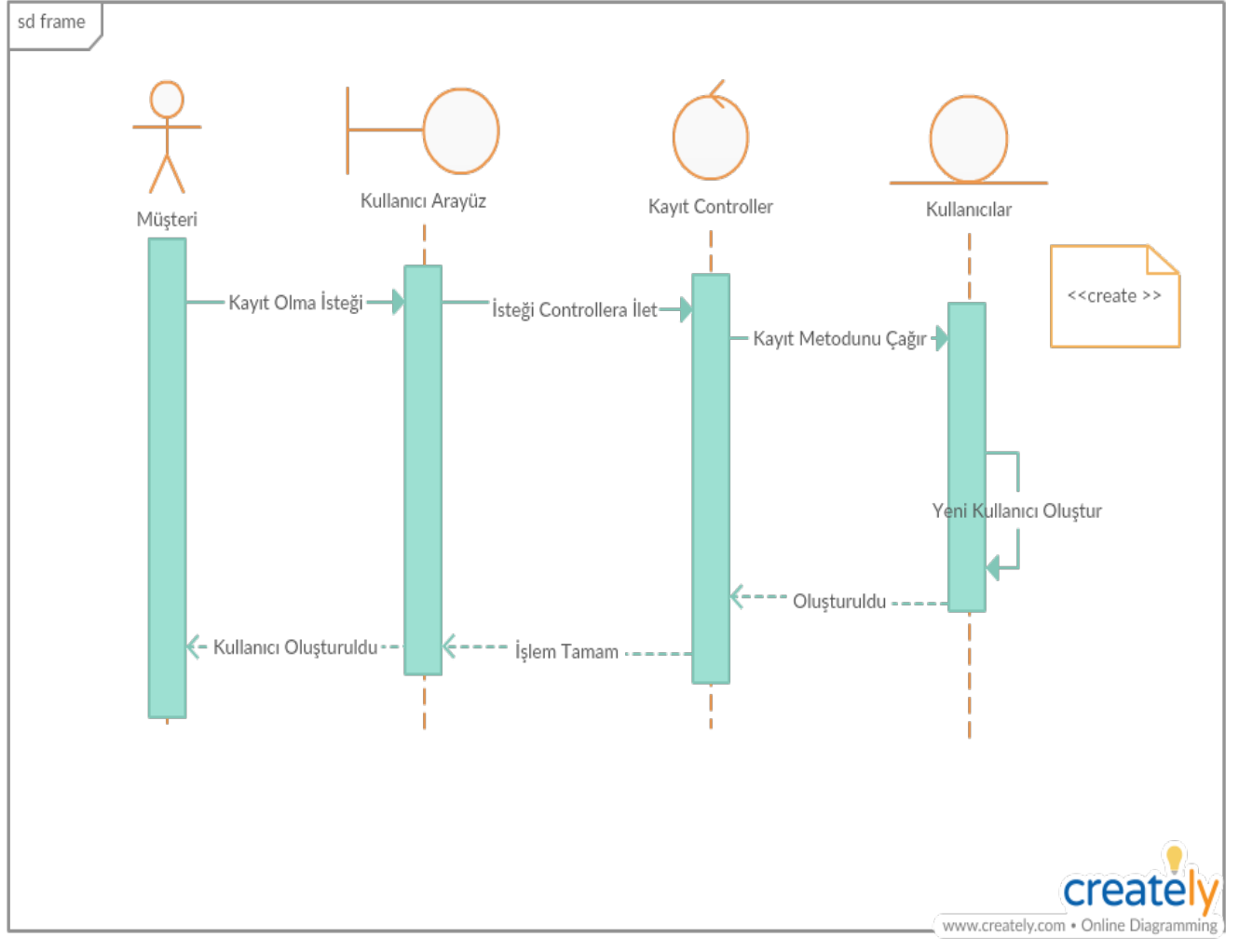


## **2.2 DİNAMİK MODELLEME**

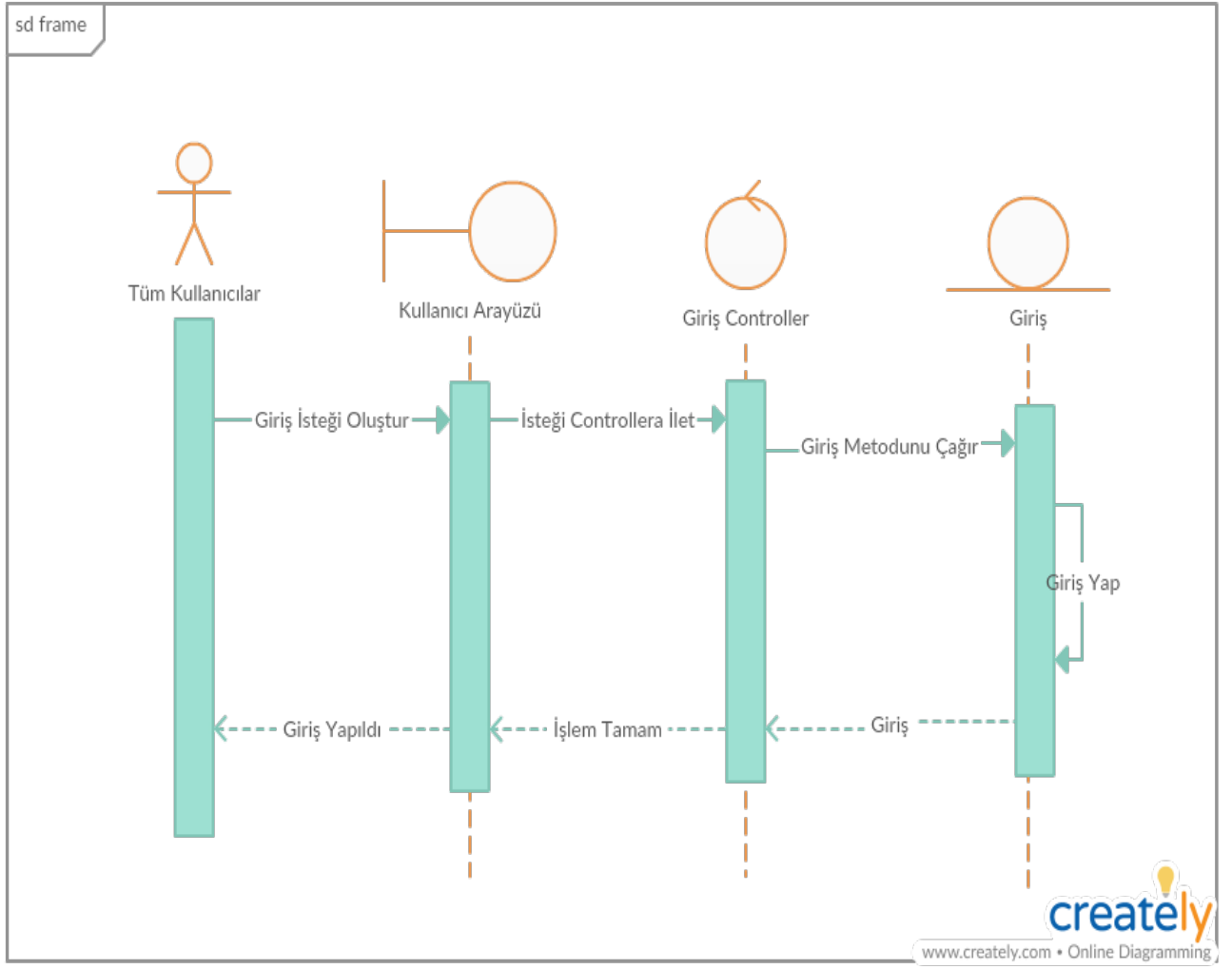


