



**YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
YAZILIM MÜHENDİSLİĞİ DERSİ
DÖNEM PROJESİ**

AYDINLIK ÜNİVERSİTESİ KÜTÜPHANESİ BİLGİ SİSTEMİ

GRUP 7

Cihat İlker GENÇ - 12011068

Okan ŞEREF - 17011042

Uğur ALTINDAL - 17011043

Şeyma ÇINAR - 17011047

Ömer Buğrahan ÇALIŞKAN - 17011076

PROJE PLANI

PROJE ALAN TANIMI

Projemiz kütüphane bilgi sistemi olup kütüphanedeki yayınları kullanıcılara ödünç verme ve kullanıcıların rezervasyon işlemlerini gerçekleştirmek için yapılmıştır. Bu proje ile öğrenci, öğretim üyesi ve memurların kullanıcılar olarak Aydınlik Üniversitesi Kütüphanesinden ödünç alma ve rezervasyon işlemleri yapılmaktadır.

Modüller:

- **Kullanıcı Modülü:** Öğrenci Ekleme, Öğretim Üyesi Ekleme, Memur Ekleme, Kullanıcı Maili Ekleme, Kullanıcı Kart Bilgi ve Bakiyesi Ekleme, Kullanıcıları Güncelleme ve Silme
- **Yayın Modülü:** Ders Kitabı Ekleme, Kitap Ekleme, Dergi Ekleme, Yayınları Güncelleme ve Silme
- **Kiosk Modülü:** Yayını Okuma, Kullanıcı Kartı Okuma, Yayının Alındığı ve Verildiği Tarih Bilgisini Tutma, Kart Bakiyesi İşlemleri Ekleme
- **İşlem Modülü:** Kullanıcı Bilgisini Tutma, Yayın Bilgisini Tutma, Ödünç Alma İşlemini Ekleme, Tarih Bilgisini tutma
- **Rezervasyon Modülü:** Yayını ve Ödünç Alacak Kullanıcı Bilgisini Tutma

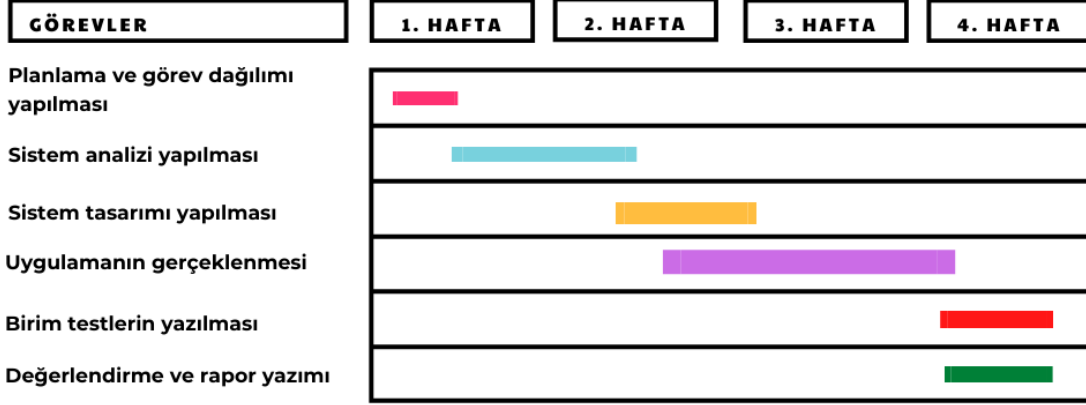
KABULLER VE KISITLAR

Bu proje gerçekleştirilirken Kiosk'un tüm kullanıcı ve yayınları önceden bildiği kabul edilmiştir.

