

# OTEL YÖNETİM SİSTEMİ

## ANALİZ RAPORU

bu icindekiler dogru yapalim bence, oylesine yaptigimiz kadin biliyor---

### İÇİNDEKİLER

<b>1. GİRİŞ</b>	<b>4</b>
1.1 Otel Yönetim Sistemi Tanımı	4
1.2 Projenin Amacı	4
1.3 Projenin Kapsamı	4
1.4 Projenin Hedefleri	5
<b>2. SİSTEM GEREKSİNİMLERİ</b>	<b>6</b>
2.1 Fonksiyonel Gereksinimler	6
2.1.1 FR1 – Oda Yönetimi	6
2.1.2 FR2 – Müşteri Yönetimi	7
2.1.3 FR3 – Rezervasyon Yönetimi	7
2.1.4 FR4 – Check-in / Check-out Yönetimi	8
2.1.5 FR5 – Sistem Kontrolleri	8
2.2 Fonksiyonel Olmayan Gereksinimler	7
2.3 Sistem Kısıtları	8
<b>3. SİSTEM MODELLERİ</b>	<b>9</b>
3.1 Aktörler	9
3.1.1 Resepsiyon Görevlisi	9
3.2 Kullanım Durumu Modeli (Use Case Model)	10
3.2.1 Kullanım Durumu Diyagramı	11
3.2.2 Kullanım Durumu Açıklamaları	12
UC1 – Oda Ekle	12
UC2 – Oda Listele	13
UC3 – Müşteri Ekle	14
UC4 – Müşteri Listele	15

UC5 – Rezervasyon Yap .....	16
UC6 – Rezervasyon Listele .....	17
UC7 – Check-in .....	18
UC8 – Check-out .....	19
UC9 – Çıkış .....	19
3.3 Dinamik Modeller .....	20
3.3.1 Sequence Diyagramları .....	20
Oda Ekleme .....	20
Oda Listeleme .....	21
Müşteri Ekleme .....	22
Müşteri Listeleme .....	23
Rezervasyon Yapma .....	24
Rezervasyon Listeleme .....	25
Check-in .....	26
Check-out .....	27
3.4 Sınıf Modelleri .....	28
3.4.1 Room Sınıfı .....	28
3.4.2 Customer Sınıfı .....	29
3.4.3 Reservation Sınıfı .....	29
3.4.4 FileManager .....	30
3.4.5 ReservationService .....	30
3.4.6 MenuController .....	31
3.5 Nesne Modeli (Object Diagram) .....	32
4. ARAYÜZ TASARIMI (CLI) .....	33
4.1 Ana Menü Ekranı .....	33
4.2 Oda Ekleme Ekranı .....	34
4.3 Oda Listeleme Ekranı .....	35
4.4 Müşteri Ekleme Ekranı .....	36
4.5 Müşteri Listeleme Ekranı .....	37
4.6 Rezervasyon Yapma Ekranı .....	38
4.6.1 Aşama 1 – Müşteri Seçimi .....	38
4.6.2 Aşama 2 – Oda Seçimi .....	39

4.6.3 Aşama 3 – Tarih ve Ücret .....	40
4.6.4 Aşama 4 – Onay .....	40
4.7 Check-in Ekranı .....	42
4.8 Check-out Ekranı .....	43
5. SONUÇ .....	44

## 1. Giriş

Bu proje, bir otelin günlük operasyonlarını Java dili ile bilgisayar üzerinde düzenli ve güvenilir şekilde yönetmek için tasarlanmış komut satırı tabanlı bir Otel Yönetim Sistemi sunar. Resepsiyon personelinin oda durumlarını takip etmesi, müşteri kayıtlarını tutması, rezervasyon işlemlerini gerçekleştirmesi, giriş (check-in) ve çıkış (check-out) süreçlerini yönetmesi bu sistemle kolaylaşır.

Amaç, otellerde yaşanan kayıp-kayıt problemlerini, çakışan rezervasyonları, hatalı ücretlendirmeyi ;daha düzenli,takip edilebilir bir iş akışı oluşturmaktır.

### BAŞLIKLARI KALIN YAP

**1.1 Otel Yönetim Sistemi Tanımı** : Bu sistem, otel işletmelerinin oda,müşteri ve rezervasyon süreçlerini dijital ortamda yönetmesini sağlayan yazılımdır.

**1.2 Projenin Amacı** : Bu projenin amacı, manuel yürütülen otel işlemlerini hatasız ve takip edilebilir hale getirmektir.

**1.3 Projenin Kapsamı** : Proje; oda yönetimi , müşteri yönetimi, rezervasyon,check-in ve check-out işlemlerini kapsamaktadır.

**1.4 Projenin Hedefleri :**

- Rezervasyon çakışmalarını önlemek
- Kayıtların güvenli saklanması
- Kullanımı kolay bir CLI sunmak

## 2. Sistem Gereksinimleri

### 2.1 Fonksiyonel Gereksinimler

#### 2.1.1 FR1 – Oda Yönetimi

- FR1.1: Yeni oda sisteme eklenebilmelidir.
- FR1.2: Tüm odalar listelenebilmelidir.
- FR1.3: Oda durumları (Boş, Dolu, Rezerve) tutulmalıdır.
- FR1.4: Oda bilgileri (fiyat, tip, kapasite) güncellenebilmelidir.
- FR1.5: Oda ID ile oda bulunabilmelidir.

#### 2.1.2 FR2 – Müşteri Yönetimi

- FR2.1: Yeni müşteri eklenebilmelidir.
- FR2.2: Müşteri listelenebilmelidir.
- FR2.3: Müşteri ID ile arama yapılabilirdir.

#### 2.1.3 FR3 – Rezervasyon Yönetimi

- FR3.1: Müşteri adına rezervasyon oluşturulabilmelidir.
- FR3.2: Sistem uygun tarih kontrolü yapmalıdır.
- FR3.3: Sistemde çakışan rezervasyon engellenmelidir.
- FR3.4: Rezervasyon ücretini otomatik hesaplamalıdır.
- FR3.5: Rezervasyon kaydı listelenebilmelidir.

#### 2.1.4 FR4 – Check-in / Check-out Yönetimi

- FR4.1: Check-in yapılırca oda durumu “Dolu” olmalıdır.
- FR4.2: Check-out sonrası oda durumu “Boş” yapılmalıdır.
- FR4.3: Check-out sırasında ödeme doğrulanmalıdır.
- FR4.4: Giriş/çıkış tarihleri kayıt altına alınmalıdır.

#### 2.1.5 FR5 – Sistem Kontrolleri

- FR5.1: Kullanıcı tüm işlemleri menü üzerinden seçebilmelidir.
- FR5.2: Hatalı girişlerde kullanıcı bilgilendirilmelidir.
- FR5.3: Sistem güvenli şekilde sonlandırılmalıdır.

## 2.2 Fonksiyonel Olmayan Gereksinimler

- **NFR1 – Kullanılabilirlik:** Menü sade, yönlendirmeler açık olmalıdır.
- **NFR2 – Performans:** Dosya okuma/yazma işlemleri hızlı olmalıdır.
- **NFR3 – Güvenilirlik:** Rezervasyon çakışmaları engellenmelidir.
- **NFR4 – Taşınabilirlik:** JVM 17 ve üzeri yüklü her ortamda çalışabilmelidir.

## 2.3 Sistem Kısıtları

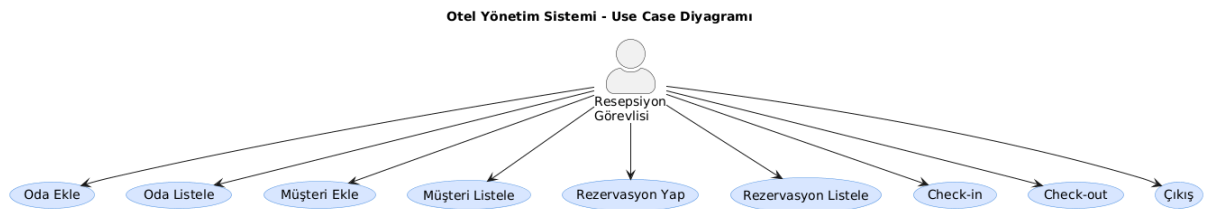
- Program tamamen **konsol tabanlıdır** (CLI).
- Tüm veriler **TXT dosyalarında** saklanacaktır.
- Sistem yalnızca **tek kullanıcı** çalışacaktır.
- Çoklu işlem veya eş zamanlı erişim desteği yoktur.
- Java 17+ gereklidir.