## 2.3 USE-CASE REALIZATION

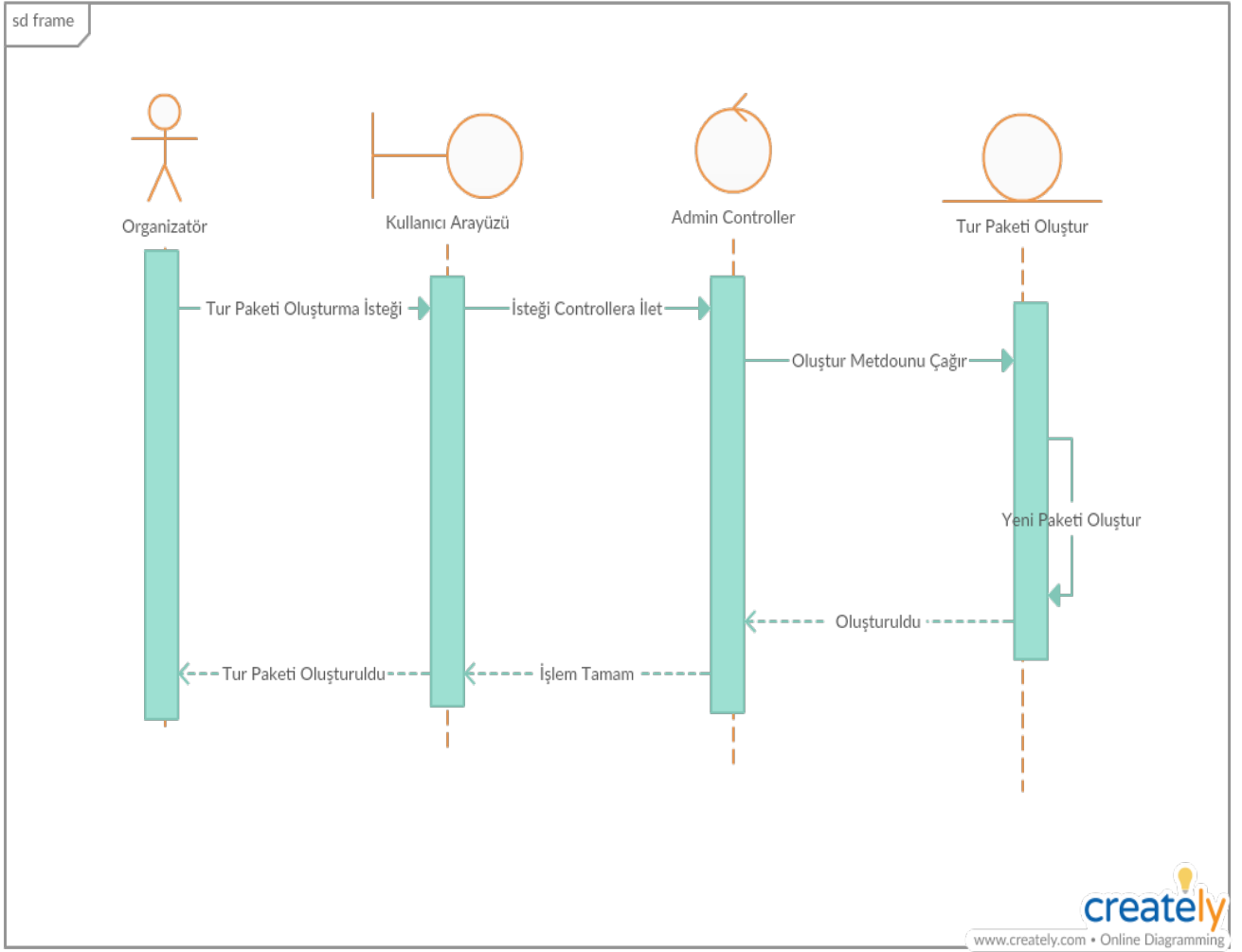
### KAYIT OL SEQUENCE DİYAGRAMI



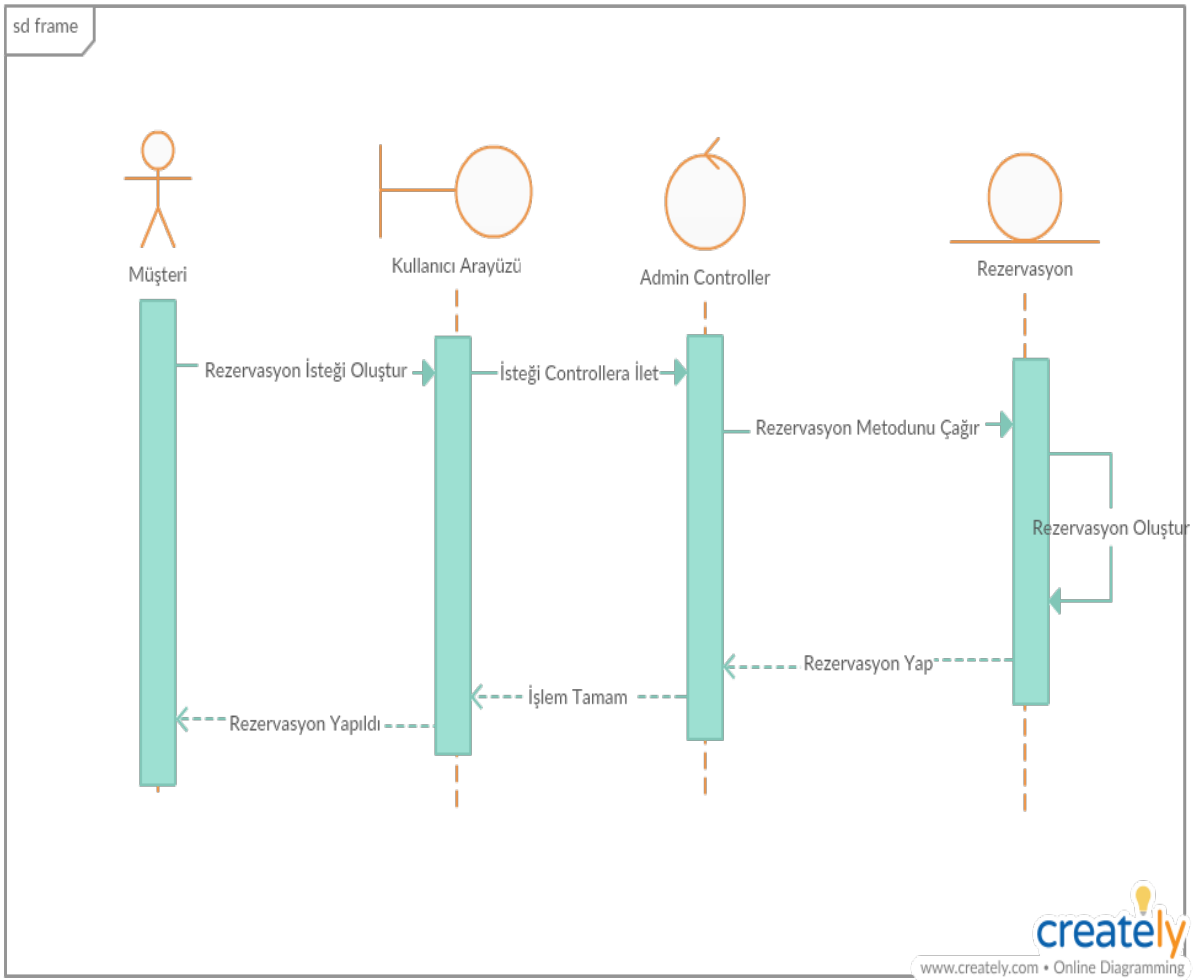