GANTT DİYAGRAMI

AYDINLIK ÜNİVERSİTESİ KÜTÜPHANESİ BİLGİ SİSTEMİ

GANTT DİYAGRAMI

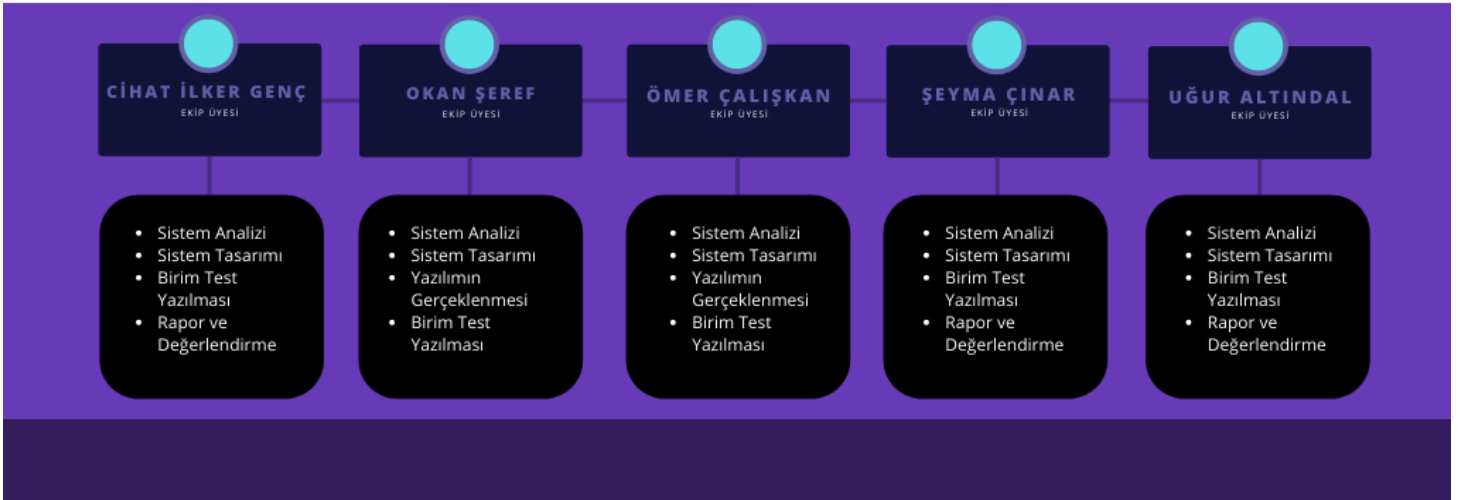


ORGANİZASYON ŞEMASI

ORGANİZASYON ŞEMASI

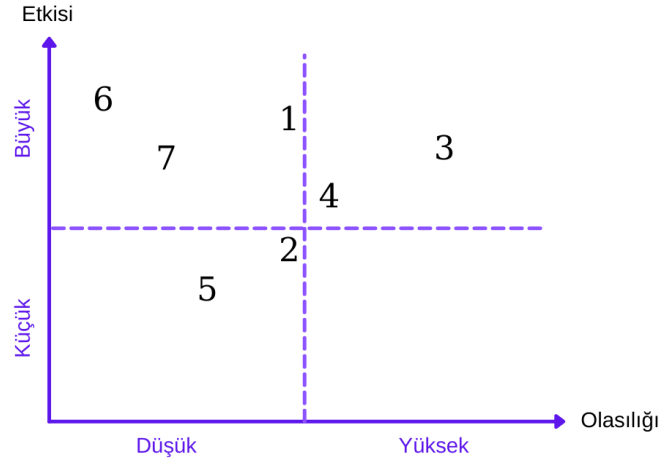
Grup 7

Aydınlık Üniversitesi Kütüphanesi Bilgi Sistemi Proje Ekibi



RİSK TABLOSU

Risk ID	Risk Adı	Türü	Etkisi	Olasılık
01	Yazılımsal ve donanımsal aksaklıklar	Teknik	Büyük	Orta
02	Yazılımın geç teslim edilmesi	Teknik	Orta	Orta
03	Yazılımın piyasada çok fazla alternatifinin olması	Ticari	Büyük	Yüksek
04	Müşterinin isteklerinin karşılanmaması	Teknik	Büyük	Orta
05	Görev dağılımından oluşan aksaklıklar	Proje	Küçük	Düşük
06	Yazılımın tamamlanamaması ve teslim edilememesi	Teknik	Büyük	Düşük
07	Kullanışlılık	Teknik	Büyük	Düşük



RİSK #02: Yazılımın geç teslim edilmesi

Olasılık : **Orta** Etki: **Orta** Türü: **Teknik**

Açıklama : Yazılımın planlanan tarihten geç teslim edilmesi

İşaretleri :

1. Projenin belirlenen programa uygun ilerlememesi.
2. Görev dağılımında aksaklıklar olması.
3. İşlerin planlanandan uzun sürmesi ve sarkması.
4. Bir işin uzamasından dolayı başka bir işe başlanamaması.