## 3. Sistem Modelleri

### 3.1 Use Case Modeli

Sistemdeki temel kullanım durumları:

- UC1: Oda Ekle
- UC2: Oda Listele
- UC3: Müşteri Ekle
- UC4: Müşteri Listele
- UC5: Rezervasyon Yap
- UC6: Rezervasyon Listele
- UC7: Check-in
- UC8: Check-out
- UC9: Çıkış



## Kullanım Durumu Açıklamaları

### UC1 – Oda Ekle

**Açıklama:** Sisteme yeni oda eklenmesi.

**Aktör:** Resepsiyon Görevlisi

**Ön Koşullar:** Sistem açık olmalıdır.

**Son Koşullar:** Oda rooms.txt dosyasına eklenir.

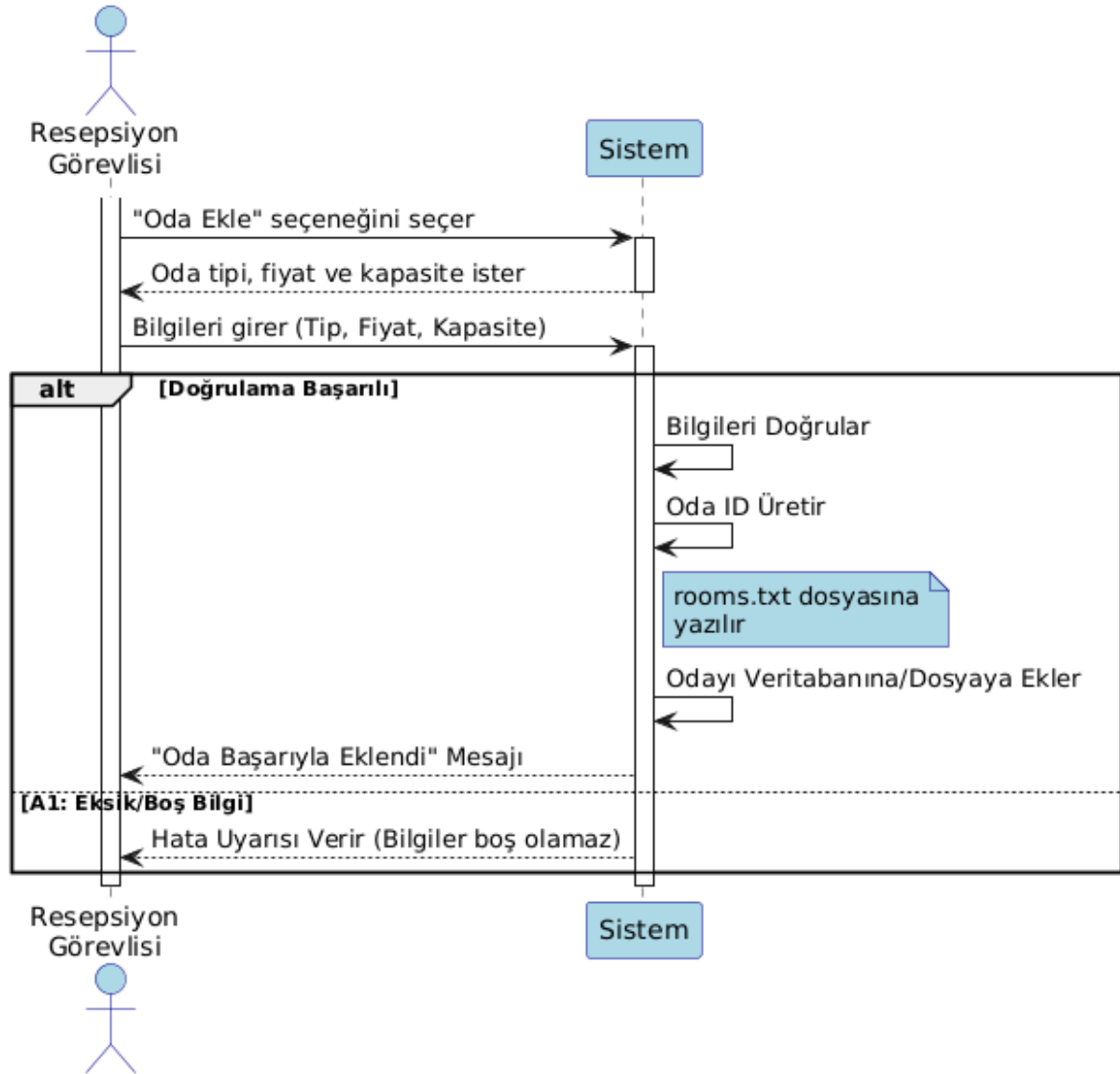
#### Ana Senaryo:

1. Kullanıcı “Oda Ekle” seçer.
2. Sistem oda tipini, fiyatı, kapasiteyi ister.
3. Kullanıcı bilgileri girer.
4. Sistem doğrular.
5. Sistem oda ID üretir.
6. Oda dosyaya eklenir.
7. Sistem başarı mesajı verir.

#### Alternatif Seneryo:

A1 – Boş bilgi girilirse uyarı verilir.

## UC1 - Oda Ekleme Senaryosu



### Açıklama:

- Resepsiyon görevlisi menüden “Oda Ekle” işlemini seçer.
- Sistem gerekli bilgileri (tip, fiyat, kapasite) ister ve girilen verileri doğrular.
- Yeni bir oda ID’si üretir.
- Oda nesnesi RAM’de oluşturulur.
- rooms.txt dosyasına yeni bir satır olarak eklenir ya da veritabanına yazılır.
- İşlem tamamlanınca görevliye “Oda başarıyla eklendi” bilgisi verilir.

### UC2 – Oda Listele

Açıklama: Sisteme kayıtlı tüm odaların görüntülenmesi.

Aktör: Resepsiyon Görevlisi

ASAĞA AL!.....OLMUYORSA SIKINTI YOK

Ön Koşullar: Sistem açık olmalıdır; rooms.txt dosyası mevcut olmalıdır.

Son Koşullar: Odaların listesi ekrana yazdırılır.

### Ana Senaryo:

Kullanıcı “Oda Listele” seçeneğini seçer.

Sistem rooms.txt dosyasını açar.

Sistem dosyadaki tüm oda kayıtlarını okur.

Sistem odaları ekranda listeler.

Sistem listeleme işleminin tamamlandığını bildirir.

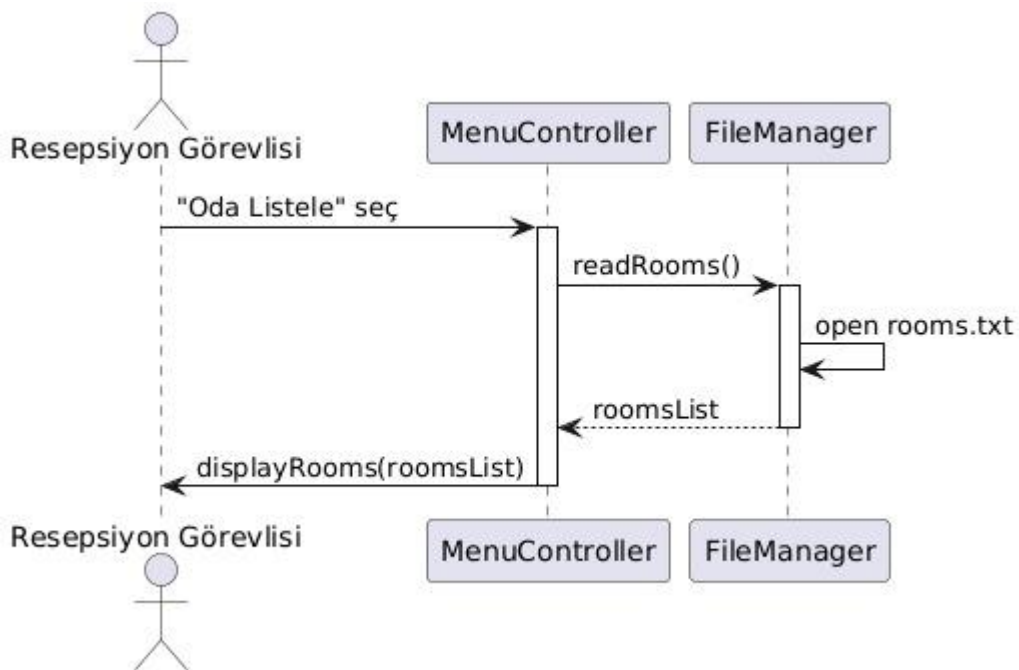
### Alternatif Senaryolar:

A1 – Oda kaydı bulunamaz:

rooms.txt dosyası boşsa veya hiç oda eklenmemişse, sistem “Listelenecek oda bulunamadı” uyarısı verir.

A2 – Dosya bulunamazsa:

rooms.txt dosyası bulunamazsa sistem “Oda dosyası bulunamadı” hatası gösterir.





**Açıklama:**

- Resepsiyon görevlisi menüden “Oda Listele” işlemini seçer.
- MenuController, oda kayıtlarını almak için FileManager’a istekte bulunur.
- FileManager rooms.txt dosyasını açar ve tüm oda bilgilerini okur.
- Okunan veriler bir liste olarak MenuController’a geri gönderilir.
- MenuController, bu listeyi ekranda göstermek üzere kullanıcıya iletir.
- Resepsiyon görevlisi tüm oda bilgilerini görüntüler ve işlem tamamlanır.

**UC3 – Müşteri Ekle**

Açıklama: Sisteme yeni bir müşterinin kaydedilmesi.

Aktör: Resepsiyon Görevlisi

Ön Koşullar: Sistem açık olmalıdır; customers.txt dosyası mevcut olmalıdır.

Son Koşullar: Müşteri bilgisi customers.txt dosyasına eklenir.

**Ana Senaryo:**

Kullanıcı “Müşteri Ekle” seçeneğini seçer.

Sistem kullanıcıdan müşterinin temel bilgilerini ister (ad, soyad, telefon).

Kullanıcı bilgileri girer.

Sistem bilgilerin doğruluğunu ve boş alan olup olmadığını kontrol eder.

Sistem müşteri için benzersiz bir müşteri ID’si üretir.

Müşteri bilgisi customers.txt dosyasına eklenir.

Sistem “Müşteri başarıyla eklendi” mesajı verir.