## GİRİŞ YAP SEQUENCE DİYAGRAMI



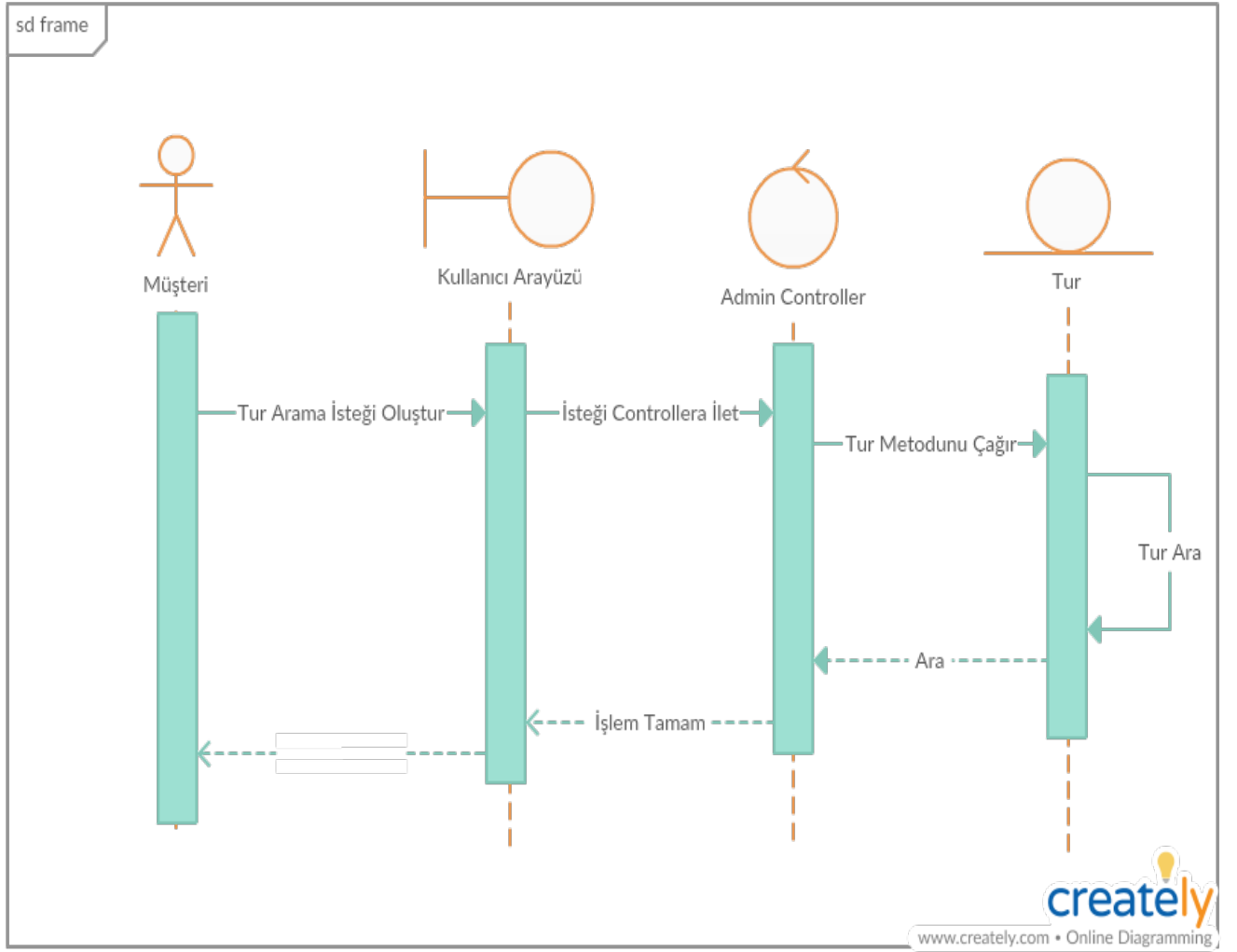
## TUR PAKETİ OLUŞTUR SQ. D.