Önlemler :

1. Projeye başlamadan önce gerçekçi bir planlama yapılması
2. Müşteriyi kazanmak için olası olmayan, çok erken bir teslim tarihi verilmemesi.
3. Ekip üyelerinin plana bağlı kalarak çalışması ve sorumluluklarını gereken zamanda yerine getirmesi.

RİSK #03: Yazılım piyasada çok fazla alternatifinin olması

Olasılık : **Yüksek** Etki: **Büyük** Türü: **Ticari**

Açıklama : Yazılımın piyasada çok fazla alternatifinin olması

İşaretleri :

1. Projenin alternatifinin az olması sebebiyle ticari pazarlama aşamasında kullanıcı ya da müşteriyi ikna etmenin zorluğu.
2. Sistemin yaygın olarak bilinmesi ve alternatiflerinin yaygın olması sebebiyle piyasadaki ticari değerinin düşmesi sebebiyle proje masraflarının kendini amorte etme süresinin belirsizliği

Önlemler :

1. Pazarlama kısmında çeşitli kampanyalar; reklam kısmında ise sistemin güvenilirliği ve kullanılabilirliği üzerine gerekli dijital tanıtım ortamlarının önceden planlanması
2. Müşteri ile satış öncesi iletişim kurularak etkileşim halinde müşterinin ana istekleri doğrultusunda düzenlemeler yapılması, beklentinin en yüksek olduğu işlevler üzerinde kullanıcı kolaylığı sağlayan iyileştirmeler yapılması.

İSTEKLERİN MODELLENMESİ

KULLANIM SENARYOSU

Kullanım Senaryosu: Ödünç Alma İşlemi

Birincil Aktör: Kullanıcı

İlgililer ve İlgili Alanları:

Kiosk: Kütüphane sistemidir.

Öğrenci, Öğretim Üyesi, Memur: Kütüphane kullanıcılarıdır. Ödünç alma ve rezervasyon işlemlerini sorunsuz bir şekilde gerçekleştirmek ister.

Ön Koşullar: Tüm kullanıcı ve yayın bilgileri sistemde bulunur.

Son Koşullar: Kullanıcıya istediği parçalar ödünç olarak verilir.

Ana Senaryo:

1. Kullanıcı kütüphaneye gelir.
2. Kullanıcı ödünç almak istediği parça ile birlikte kioska gider.
3. Kullanıcı önce ödünç almak istediği parçayı, daha sonra akıllı kartını kioska okutur.
4. Sistem, kullanıcıya parçayı ödünç verir.

Alternatif Senaryolar:

2.a. İade Senaryosu

1. Kullanıcı iade etmek istediği parça ile birlikte kioska gider.
2. Kullanıcı önce iade edeceği parçayı, daha sonra akıllı kartını okutur.
3. Sistem iade işlemini gerçekleştirir.
4. Eğer teslim süresi geçmiş ise geçen her gün için sistem 1 TL ceza uygular.
5. Toplam ceza tutarı, kullanıcının akıllı kartındaki bakiyesinden düşülür.

2.b. Rezervasyon Senaryosu

1. Kullanıcı kiosktan ilgili parçanın kodunu girer ve kendi kartını okutur.
2. Kullanıcı ilgili parçayı rezerve eder.
3. İlgili parça kütüphaneye iade edildiğinde sistem, kullanıcıya e-posta göndererek haber verir.