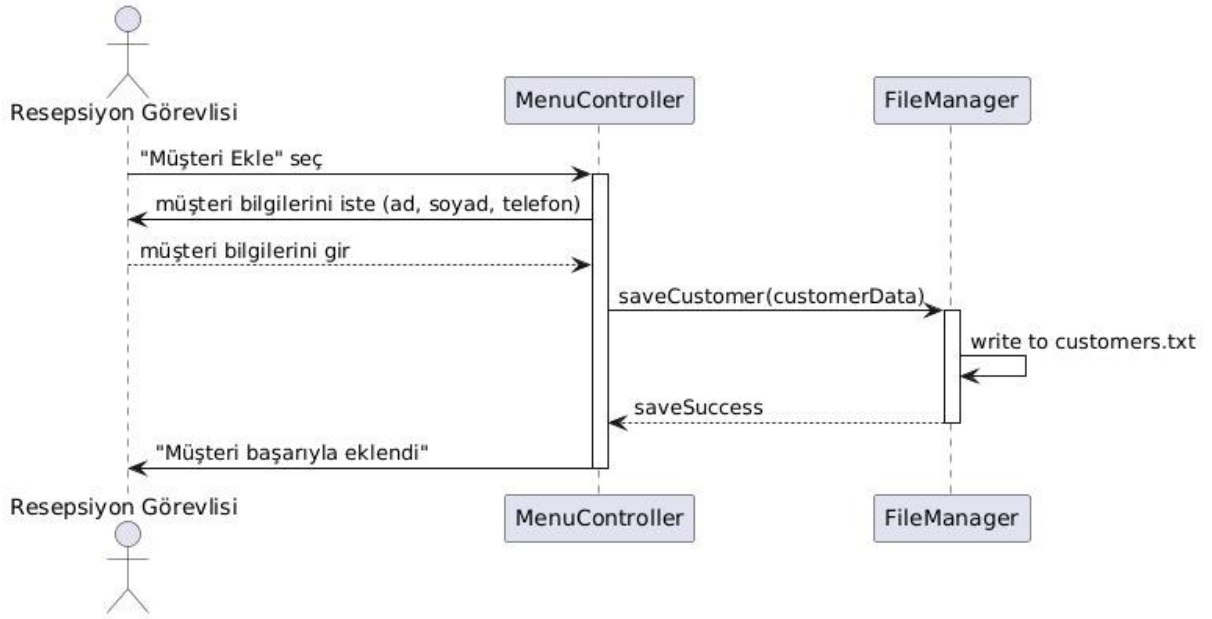
**Alternatif Senaryolar:**

A1 – Boş bilgi girilirse:

Sistem “Bilgiler eksik, lütfen tekrar deneyin” uyarısı verir ve doğru giriş yapılana kadar işlem tekrarlanır.

A2 – customers.txt dosyası oluşturulamıyorsa:

Sistem “Müşteri dosyasına erişilemedi” şeklinde hata mesajı verir.



## UC4 – Müşteri Listele (List Customer)

Açıklama: Sistemde kayıtlı tüm müşterilerin listelenmesi.

Aktör: Resepsiyon Görevlisi

Ön Koşullar: Sistem açık olmalıdır, customers.txt dosyası mevcut olmalıdır.

Son Koşullar: Mevcut müşteriler ekrana yazdırılır.

### Ana Senaryo:

Kullanıcı “Müşteri Listele” seçeneğini seçer.

Sistem customers.txt dosyasını okur.

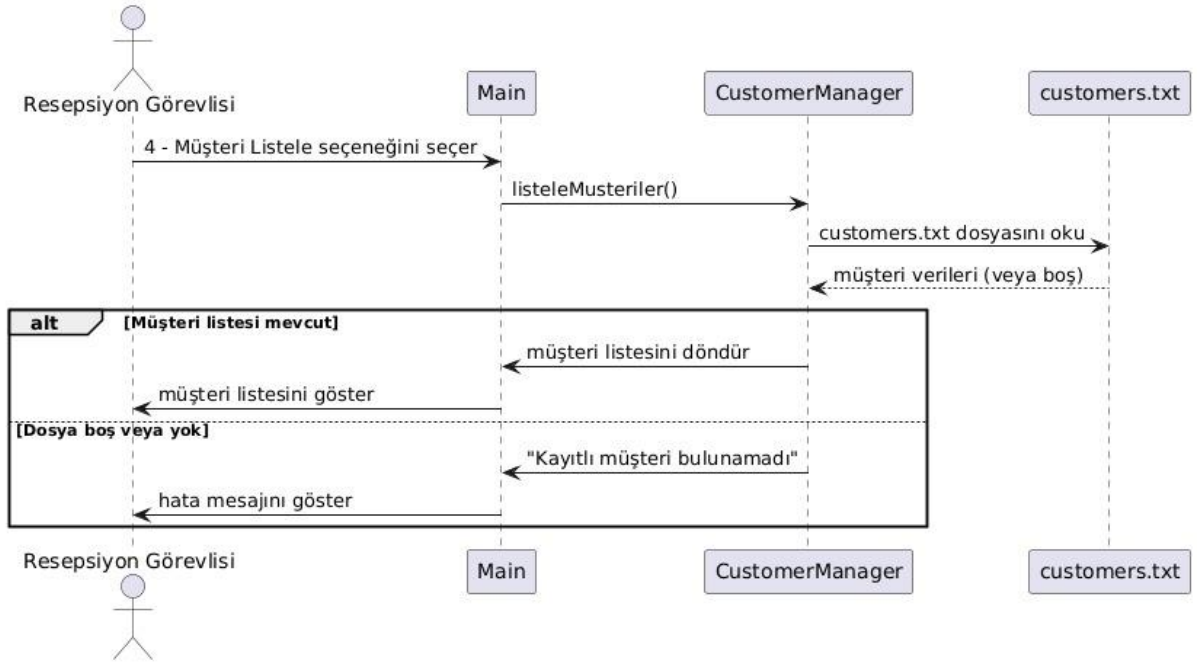
Sistem dosyadaki müşteri bilgilerini çözümler.

Sistem müşteri listesini kullanıcıya gösterir.

### Alternatif Senaryolar:

A1 – Dosya boş veya bulunamadı:

Sistem “Kayıtlı müşteri bulunamadı.” mesajı gösterir.



### Açıklama:

- Resepsiyon görevlisi “Müşteri Listele”yi seçer.
- Sistem customers.txt dosyasını okuyup müşteri verilerini çeker.
- Dosyada kayıt varsa liste ekranda gösterilir.
- Dosya boşsa “Kayıtlı müşteri bulunamadı” uyarısı verilir.

## UC5 – Rezervasyon Yap (Make a Reservation)

Açıklama: Sisteme yeni bir rezervasyon eklenmesi.

Aktör: Resepsiyon Görevlisi

### Ön Koşullar:

Sistem açık olmalıdır.

rooms.txt ve customers.txt dosyaları mevcut olmalıdır.

Müşteri ve oda bilgileri sistemde kayıtlı olmalıdır.

### Son Koşullar:

Rezervasyon reservations.txt dosyasına kaydedilir.

### Ana Senaryo:

Kullanıcı “Rezervasyon Yap” seçeneğini seçer.

Sistem müşteri ID’sini ister.

Kullanıcı müşteri ID’sini girer.

Sistem müşteri varlığını doğrular.

Sistem oda ID’sini ister.

Kullanıcı oda ID’sini girer.

Sistem oda uygunluğunu kontrol eder.

Sistem giriş ve çıkış tarihlerini ister.

Sistem tarihleri doğrular.

Sistem rezervasyon ID’si üretir.

Sistem bilgileri reservations.txt dosyasına ekler.

Sistem başarı mesajı gösterir.

## Alternatif Akışlar

renkli basliklari buyut ve kalin yap diger UC lerde ayni olsun

### A1 – Müşteri bulunamadı:

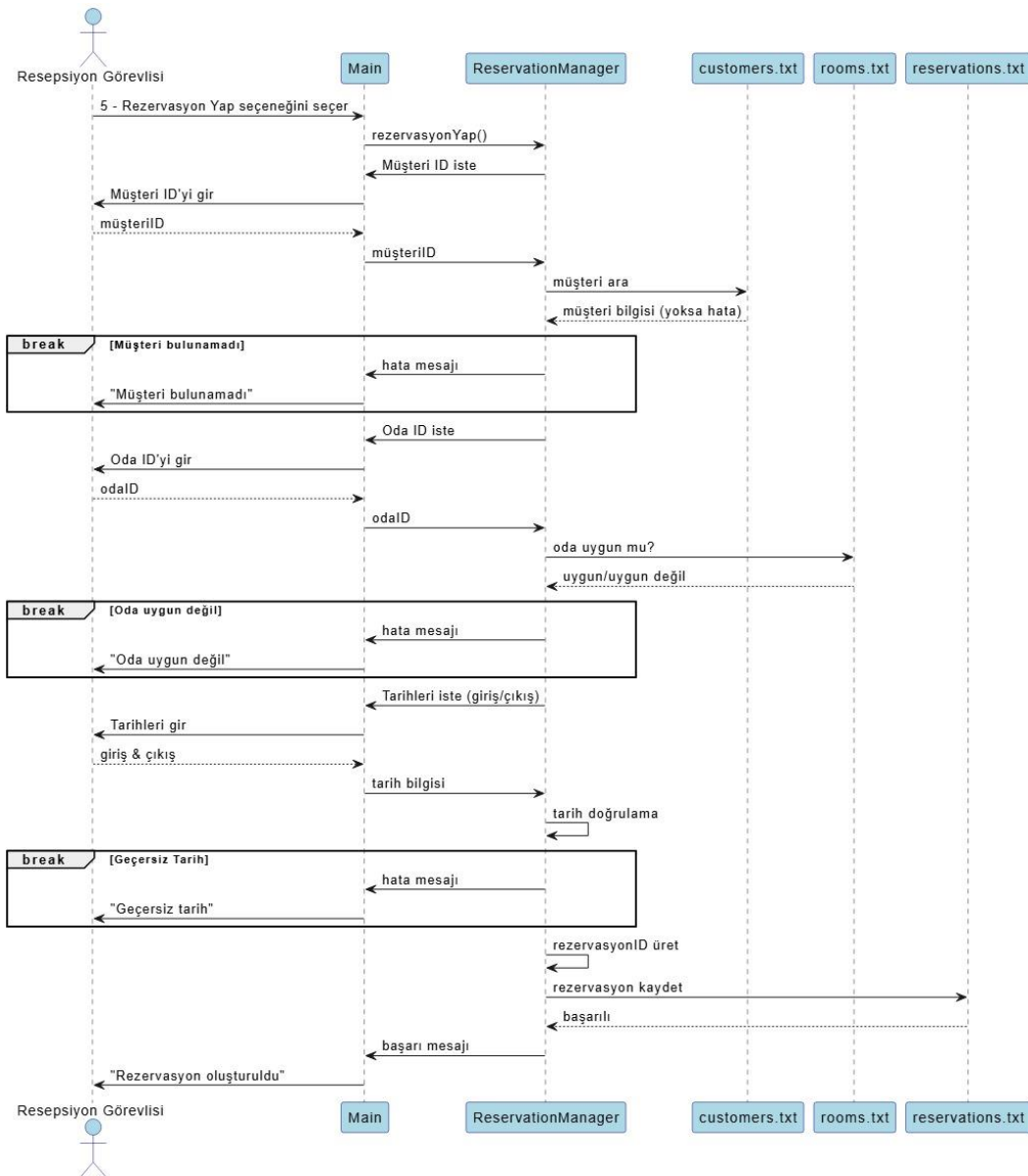
Sistem hata verir → işlem iptal edilir.