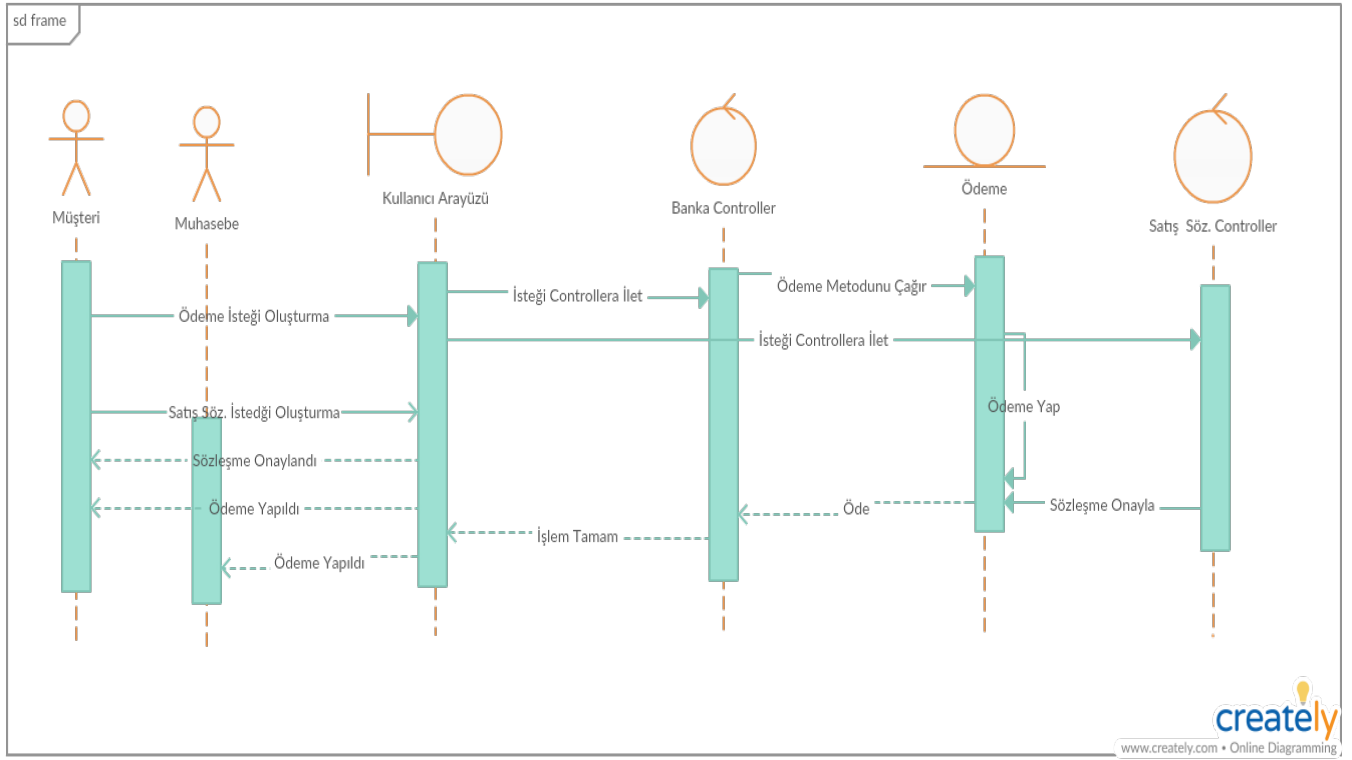
## REZERVASYON SQ. D.



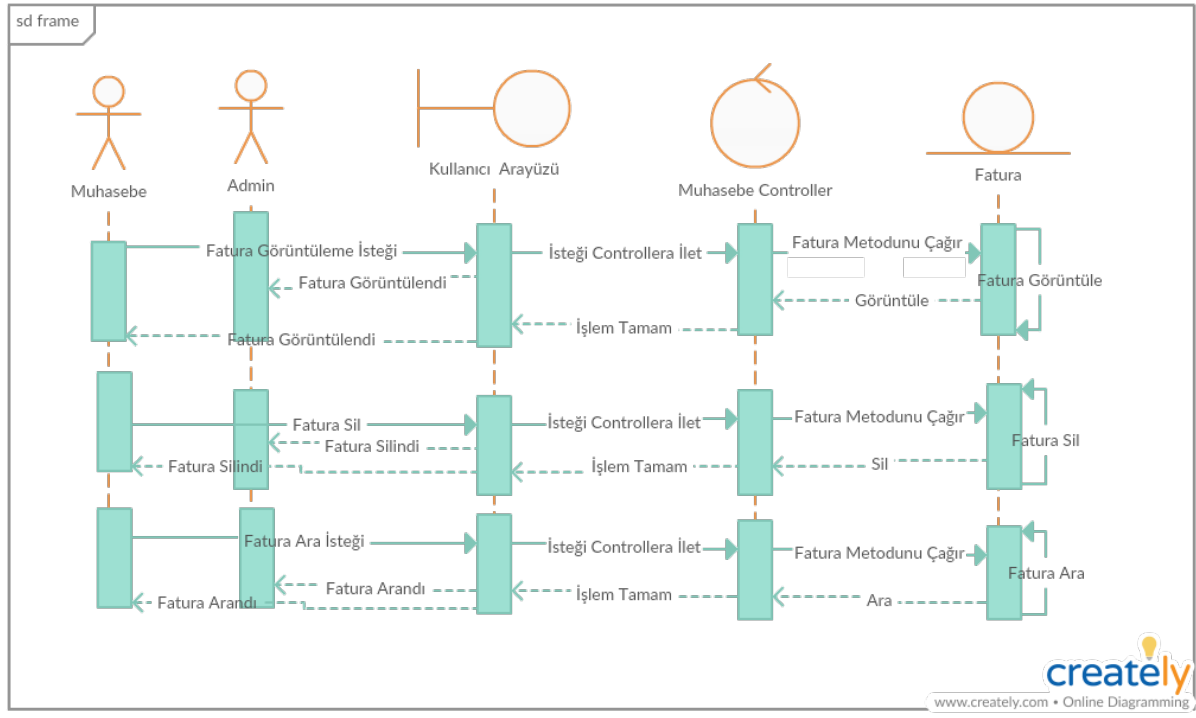
## TUR İŞLEMLERİ SQ D.