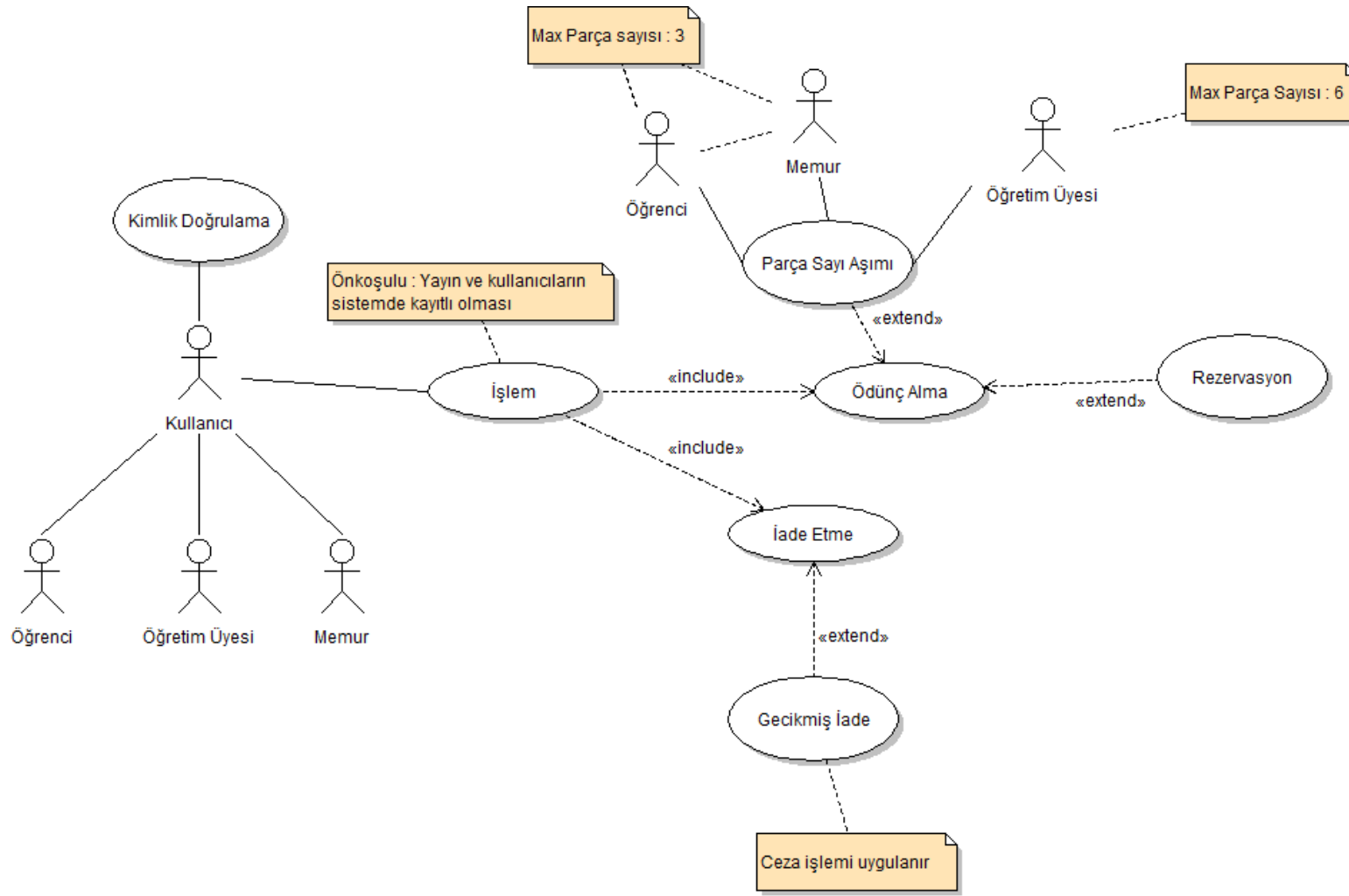
4.a. Kullanıcının Ödünç Alamama Senaryosu

1. Kullanıcı üzerindeki mevcut parçaların kontrolü yapılır.
2. Öğrenci ve memurlar aynı anda en fazla 3, öğretim üyeleri en fazla 6 kitap ödünç alabilir.
3. Kullanıcı üzerinde limitinden fazla parça varsa ödünç alma işlemi gerçekleşmez.

4.b. Kullanıcının Ödünç Alamama Senaryosu

1. Kullanıcı üzerindeki mevcut parçaların teslim tarihi kontrol edilir.
2. Öğrenci ve memurların ödünç alma süresi 1 ay, öğretim üyelerinin ki ise 3 aydır.
3. Kullanıcı üzerinde teslim tarihi geçmiş parça varsa ödünç alma işlemi gerçekleşmez.