### A2 – Oda mevcut değil / uygun değil:

Sistem uyarı verir → işlem iptal edilir.

### A3 – Tarihler geçersiz:

Sistem uyarı verir → yeniden tarih girişi istenir.



başlık olarak açıklama yazz!!!!

bununla degistir --> ekranda müşteri bulunamadi yazar.

- **Müşteri Kontrolü:** Resepsiyon görevlisi müşteri ID'si girer. Sistem, müşteri ID'sini kontrol eder. (Yoksa **hata verir.**)
- **Oda Kontrolü:** Oda ID'si girilir ve sistem odanın uygunluğunu kontrol eder. (Uygun değilse hata verir.) **Hata: Oda bulunamadı.**
- **Tarih Kontrolü:** Giriş/Çıkış tarihleri girilir ve sistem tarihlerin geçerliliğini doğrular. (Geçersizse hata verir.) **Hata: Tarihler geçersiz. Giriş tarihi bugünden önce olamaz ve çıkış tarihinden önce olmalıdır.**
- **Kayıt:** Tüm kontroller başarılıysa, sistem bir rezervasyon ID'si üretir ve rezervasyonu kalıcı olarak dosyaya kaydeder (reservations.txt).
- **Sonuç:** Görevliye "**Rezervasyon oluşturuldu**" mesajı gösterilir.

## UC:6 – Rezervasyon Listele (List Reservation)

Açıklama: Sistemde kayıtlı tüm rezervasyonların listelenmesi.

Aktör: Resepsiyon Görevlisi

### Ön Koşullar:

Sistem açık olmalıdır.

reservations.txt dosyası bulunmalıdır.

### Son Koşullar:

Rezervasyon bilgileri ekrana yazdırılır.

### Ana Senaryo:

Kullanıcı "Rezervasyon Listele" seçeneğini seçer.

Sistem reservations.txt dosyasını okur.

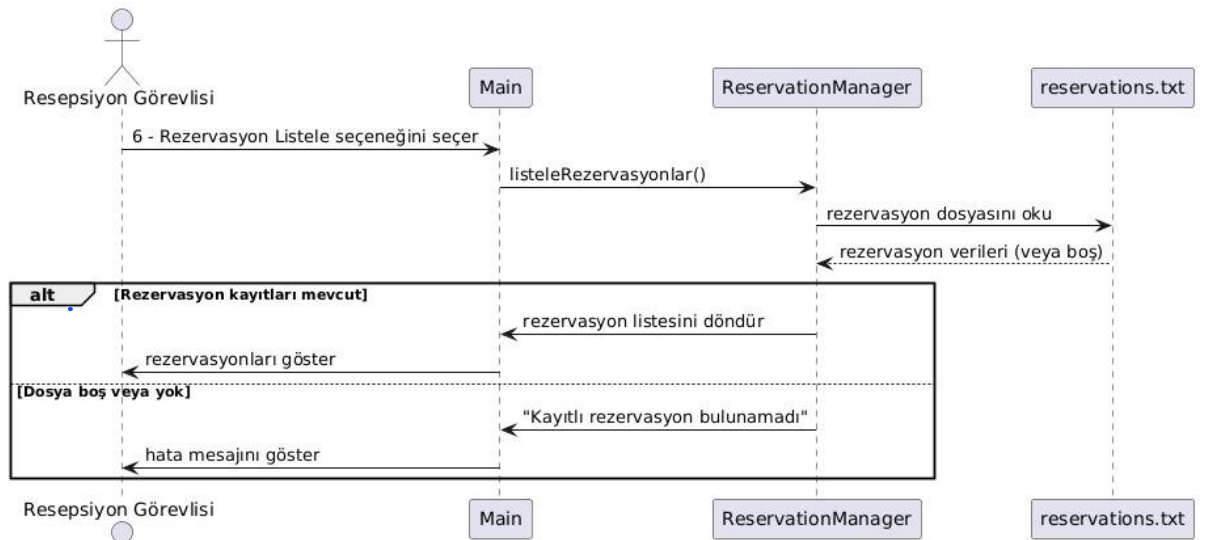
Sistem dosyadaki rezervasyon kayıtlarını çözümler.

Sistem tüm rezervasyonları ekrana listeler.

### Alternatif Senaryo:

A1 – Dosya boş ya da hiç kayıt yok:

Sistem: "Kayıtlı rezervasyon bulunamadı" mesajı verir.



Acıklama ekle buraya.....kalin olarak

- **İstek:** Görevli, listeleme işlemini başlatır.
- **Veri Çekme:** Sistem, kayıtlı rezervasyonların tutulduğu dosyayı (reservations.txt) okur.
- **Kontrol:** burasi baslik olacakti.....
- **Kayıt Varsa:** Listeyi görevliye gösterir.
- **Kayıt Yoksa:** "Kayıtlı rezervasyon bulunamadı" mesajını gösterir.

## UC7 – Check-in (GIRIS)

Açıklama: Otele gelen müşterinin rezervasyonunun doğrulanması ve oda giriş işleminin yapılması.

Aktör: Resepsiyon Görevlisi

### Ön Koşullar:

Sistem açık olmalıdır.

Müşterinin sistemde kayıtlı bir rezervasyonu olmalıdır.

Oda dolu olmamalıdır.

### Son Koşullar:

Rezervasyon “aktif” duruma güncellenir veya check-in kaydı oluşturulur.

Oda durumu “DOLU” olarak işaretlenir.

**Ana Senaryo** başlık....ve kalin yap ,altta

Kullanıcı “Check-in” seçeneğini seçer.

Sistem müşteri ID’si ister.

Kullanıcı müşteri ID’sini girer.

Sistem müşterinin rezervasyonunu arar.

Sistem rezervasyonun tarihlerini doğrular (bugün giriş günü mü?).

Sistem oda durumunu kontrol eder.

Sistem check-in işlemini tamamlar.

Sistem başarı mesajı gösterir.

#### Alternatif Senaryolar

**A1 – Müşteri rezervasyonu yok:**

→ Sistem “Bu müşteriye ait aktif rezervasyon bulunamadı” yazar, işlem durur.

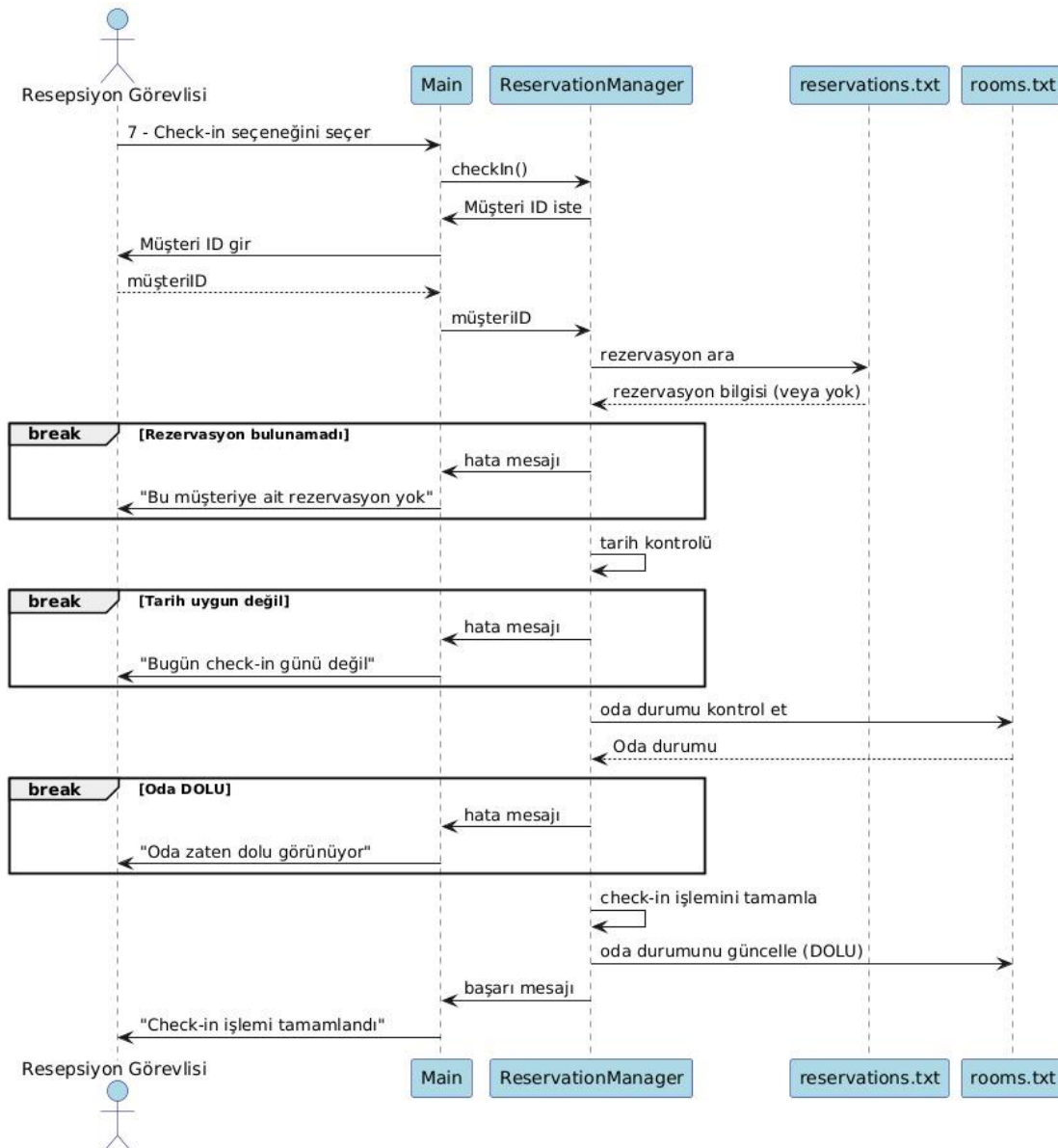
**A2 – Oda zaten dolu:**

→ Sistem “Oda dolu, check-in yapılamaz” mesajı verir.