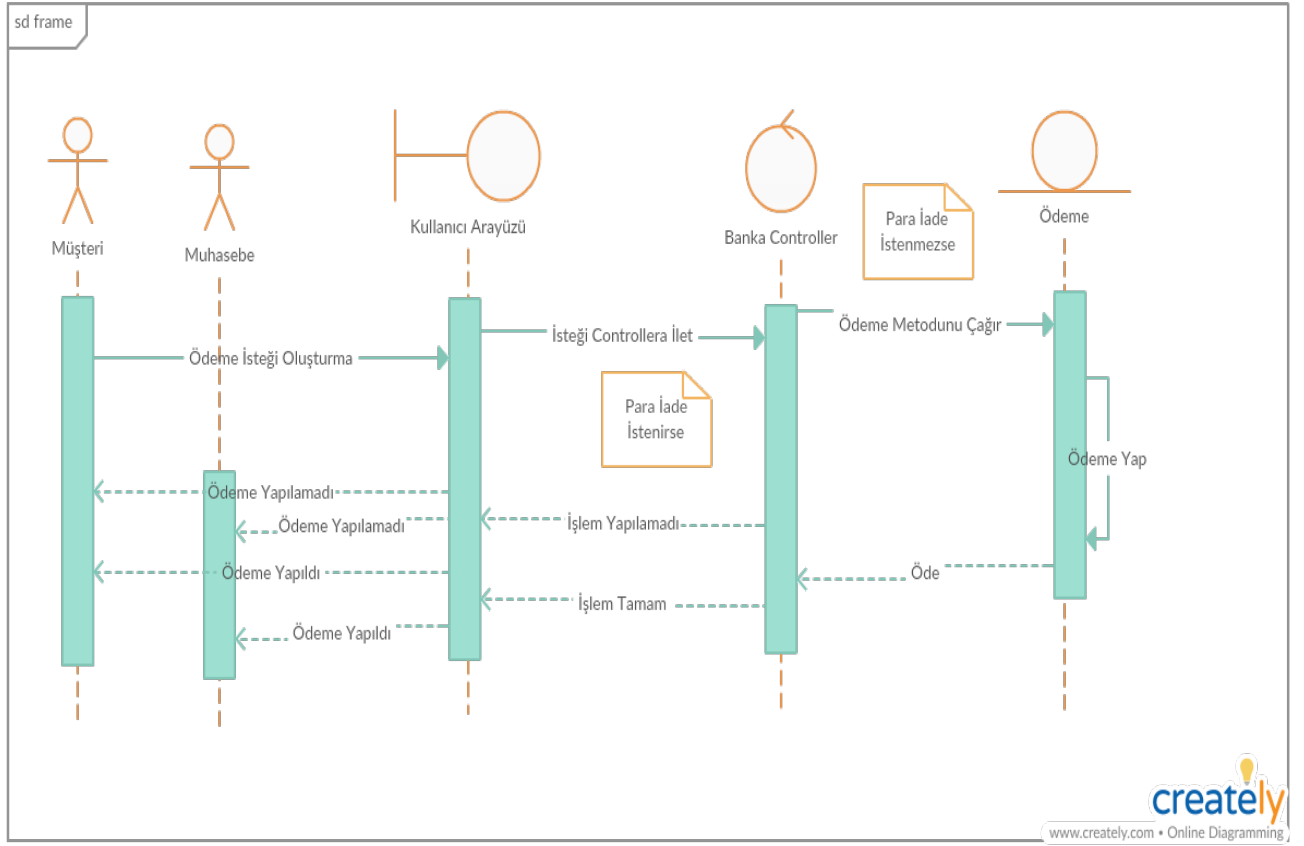
## ÖDEME SQ D.



## FATURA SQ D.



## MUHASEBE SQ D.



## **2.4 Yazılım Proje Yönetim Planı:**

### **2.4.1 Geniş Özet:**

#### **2.4.1.1 Amaç, Kapsam ve Hedefler:**

Projenin amacı bir turizm acentasının online satış yapabilmesi için gerekli zemini hazırlamak , müşteri kayıtlarının daha hızlı bir şekilde daha güvenli bir ortamda tutulmasını sağlamaktır. Yapılan ödemelerin banka desteği ile güvenliğini arttırmak ve e – fatura sistemiyle kullanıcının vakitten tasarruf etmesi hedeflenmektedir . Sistemimiz bilgi girişi ve bilgiye ulaşım esnasında harcanan zamandan ve maliyetten en yüksek verimi almak için tasarlanmıştır.

Projemizi hazırlarken ki başlıca hedeflerimiz:

- Projenin belirlenen tarihte teslimi.
- Müşterinin ihtiyacı olan tüm işlemlerin yerine getirilmesi.
- Sistemde minimum bug olması .