USE-CASE DİYAGRAMI

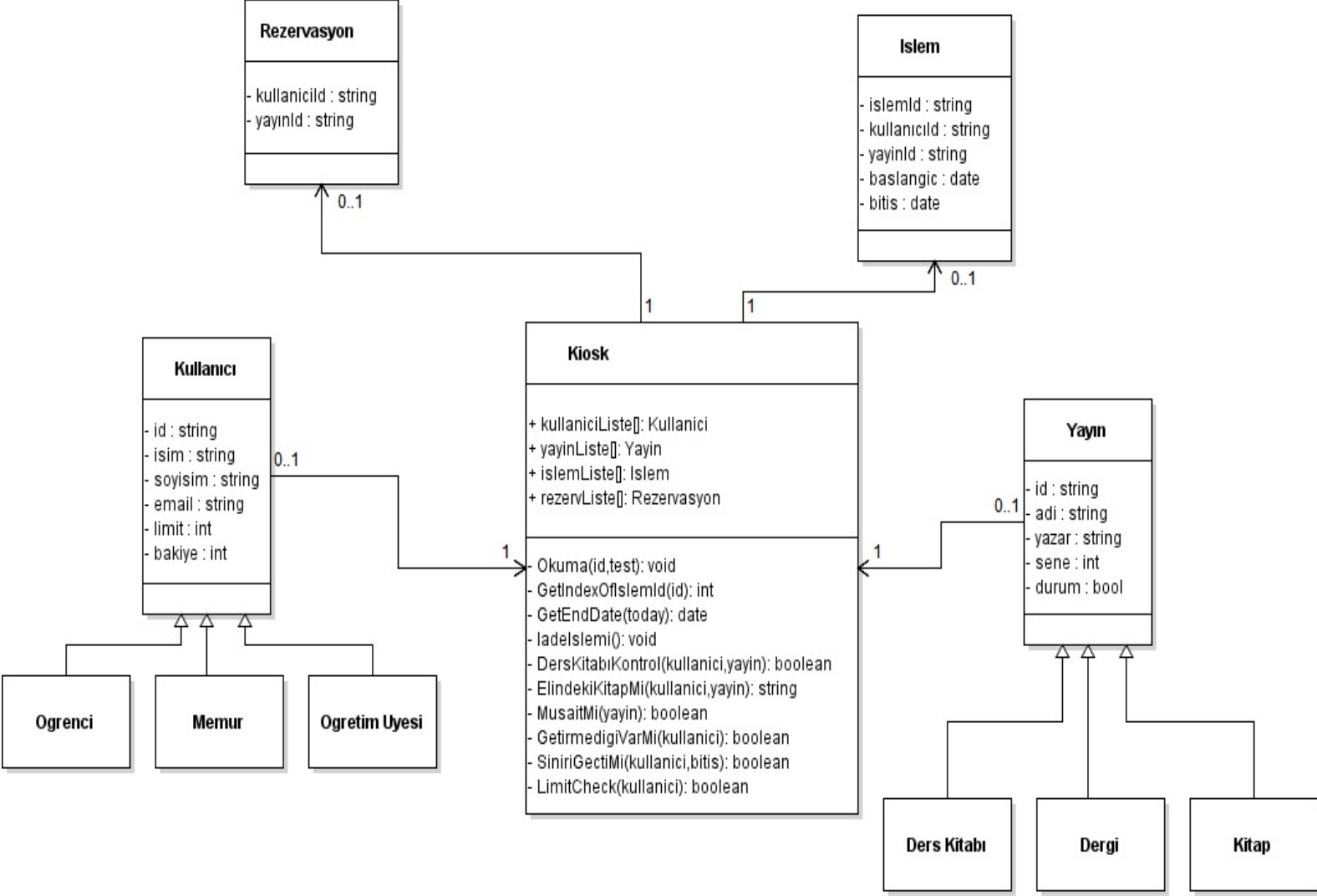


İZLENEBİLİRLİK TABLOSU

	Yayın	Kullanıcı	Rezervasyon	İşlem	Kiosk	testCase
Ödünç alma	✓	✓		✓	✓	
Rezervasyon	✓	✓	✓		✓	
Ceza işlemi		✓			✓	
İade	✓	✓		✓		