**A3 – Giriş tarihi uygun değil:**

→ Sistem “Check-in tarihi bugün değil” mesajı verir.





Açıklama yaz başlık olarak

- **İstek:** Görevli, Müşteri ID'si girerek Check-in işlemini başlatır.
- **Kontroller:** Sistem sırasıyla şunları kontrol eder: acık mavi satiri altta taşı.....Kontroller kalın ve başlık olacak şekilde
- Müşteriye ait **rezervasyon var mı?** (Yoksa hata verir.)
- **Tarih** bugün check-in gününe **uygun mu?** (Değilse hata verir.)
- **Oda** zaten **dolu mu?** (Doluysa hata verir.)
- **Başarı:** Tüm kontrollerden geçilirse, sistem odanın durumunu **DOLU** olarak günceller.
- **Sonuç:** Görevliye "**Check-in işlemi tamamlandı**" mesajı gösterilir.

## UC8 – Check-out (ÇIKIS)

Açıklama: Otelden ayrılan müşterinin konaklama bitiş işlemlerinin yapılması.

Aktör: Resepsiyon Görevlisi

### Ön Koşullar:

Sistem açık olmalıdır.

Müşterinin aktif bir rezervasyonu / check-in kaydı olmalıdır.

Oda durumu “DOLU” olarak işaretlidir.

### Son Koşullar:

Rezervasyon durumu “tamamlandı” veya “kapandı” olarak güncellenir.

Oda durumu “BOŞ” olarak ayarlanır.

Ana Senaryo

Kullanıcı “Check-out” seçeneğini seçer.

Sistem müşteri ID’sini ister.

Kullanıcı müşteri ID’sini girer.

Sistem müşteriye ait aktif rezervasyonu arar.

Sistem rezervasyonun check-in yapılmış olduğunu doğrular.

Sistem oda bilgisini alır.

Sistem oda durumunu “BOŞ” olarak günceller.

şurayı boşluk sil.....yanlış anlaşılma olur

Sistem rezervasyon durumunu “tamamlandı” yapar.

Sistem başarı mesajı gösterir.

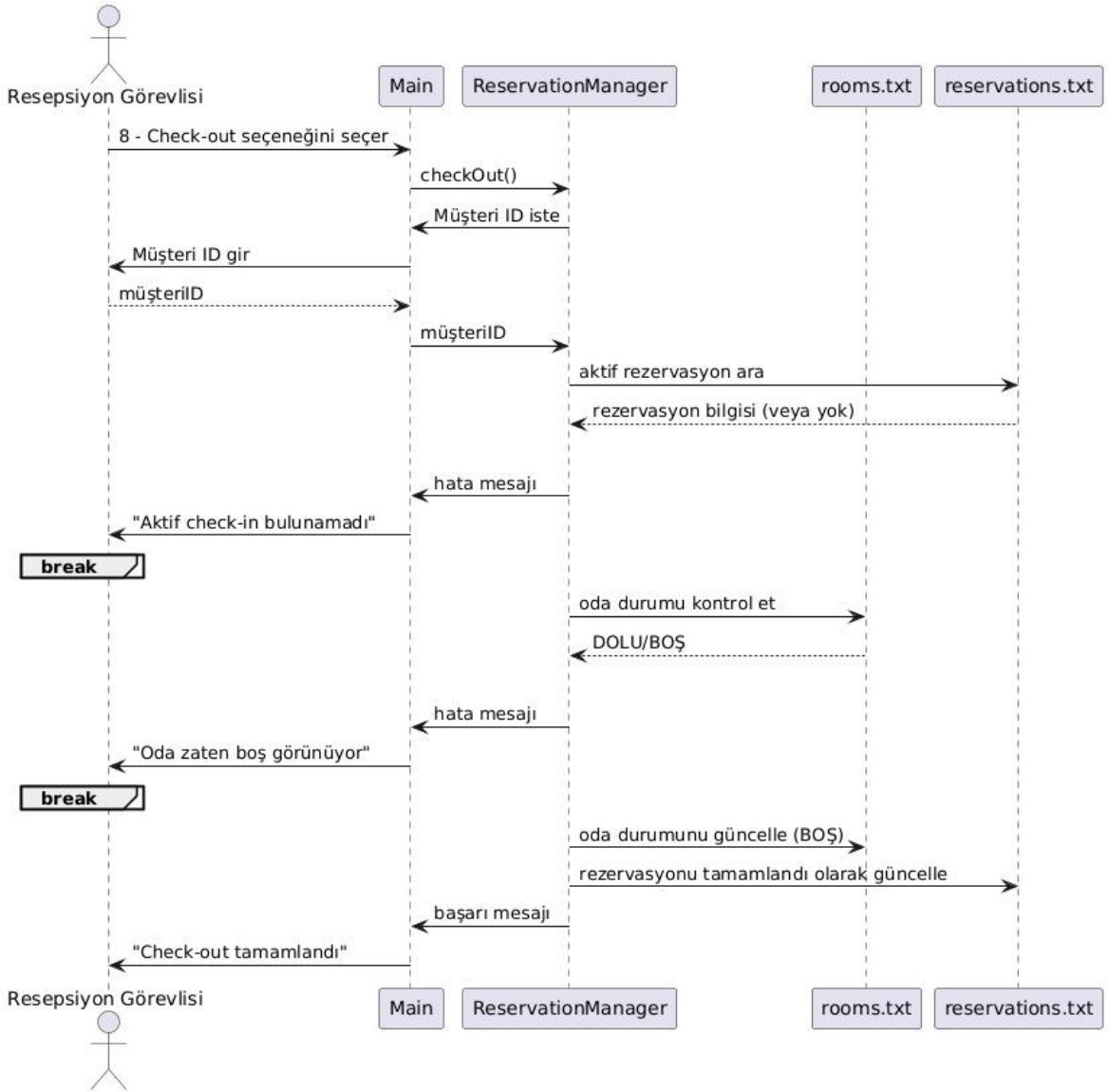
**Alternatif Senaryolar** **kalinlastir**

A1 – Müşterinin aktif rezervasyonu yok:

→ Sistem “Bu müşteriye ait aktif check-in bulunamadı.” mesajı verir.

A2 – Oda zaten boş görünüyorsa (veri hatası):

→ Sistem uyarı verir, işlem iptal edilir.



başlık - Açıklama olacaktı burada, istek -> k:

İstek: Görevli, Müşteri ID'si girerek Check-out işlemini başlatır.

Kontrol 1 (Aktif Rezervasyon): Sistem, müşteriye ait aktif bir check-in (yani tamamlanmamış bir rezervasyon) arar.

Hata: Bu rezervasyon için henüz check-in yapılmamış veya işlem zaten bitmiş

Yoksa: Hata verir ("Aktif check-in bulunamadı").

mavi isareti yeri ,  
yani aradaki  
Kontroller yaz ....kali  
n yap

Kontrol 2 (Oda Durumu): Sistem, odanın mevcut durumunu kontrol eder.

Boş Görünüyorsa: Hata verir ("Oda zaten boş görünüyor").

İşlem: Kontrollerden geçilirse, sistem iki güncelleme yapar:

bununla değiştir...  
Hata: Oda verisine ulaşılamadı

üstteki işlem adında başlık yap .....altakileri içerik olarak yap yani

Odanın durumunu BOŞ olarak günceller (rooms.txt).

Rezervasyonu tamamlandı olarak günceller (reservations.txt).

**Sonuç:** Görevliye "Check-out tamamlandı" mesajı gösterilir.

## UC9 – Çıkış (Exit)

Açıklama: Kullanıcının otel yönetim sisteminden güvenli şekilde çıkış yapması.

Aktör: Resepsiyon Görevlisi

### Ön Koşullar:

Sistem çalışır durumda olmalıdır.

### Son Koşullar:

Program sonlanır.

Kullanıcı ana menüye dönemez, uygulama tamamen kapanır.

### Ana Senaryo:

Kullanıcı menüden “Çıkış” seçeneğini seçer.

Sistem çıkış işleminin onayını alabilir (opsiyonel).