### **2.4.2 Proje Çıktıları:**

- Yazılımın kendisi.
- Ürün kabul esnasında kullanıcıyı bilgilendirme.
- Proje dokümanları



### **2.4.3 Takvim ve Bütçe Özeti**

- Projenin tahmin edilen bütçesi 7.000 liradır.
- Proje bitiminden itibaren yıllık bakımı 1.000 lirayı aşmayacaktır.
- Projenin teslim alındığı tarih: 19.11.2018
- Projenin tahmini teslim tarihi: 21.12.2018

### **2.4.4 Risk Yönetim Planı:**

Projede var olan riskler şu şekildedir:

- Projenin belirlenen zaman aralığında bitirilip teslim edilememesi.  
Proje çalışanları olarak alınan ortak kararda gerekirse fazla mesai yaparak bu riskin önüne geçildi.
- Projenin gerçekleştirilmesi esnasında müşteriden yeni taleplerin gelmesi.  
Projemizi geliştirirken kullanılan metot sayesinde bu riskin önüne geçildi.

### **2.4.5 Metadoloji ve Araçlar:**

#### **Geliştirme Metodolojisi:**

Yazılım geliştirilirken kullanılan metot Çevik (Agile) Yazılım Geliştirme Metodu'dur.

### **Araçlar:**

- Takım Bilgisayarlarındaki Yazılımlar
  - Microsoft Windows 8.1
  - Microsoft Windows 10
  - Microsoft Windows 7
  - MacOS Mojave
  - Microsoft Visual Studio 2017
  - Google Chrome Tarayıcı
  - Internet Explorer Tarayıcı
  - Microsoft Office 365
- Müşteri
  - Microsoft Windows 10

## **2.4.6      Kabul Planları:**

Müşteriden herhangi bir ek istek gelmediği ve yazılımı almayı kabul ettiği takdirde, gerekli dokümanlar ile birlikte yazılımın müşteriye teslim edilmesi. Teslim sırasında yazılım ile ilgili gerekli bilgilerin müşteriye verilmesi ve kurulumun yapılmasına karar verildi.

## **2.5      Yazılım Test Planı:**

### **2.5.1. Kapsam:**

#### **2.5.1.1. Bilgilendirme:**

Yazılımın test aşamasındaki amacımız gerçekleştirmekte olduğumuz yazılımın istenilen işlevleri yerine getirip getiremediğini anlamak ve bu aşamada karşımıza çıkabilecek olası hataların önüne geçmektir.

#### **2.5.1.2. Sistem Özellikleri:**

- Sistem Marka/Model: MSI AEGIS
- İşletim Sistemi: Microsoft Windows 10
- Ram: 32 GB
- Harddisk: 1 TB
- SSD:500 GB
- İşlemci: Intel I7 7700K
- Ekran Kartı: Nvidia GEFORCE GTX 1060 6 GB
- Anakart: MSI M7 GAMING PCI-E

#### **2.5.2. YAZILIM TEST ORTAMI:**

##### **2.5.2.1 Kullanılan Donanım:**

Kullanılan donanım yukarıda özellikleri belirtilen donanımdır.

##### **2.5.2.2. Katılımcı Kuruluşlar:**

- Katılımcı kuruluş bulunmamaktadır. Tüm aşamalar kendi ekibimiz tarafından hazırlanıp yürütülmüştür.

### **2.5.2.3      Personel:**

- Yazılımın test aşamasında çalışan personel listesi aşağıdaki gibidir;
  - Can KARABULUT
  - Melike SARI
  - Bekir ÇİFTLER
  - Hasan BAŞARA

### **2.5.2.4      Uygulanan Testler:**

- Ana Test
- İşlevsel Testler
- Kontrol Testleri
- Optimizasyon Testleri

## **2.5.3TEST BİLGİLERİ:**

### **2.5.3.1      Test Seviyeleri:**

- Gereksinim seviyesi
- Analiz Seviyesi
- Tasarım Seviyesi
- Gerçekleştirim Seviyesi
- Bakım Seviyesi

#### **2.5.3.2      Genel Test Durumları:**

- Test aşamalarında herhangi aksi bir durumla karşılaşılmamıştır.

#### **2.5.3.3      Testin Uygulanması :**

- Test aşamalarının tamamı kendi ekibimiz tarafından yapılmıştır .

#### **2.5.3.4      Veri Kaydı, İndirgemesi ve Analizi:**

- Veri kaydı ve analiz işlemini kendi yazılım ekibimiz yapmış olup sonuçları değerlendirmiştir.

# DESIGN PHASE

### 3.1 Sınıfların İç Detaylarının Belirlenmesi:

#### Kullanıcı Classı (Public):

<u>Sr.NO</u>	<u>İsim</u>	<u>Data Type</u>	<u>Constrait</u>	<u>İçeren Fonksyon</u>
<u>1</u>	KULLANICI ID	INT	Primary Key	Kullanıcılar KullaniciEkle();  Kullanıcılar KullaniciSil();  Kullanıcılar KullaniciGuncelle();
<u>2</u>	AD	Varchar(10)	Not null	Kullanıcılar KullaniciEkle();  Kullanıcılar KullaniciGuncelle();
<u>3</u>	SOYAD	Varchar(10)	Not null	Kullanıcılar KullaniciEkle();  Kullanıcılar KullaniciGuncelle();
<u>4</u>	TCNO	Numeric(18, 0)	Not null	Kullanıcılar KullaniciEkle();  Kullanıcılar KullaniciGuncelle();
<u>5</u>	TELNO	Numeric(18, 0)	Not null	Kullanıcılar KullaniciEkle();  Kullanıcılar KullaniciGuncelle();
<u>6</u>	ADRES	nvarchar(MAX)	Not null	KullaniciEkle();

<b><u>7</u></b>	EPOSTA(Kullanici Adi)	nvarchar(50)	Not null	Kullanıcılar KullaniciEkle();  Kullanıcılar KullaniciGuncelle();
<b><u>8</u></b>	SIFRE	Numeric(18, 0)	Not null	Kullanıcılar KullaniciEkle();  Kullanıcılar KullaniciGuncelle();

**Giriş Classı(Public):**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	EPosta(KullaniciAdi)	nvarchar(50)	Not null	Giris GirisYap();
<b><u>2</u></b>	Şifre	Numeric(18, 0)	Not null	Giris GirisYap();



**Tur Oluştur Classı(Public):**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	TurID	INT	Primary Key	TurOlustur YeniTur();
<b><u>2</u></b>	MaxKisiSayisi	Numeric(18, 0)	Not null	TurOlustur YeniTur();
<b><u>3</u></b>	TarihSec	date	Not null	TurOlustur YeniTur();
<b><u>4</u></b>	Fiyat	Numeric(18, 0)	Not null	TurOlustur YeniTur();
<b><u>5</u></b>	KonaklamaTuru	Varchar(10)	Not null	TurOlustur YeniTur();
<b><u>6</u></b>	UlasimTuru	Varchar(10)	Not null	TurOlustur YeniTur();