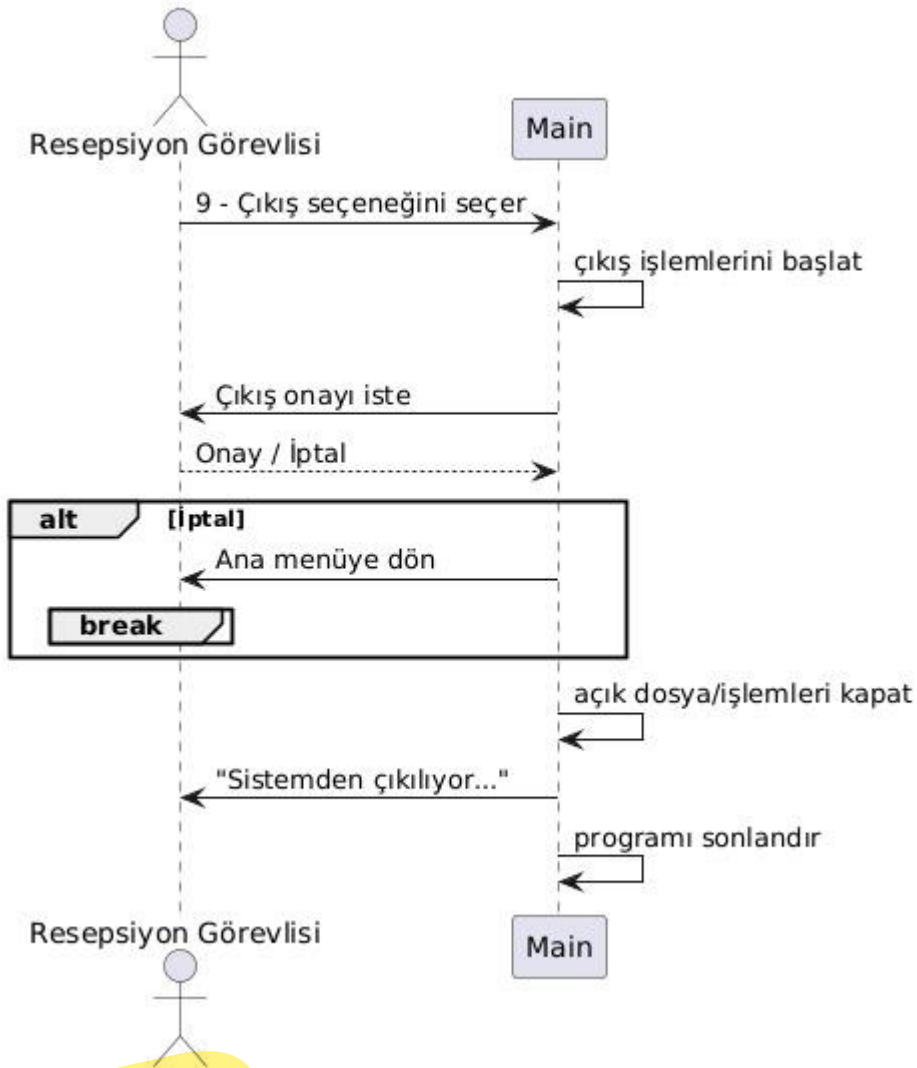
Sistem açık dosya işlemlerini (varsa) kapatır.

Sistem programı güvenli şekilde sonlandırır.

### Alternatif Senaryolar:

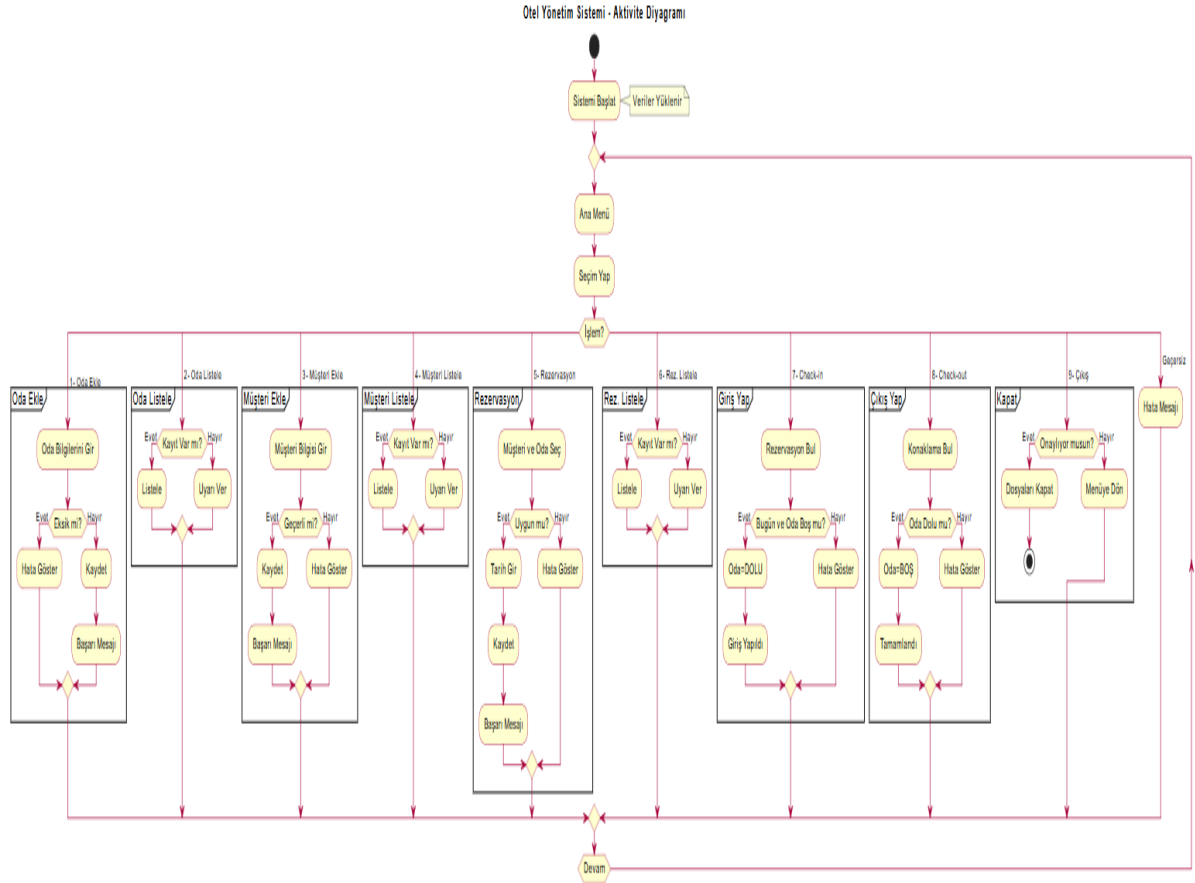
A1 – Kullanıcı onay vermez (opsiyonel tasarım):

→ Sistem ana menüye geri döner.



acıklama yaz burada

- **İstek:** Resepsiyon Görevlisi, sistemde "9 - Çıkış" seçeneğini seçer.
- **Onay:** Sistem, görevliye **Çıkış onayı** sorar.
- **Alternatif (İptal):**
- Eğer görevli çıkışı **iptal** ederse, sistem ana menüye geri döner (akış sona erer).
- **Çıkış:**
- Eğer görevli çıkışı **onaylarsa**, sistem tüm **açık dosya/işlemleri kapatır**.
- Görevliye "**Sistemden çıkılıyor...**" mesajını gösterir.
- Son olarak, **programı sonlandırır**.



## Otel Yönetim Sistemi - Aktivite Akış Adımları

1. **Sistemi Başlat:** Program açılır ve veri saklama dosyaları (rooms.txt, customers.txt vb.) sisteme yüklenir.
2. **Ana Menü Döngüsü:** Kullanıcıya yapabileceği işlemlerin listesi sunulur ve bir seçim yapması beklenir. İşlem bitince sistem tekrar buraya döner.
3. **A.Oda Ekle:** Kullanıcıdan oda tipi, fiyat ve kapasite bilgileri istenir. Bilgiler eksiksiz girilirse oda kaydedilir, eksikse hata mesajı verilir.
4. **B.Oda Listele:** Oda dosyasının dolu olup olmadığı kontrol edilir. Kayıt varsa tüm odalar listelenir, yoksa "Kayıt yok" uyarısı verilir.
5. **C.Müşteri Ekle:** Müşterinin ad, soyad ve telefon bilgileri alınır. Veriler geçerliyse yeni müşteri ID'si üretilip kaydedilir.
6. **D.Müşteri Listele:** Sistemde kayıtlı müşteri olup olmadığına bakılır. Varsa liste ekrana dökülür, yoksa uyarı verilir.

numara olanı harf / roman ile değiştiriver .....koddan oda ekle 1'den sona gidiyor harfleri o sıradan kaldırı!!!!!!

### 7. E.Rezervasyon Yap:

- Müşteri Kontrolü:* Girilen ID'ye sahip müşteri var mı?
- Oda Kontrolü:* Seçilen oda o tarihlerde uygun mu?
- Tarih Kontrolü:* Tarihler geçerli mi?
- Sonuç:* Tüm şartlar sağlanırsa rezervasyon dosyaya işlenir.

### 8. F. Rezervasyon Listele: Mevcut rezervasyon kayıtları taranır. Dosya doluysa liste gösterilir, boşsa kullanıcı bilgilendirilir.

### 9. 7. C Check-in (Giriş İşlemi): Müşterinin rezervasyonu bulunur. Giriş tarihinin *bugün* olup olmadığı ve odanın *boş* olduğu doğrulanır. Onaylanırsa oda durumu "DOLU" yapılır.

### 10. 8. Check-out (Çıkış İşlemi): Müşterinin halihazırda otelde konaklayıp konaklamadığı (aktif check-in) kontrol edilir. Varsa oda durumu "BOŞ" yapılır ve konaklama tamamlanır.

[burayı 9 yap cikis önünde](#)

**Çıkış (Programı Kapat):** Kullanıcıdan çıkış onayı istenir. "Evet" denirse açık dosyalar kapatılır ve program sonlandırılır. "Hayır" denirse ana menüye dönlür.

**Hatalı Seçim:** Menüde olmayan bir rakam girilirse (örn: 15), sistem hata mesajı gösterip tekrar menüyü açar.

## 3.3 Sınıf Modelleri

Bu bölümde sistemde yer alan sınıflar, sorumlulukları, ilişkileri ve aralarındaki işlevsel bağımlılıklar açıklanmaktadır.

### 1. Room Sınıfı Sorumluluklar

Odaya ait temel bilgileri (oda numarası, tip, kapasite, fiyat) saklar.

Odanın durumunu yönetir (Boş, Dolu, Rezerve).

Room verisini TXT formatında yazılabilir hale getirir.

Öznitelikler (Varsayılan)

roomId : int

type : String

yerine ATTRIBUTES YAZ (OZNITELIKLER).....BU SEKILDE GUZEL OLUR

capacity : int

price : double

status : String

Davranışlar

kalin yap

toFileString()

setStatus()

İlişkiler

1 Room → N Reservation Bir oda birden fazla rezervasyon ile ilişkilendirilebilir.

## 2. Customer Sınıfı Sorumluluklar

Müşteriye ait temel bilgileri (ad, soyad, telefon) saklar.

TXT dosya formatı üretir.

Öznitelikler

customerId : int

name : String

surname : String

phone : String

Davranışlar

toFileString()

İlişkiler

1 Customer → N Reservation Bir müşteri birden fazla rezervasyon yapabilir.

## 3. Reservation Sınıfı Sorumluluklar

Rezervasyon bilgilerini (giriş-çıkış tarihi, oda, müşteri) tutar.

Ücret hesaplama işlemlerine destek sağlar.

Check-in / Check-out süreçleri için gerekli bilgileri içerir.

odaid  
tip  
kapasite  
fiyat  
durum

olacak şekilde güncelle Ömer

Attributes(oznitelikler)  
musteriid  
ad  
soyad  
telefon



### Öznitelikler

reservationId : int

customerId : int

roomId : int

checkInDate : LocalDate

checkOutDate : LocalDate

totalPrice : double

status : String (Rezerve / Check-in / Check-out)

Attributes (oznitelikler)  
reservasyonId  
musteriId  
roomId  
girisTarihi  
cikisTarihi  
toplamUcret  
durum

### Davranışlar

calculatePrice()

toFileString()

### İlişkiler

Reservation → 1 Customer

Reservation → 1 Room

## 4. FileManager Sınıfı Sorumluluklar

Sistem verilerini TXT dosyalarına okur ve yazar.

Oda, müşteri ve rezervasyon kayıtlarını yönetir.

Unique ID üretir (roomId, customerId, reservationId).

Tüm sınıflarla veri bağı olan tek katmandır.

### Öznitelikler

roomsFile : String

customersFile : String

reservationsFile : String

Attributes  
odaDosyasi  
musteriDosyasi  
reservasyonDosyasi

### Davranışlar

readRooms()

writeRoom()

readCustomers()

writeCustomer()

readReservations()

writeReservation()

generateId()

### İlişkiler

FileManager → Room

FileManager → Customer

FileManager → Reservation

(Dosya yönetimi sebebiyle tüm sınıfların veri erişim noktasıdır.)

## 5. ReservationService Sınıfı Sorumluluklar

İş kurallarını yönetir (tarih çakışması, uygun oda kontrolü).

Check-in ve check-out işlemlerini yönetir.

Ücret hesaplama işlemlerini yürütür.

FileManager ile koordineli çalışarak rezervasyon yönetimini sağlar.

### Öznitelikler

fileManager : FileManager

### Davranışlar

createReservation()

isDateAvailable()

checkIn()

checkOut()

calculatePrice()

İlişkiler

ReservationService → FileManager

ReservationService → Room

ReservationService → Reservation

ReservationService → Customer

## 6. MenuController Sınıfı Sorumluluklar

Kullanıcı (resepsiyon görevlisi) ile sistem arasındaki tüm etkileşimi yönetir.

Menü seçimlerini ilgili servislere yönlendirir.

Kullanıcıdan veri alır, doğrular, iş akışını yönetir.

Öznitelikler | kalın yapalım,

reservationService : ReservationService

fileManager : FileManager

Davranışlar

showMenu() menuGoste

handleRoomOperations() odalislemleri()

handleCustomerOperations() musterislemleri()

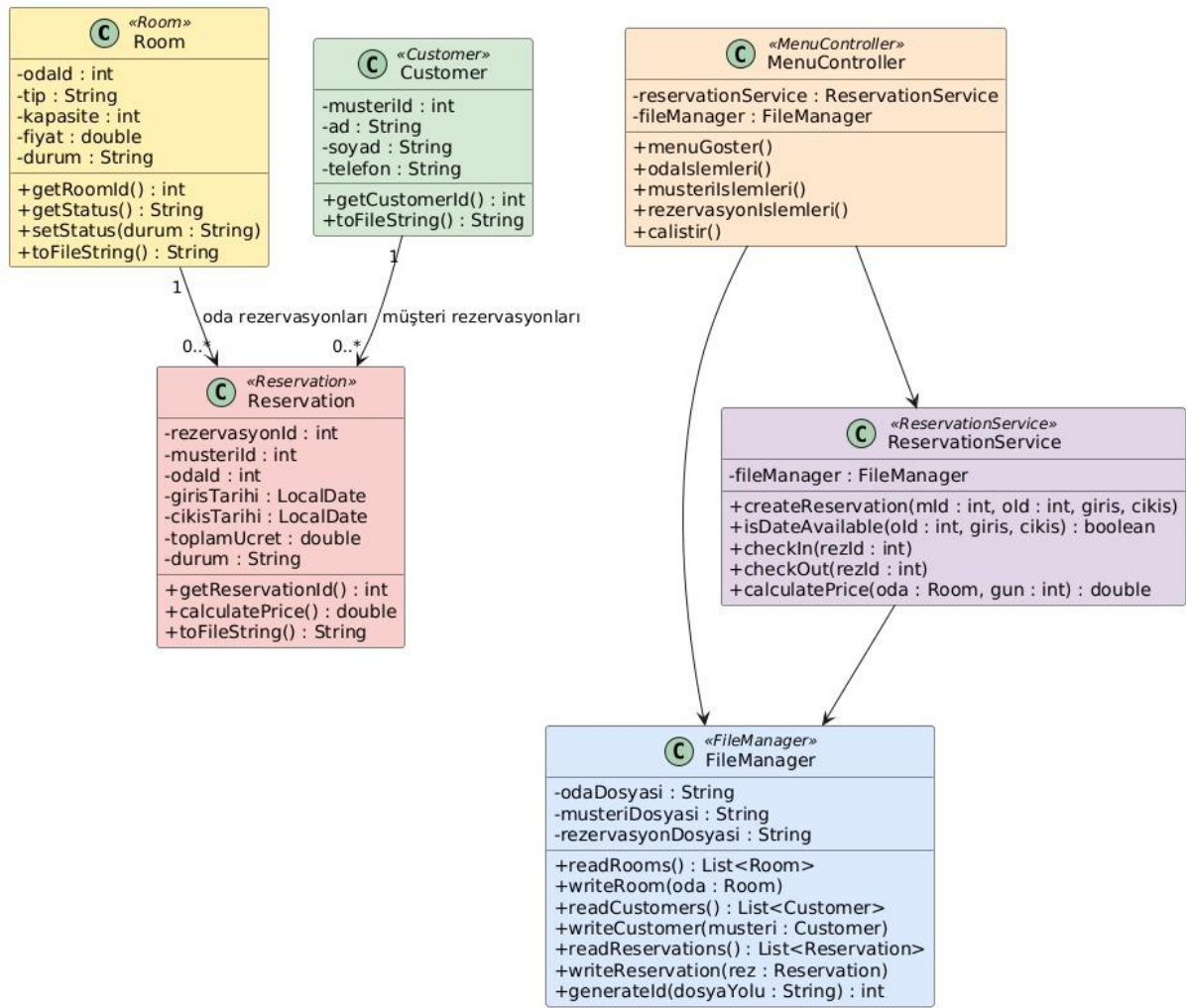
handleReservationOperations() rezervasyonislemleri()

run() calistir()

İlişkiler

MenuController → ReservationService

MenuController → FileManager



**... Room Sınıfı (Oda) Açıklama** Sistemde konaklama hizmeti sunulan fiziksel odayı temsil eder (tek kişilik, çift kişilik, süit vb.). **Sorumluluklar**

- Oda bilgilerini (tip, fiyat, kapasite) tutmak
- Oda durumunu (Boş/Dolu) yönetmek
- Dosya yazımı için satır formatı üretmek

### İlişkiler

- 1 Room → Çok sayıda Reservation (Bir oda farklı zamanlarda birçok kez rezerve edilebilir.)

**... Customer Sınıfı (Müşteri) Açıklama** Sistemde kayıtlı müşterinin tüm kimlik ve iletişim bilgilerini içerir. **Sorumluluklar**

- Müşteri bilgilerini saklama
- Dosya formatında kendini yazılabilir hâle getirme

### İlişkiler

- 1 Customer → Çok sayıda Reservation

... **Reservation Sınıfı (Rezervasyon) Açıklama** Her rezervasyon işlemi için bir tane Reservation nesnesi oluşturulur. **Sorumluluklar**

- Rezervasyonun durumunu takip etmek
- Giriş ve çıkış tarihlerini tutmak
- Ücret hesaplamak
- Dosyaya yazılabilir hâle gelmek

önünde numalandır, açıklama ve sorumlulukları altta al kalın kalsin!!!

### İlişkiler

- Bir Reservation; bir Customer ve bir Room ile bağlantılıdır.
- Bağımlılık yönü: Reservation → Customer / Room (ID üzerinden)

**1. FileManager (Dosya Yönetimi) Açıklama** Sistemin veri katmanıdır. Tüm bilgiler TXT dosyalarında saklandığı için FileManager, veri bütünlüğünün merkezidir. **Sorumluluklar**

- Dosya okuma
- Dosyaya ekleme
- Dosya yeniden yazma
- ID üretme

### İlişkiler

- Tüm sınıflar FileManager kullanır.