**Tur İşlemleri(Public) : Tur TurListele(); , Tur TurAra(); Tur SatınAl();**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	TurID	INT	Primary Key	Tur TurListele(); Tur TurAra(); Tur SatınAl();
<b><u>2</u></b>	GeziYeri	nvarchar(MAX)	Not null	Tur TurListele(); Tur TurAra();
<b><u>3</u></b>	Fiyat	Numeric(18, 0)	Not null	Tur TurListele(); Tur TurAra(); Tur SatınAl();
<b><u>4</u></b>	KonaklamaTürü	nvarchar(MAX)	Not null	Tur TurListele(); Tur TurAra();
<b><u>5</u></b>	UlaşımTürü	varchar(10)	Not null	Tur TurListele(); Tur TurAra();
<b><u>6</u></b>	TarihSec	date	Not null	Tur TurListele(); Tur TurAra();

**Fatura Class (Public) : FaturaEkle(); FaturaBul(); :**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	FaturaID	INT	Primary Key	Fatura FaturaEkle();  Fatura FaturaBul();
<b><u>2</u></b>	FaturaNo	Numeric(18, 0)	Not null	Fatura FaturaEkle();  Fatura FaturaBul();
<b><u>3</u></b>	Fiyat	Numeric(18, 0)	Not null	Fatura FaturaEkle();  Fatura FaturaBul();
<b><u>4</u></b>	TarihSec	date	Not null	Fatura FaturaEkle();  Fatura FaturaBul();
<b><u>5</u></b>	HesapAdi	nvarchar(MAX)	Not null	Fatura FaturaEkle();  Fatura FaturaBul();
<b><u>6</u></b>	HesapNo	Numeric(18, 0)	Not null	Fatura FaturaEkle();  Fatura FaturaBul();

### **Ödeme Sınıfı(Protected) :**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	KrediKartı	nvarchar(50)	Primary Key	Odeme KrediKartiOdeme();
<b><u>2</u></b>	Havale	Numeric(18, 0)	Not null	Odeme HavaleOdeme();
<b><u>3</u></b>	EFT	Numeric(18, 0)	Not null	Odeme EFTOdeme();
<b><u>4</u></b>	SatışSoz.	binary(2)	Not null	Fatura FaturaEkle(); Fatura FaturaBul();

### **Muhasebe Class(Protected) :**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	ParaladeMiktari	Numeric(18, 0)	Not Null	Muhasebe Paralade();
<b><u>2</u></b>	FaturaID	int	Primary Key	Fatura FaturaBul();
<b><u>3</u></b>	OdemeOnay	binary(2)	Not Null	Muhasebe OdemeOnay();
<b><u>4</u></b>	KullaniciVerileri	String	-	Kullanıcı verilerinden çekiliyor.

**Kayıt Controller Classı(Public):**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	AD	Varchar(10)	Not null	Kullanıcılar VeriKontrol();
<b><u>2</u></b>	SOYAD	Varchar(10)	Not null	Kullanıcılar VeriKontrol();
<b><u>3</u></b>	TCNO	Numeric(18, 0)	Not null	Kullanıcılar VeriKontrol();
<b><u>4</u></b>	TELNO	Numeric(18, 0)	Not null	Kullanıcılar VeriKontrol();
<b><u>5</u></b>	ADRES	nvarchar(MAX)	Not null	Kullanıcılar VeriKontrol();
<b><u>6</u></b>	EPOSTA(Kullanıcı Adı)	nvarchar(50)	Not null	Kullanıcılar VeriKontrol();
<b><u>7</u></b>	SIFRE	Numeric(18, 0)	Not null	Kullanıcılar VeriKontrol();

**Giriş Controller Classı (Public):**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	EPosta(KullanıcıAdı)	nvarchar(50)	Not null	Giris GirisControl();
<b><u>2</u></b>	Şifre	Numeric(18, 0)	Not null	Giris GirisControl();

### **Muhasebe Controller(Protected) Class :**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	ParaladeMiktari	Numeric(18, 0)	Not Null	Muhasebe ParaladeKontrol ();
<b><u>2</u></b>	FaturaID	int	Primary Key	Muhasebe FaturaSiraKontrol();
<b><u>3</u></b>	OdemeOnay	binary(2)	Not Null	Muhasebe OdemeOnay();

### **Banka Controller(Private) Class :**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	KrediKartOnay	Bool	Not Null	Banka K.KartiOnayKontrol();
<b><u>2</u></b>	KrediKartiKontrolu	Bool	Not Null	Banka K.KartiGecerlimiKontrol ();
<b><u>3</u></b>	HesapKontorulu	Bool	Not Null	Banka BakiyeKontrol();

### **Satis Söz. Controller(Protected)**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	SatisSoz.	Bool	Not Null	SatisSoz. OnayVarMı();

### **Admin Controller(Protected) Class :**

<b><u>Sr.NO</u></b>	<b><u>İsim</u></b>	<b><u>Data Type</u></b>	<b><u>Constrait</u></b>	<b><u>İçeren Fonksyon</u></b>
<b><u>1</u></b>	TurPaketiOnayKontrol	Bool	Not Null	Muhasebe ParaladeKontrol ();
<b><u>2</u></b>	RezervasyonKontrol	Bool	Primary Key	Muhasebe FaturaSiraKontrol();

### 3.2 PSEUDO CODES

```
4. public class Kullanicilar
5.
6. private int KullaniciID;
7. private string ad;
8. private string soyad;
9. private string TCNO;
10.    private string adres;
11.    private int TelNo;
12.    private string email;
13.
14.    public void KullaniciGuncelle()
15.    {
16.        // TODO - implement
        Kullanicilar.KullaniciGuncelle
17.        throw new
        System.InvalidOperationException();
18.    }
19.
20.    public void KullaniciSil()
21.    {
22.        // TODO - implement
        Kullanicilar.KullaniciSil
23.        throw new
        System.InvalidOperationException();
24.    }
25.
26.    public void KullaniciEkle()
27.    {
28.        // TODO - implement
        Kullanicilar.KullaniciEkle
29.        throw new
        System.InvalidOperationException();
30.    }
31.
```



```

32.     public string GetKullaniciBilgileri()
33.     {
34.         string KullaniciBilgiler = ""; //
        Bilgileri Çeker.
35.         return KullaniciBilgiler;
36.
37.
38.         public void SepetOlustur()
39.         {
40.             // TODO - implement SepetOlustur
41.             throw new
        System.InvalidOperationException();
42.         }
43.
44.         public void TurAra()
45.         {
46.             // TODO - implement Tur.TurAra();
47.             throw new
        System.InvalidOperationException();
48.         }
49.
50.         public void TurSatinAl()
51.         {
52.             // TODO - implement
        Kullanici.SatinAl
53.             throw new
        System.InvalidOperationException();
54.         }
55.
56.
57.     public class Giris
58.     {
59.         public bool KullaniciAdiKontrol
60.
61.         public bool SifreKontrol
62.     }
63.

```

```

64.     //Kullanici bilgilerini kontrol ediyor.
65.
66.     public class Tur
67.     {
68.         private string GeziAlani;
69.
70.
71.         public void TurAra()
72.         {
73.             throw new
74.             System.InvalidOperationException();
75.         }
76.
77.         public void GirilenLokasyonEslesti()
78.         {
79.             // TODO - implement
80.             Tur.GirilenLokasyonEslesti
81.             throw new
82.             System.InvalidOperationException();
83.         }
84.     }
85.     public class TurSatinAl
86.     {
87.         private int TurID;
88.         private int KullaniciID;
89.         private string ulasim;
90.         private string konaklama;
91.
92.         public string TurBilgileriniGetir()
93.         {
94.             string ulasimbilgisi = ""; //
95.             Bilgiyi getir.
96.             return ulasimbilgis;
97.         }
98.
99.         public void TurIptal()
100.        {

```

```

97.          // TODO - implement Tur.TurIptal
98.          throw new
    System.InvalidOperationException();
99.      }
100.
101.      public void IsConfirmed()
102.      {
103.          throw new
    System.InvalidOperationException();
104.      }
105.
106.  }
107.
108.  public class Odeme
109.  {
110.      private int TurID;
111.      private double Fiyat;
112.
113.      public void OdemeSekli()
114.      {
115.          // TODO - implement
    Odeme.OdemeSekli
116.          throw new
    System.InvalidOperationException();
117.
118.      public void OdemeKontrol()
119.      {
120.          // TODO - implement
    Odeme.OdemeKontrol
121.          throw new
    System.InvalidOperationException();
122.      }
123.  }
124.
125.  public class Fatura : Odeme
126.  {
127.      private double Fiyat;

```

```
128.     }
129.
130.     public class ParaIade : Odeme
131.     {
132.         private string KullaniciAdi;
133.         private int bankaID;
134.     }
135.
136.     public class KrediKarti : Odeme
137.     {
138.         private int KartNumarasi;
139.         private string SonKullanmaTarihi;
140.         private string CVCKod;
141.     }
142.
143.     public class Ulasim
144.     {
145.         private int ulasimID;
146.         private string from;
147.         private string GeziAlani;
148.         private string Tarih;
149.
150.         public class UlasimDetay
151.         {
152.             private string durum;
153.             private double TurFiyat;
154.             private string ulasimturu;
155.
156.         }
157.
158.         public class Konaklama
159.         {
160.             private int KonaklamaID;
161.             private string GeziAlaniAdresi;
162.             public int TurFiyat;
163.         }
164.         public class KonaklamaDetay
```

```
165.      {
166.          private double TurFiyat;
167.          private int TurID;
168.          private string GeziAlaniAdresi;
169.      }
170.
171.      public class Fatura
172.      {
173.          protected int FaturaID;
174.          protected int FaturaNo;
175.          public int Fiyat;
176.          public string Tarih;
177.      }
```

### 3.3 FORM ARAYÜZ TASARIM

Ana Sayfa

Kayıt İşlemleri

Giriş

Tur İşlemleri

Tur Paketleri

Fatura İşlemleri

Banka İşlemleri

Muhasebe İşlemleri

Kullanıcı Adı:

Şifre:

Giriş

Ana Sayfa

Kayıt İşlemleri

Giriş

Tur İşlemleri

Tur Paketleri

Fatura İşlemleri

Banka İşlemleri

Muhasebe İşlemleri

Ad:

Soyadı:

E-posta:

TC kimlikno:

Telefon No:

Adres:

Kayıt Ol

Vazgeç

Ana Sayfa

Kayıt İşlemleriGirişTur İşlemleriTur PaketleriFatura İşlemleriBanka İşlemleriMuhasebe İşlemleri

Tur Ekle

Tur Sil

Tur Arama

Tur

Ana Sayfa

Kayıt İşlemleriGirişTur İşlemleriTur PaketleriFatura İşlemleriBanka İşlemleriMuhasebe İşlemleri

Kredi Kart:

Havale Yap:

Hesap Kontrolü: ☐ Doğru ☐ Yanlış

Onayla

Ana Sayfa

Kayıt İşlemleriGirişTur İşlemleriTur PaketleriFatura İşlemleriBanka İşlemleriMuhasebe İşlemleri

Satın Alma İşlem İptal Edildi

Fatura ID:

Fatura Ara:

Havale:

Fatura Ekle:

Para İade

Faturaları Listele

Ana Sayfa

Kayıt İşlemleriGirişTur İşlemleriTur PaketleriFatura İşlemleriBanka İşlemleriMuhasebe İşlemleri

Fatura ID:

Fatura Numarası:

Fatura Tarihi: 20 Aralık 2018 Perşembe

Ödeme Şekli:

Faturayı Yazdır



Kayıt İşlemleri Giriş Tur İşlemleri Tur Paketleri Fatura İşlemleri Banka İşlemleri Muhasebe İşlemleri

Tarih: 20 Aralık 2018 Perşembe

Mekan:

Kişi Sayısı:

Ulaşım Aracı:

Fiyat:

Onayla