**2. ReservationService (İş Kuralları) Açıklama** Tüm iş kurallarını yöneten katmandır. **Sorumluluklar**

- Rezervasyon işlemindeki tüm doğrulamaları (tarih, oda uygunluğu) yapmak
- Oda durumunu güncelleme (Dolu/Boş)
- Rezervasyon kaydı oluşturma
- Dosyaları senkron şekilde güncelleme

### İlişkiler

- ReservationService → FileManager
- ReservationService → Customer / Room / Reservation

... **MenuController (Kullanıcı Arayüzü Kontrolcüsü) Açıklama** Kullanıcı (Resepsiyon Görevlisi) ile sistem arasındaki tüm etkileşimin merkezidir. **Sorumluluklar**

- Menü ekranda gösterilir

- Kullanıcıdan veri alınır
- Doğru servislere yönlendirme yapılır

## İlişkiler

- MenuController → ReservationService
- MenuController → FileManager

şunu kaldır bir zahmet

5. Arayüz Tasarımı (CLI) kısmını genişleterek profesyonel bir üniversite raporu formatında *tam bölüm* hâline getiriyorum. Bu bölümde:

- ✓ Her ekran bir “CLI Wireframe” formatında
- ✓ Açıklamalı
- ✓ Kullanıcı adımları yazılmış
- ✓ Hatalı giriş senaryoları eklenmiş
- ✓ Gereksinimlere uygun şekilde tamamlanmış

## 3.4 NESNE MODELİ (Object Diagram)

Bu senaryoda otelde şu anda RAM’de bulunan nesnelerin durumu şöyledir:

- ✓ Odalar (Toplam 4 adet)

Oda #101 → Boş

Oda #102 → Dolu (check-in yapılmış)

Oda #103 → Rezerve (müşteri gelmemiş ama rezervasyon aktif)

Oda #104 → Serbest (Daha önce check-out yapılmış)

- ✓ Müşteriler (Toplam 3 adet)

Ahmet Yılmaz

Ayşe Demir

Murat Kaya

- ✓ Aktif Rezervasyonlar (Toplam 2 adet)

Rezervasyon #5001 → Check-in yapılmış

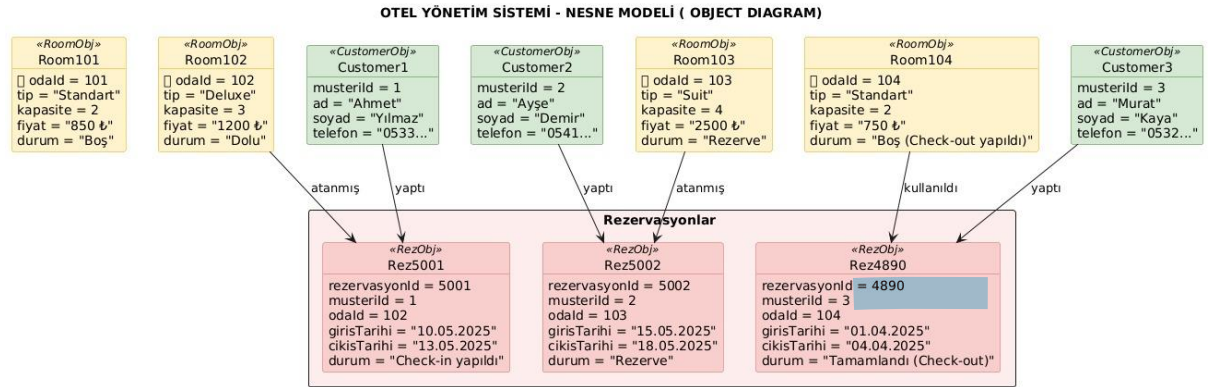
Rezervasyon #5002 → Yarın için rezerve, henüz check-in yok

- ✓ Geçmiş Bir Rezervasyon (1 adet)

Rezervasyon #4890 → Müşteri çıkış yapmış (tamamlanmış)

4890 yerine 5003 yap.....

Bu tabloyu sisteme uygun Object Diagram (Nesne Diyagramı) olarak çiziyoruz



Nesne Tipi	Kaç Adet?	Açıklama
Room (Oda)	4	Boş, dolu, rezerve ve check-out yapılmış
Customer (Müşteri)	3	RAM’de kayıtlı müşteri nesneleri
Reservation (Rezervasyon)	3	2 aktif + 1 tamamlanmış
İlişkiler	Var	Her rezervasyon → bir müşteri + bir oda

## 4. Arayüz Tasarımı (CLI)

Bu bölümde sistemin komut satırı (CLI) üzerinden çalışan arayüzleri açıklanmaktadır. Sistem grafik arayüz (GUI) içermediği için tüm kullanıcı etkileşimi menüler, metin girişleri ve doğrulama mesajları üzerinden gerçekleşmektedir.

### 4.1 Ana Menü Ekranı

=====

#### OTEL YÖNETİM SİSTEMİ

=====

1 – Oda Ekle 2 – Oda Listele 3 – Müşteri Ekle 4 – Müşteri Listele 5 – Rezervasyon Yap 6 – Rezervasyon Listele 7 – Check-in 8 – Check-out 0 – Çıkış

Seçiminiz: \_

yer kaplasin ister istemez alta alta olmasi lazimkardesim.....  
sifir yerine 9 yaz.....

### Açıklama:

- Sistem açıldığında kullanıcıyı karşılayan ana ekrandır.
- Bütün işlemlerin giriş noktasıdır.
- Kullanıcı işlem numarasını girerek ilgili bölüme yönlendirilir.

### Hatalı Giriş Durumu:

- Girilen değer 0–8 arasında değilse:

Hatalı seçim! Lütfen geçerli bir değer giriniz.

### 4.2 Oda Ekleme Ekranı

----- ODA EKLEME EKRANI -----

Oda Tipi: \_\_\_\_\_ Kapasite: \_\_\_\_\_ Günlük Fiyat: \_\_\_\_\_

[Kaydet] (Devam etmek için ENTER'a basınız)

### Açıklama:

Resepsiyon görevlisi bu ekranda yeni oda tanımlaması yapar.

### Kontroller:

- Kapasite → Sayısal olmalı
- Fiyat → Pozitif sayı olmalı
- Tip → Boş bırakılamaz

### Hatalı Giriş Örneği:

Hata: Fiyat bilgisi geçersiz!

### 4.3 Oda Listeleme Ekranı

----- ODA LİSTELEME -----

ID: Tip: Kapasite :Fiyat: Durum:



1 Standart 2 850 Boş 2 Deluxe 3 1200 Dolu 3 Suit 4 2500 Rezerve

Bir tuşa basarak ana menüye dönünüz...

### Açıklama:

- Sistem kayıtlı odaları tablo şeklinde gösterir.

### 4.4 Müşteri Ekleme Ekranı

----- MÜŞTERİ EKLEME -----

Ad: \_\_\_\_\_ Soyad: \_\_\_\_\_ Telefon: \_\_\_\_\_

[Kaydet]

### Hatalı Giriş:

Hata: Telefon numarası boş olamaz!

### 4.5 Müşteri Listeleme Ekranı

----- MÜŞTERİ LİSTELEME ----- ID Ad Soyad Telefon

1 Ali Yılmaz 0533 xxx xx xx 2 Ayşe Demir 0541 xxx xx xx

Bir tuşa basarak ana menüye dönünüz...

### 4.6 Rezervasyon Yapma

Rezervasyon işlemi 4 aşamalı bir süreç olarak tasarlanmıştır.

### Aşama 1 – Müşteri Seç

----- REZERVASYON: AŞAMA 1 ----- Lütfen müşteri ID'sini seçiniz:

### ID Ad Soyad

1 Ali Yılmaz 2 Ayşe Demir

Müşteri ID: \_

### Hatalı ID:

Hata: Bu ID'ye sahip müşteri bulunamadı!

### Aşama 2 – Oda Seç

----- REZERVASYON: AŞAMA 2 ----- Uygun odalar:

### ID Tip Kapasite Fiyat Durum

1 Standart 2 850 Boş 3 Suit 4 2500 Boş

Oda ID: \_

### Hatalı Seçim:

Hata: Seçilen oda rezerve edilemez.

### Aşama 3 – Tarih ve Ücret

----- REZERVASYON:

AŞAMA 3 ----- Giriş Tarihi (GG-AA-YYYY): \_\_\_\_\_ Çıkış Tarihi (GG-AA-YYYY):  
\_\_\_\_\_

Hesaplanan Ücret: 2550 TL Devam etmek için ENTER →

### Tarih Hatası:

Hata: Çıkış tarihi giriş tarihinden önce olamaz!

### Aşama 4 – Onay

----- REZERVASYON: AŞAMA 4 ----- Müşteri: Ali Yılmaz Oda: **Suit** (4 Kişilik)  
Toplam Ücret: 2550 TL Tarih: 10-05-2025 → 13-05-2025 Onaylıyor musunuz? (E/H): \_

### Onay verildikten sonra:

Rezervasyon başarıyla oluşturuldu!

## 4.7 Check-in Ekranı

----- CHECK-IN ----- Rezervasyon ID: \_\_\_\_\_

[Onayla]

### Başarılı:

Check-in tamamlandı. Oda artık 'Dolu' olarak işaretlendi.

## 4.8 Check-out Ekranı

----- CHECK-OUT ----- Rezervasyon ID: \_\_\_\_\_ Ödeme alındı mı? (E/H): \_

[Çıkış Tamamla]

### Başarılı:

Check-out tamamlandı. Oda artık 'Boş'.

## SONUÇ

- Sistem; oda yönetimi, müşteri kaydı, rezervasyon akışları ve check-in/check-out süreçlerini destekleyen bütünleşik bir mimari ile tasarlanmıştır.
- Tüm analiz ve tasarım adımları OOA/OOD prensipleri ile yürütülmüş, UML diyagramları (Use Case, Class, Sequence, Activity, Object) ile doğrulanmıştır.
- Sınıf yapısı modüler, servis katmanı genişletilebilir, dosya yönetimi ise basit ve kararlı şekilde yapılandırılmıştır.
- Komut satırı arayüzü süreçleri adım adım yönlendiren minimal bir kullanıcı deneyimi sunmaktadır.
- Ortaya çıkan mimari, hem akademik hem pratik açıdan uygulanabilir, esnek ve sürdürülebilir bir otel otomasyon altyapısı sağlamaktadır.