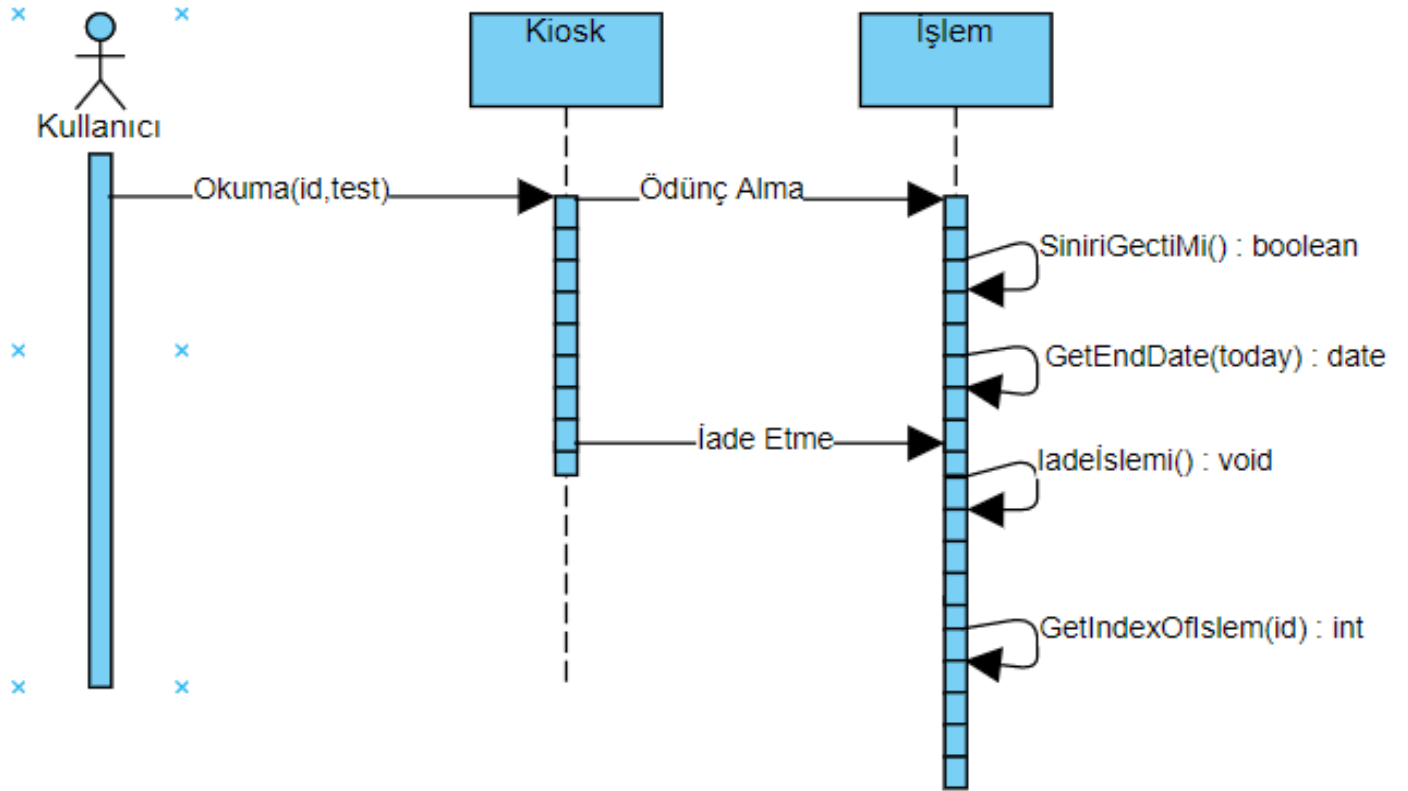
NESNEYE DAYALI MODELLEME

SINIF DİYAGRAMI



NESNEYE DAYALI TASARIM

SIRALAMA (SEQUENCE) DİYAGRAMI

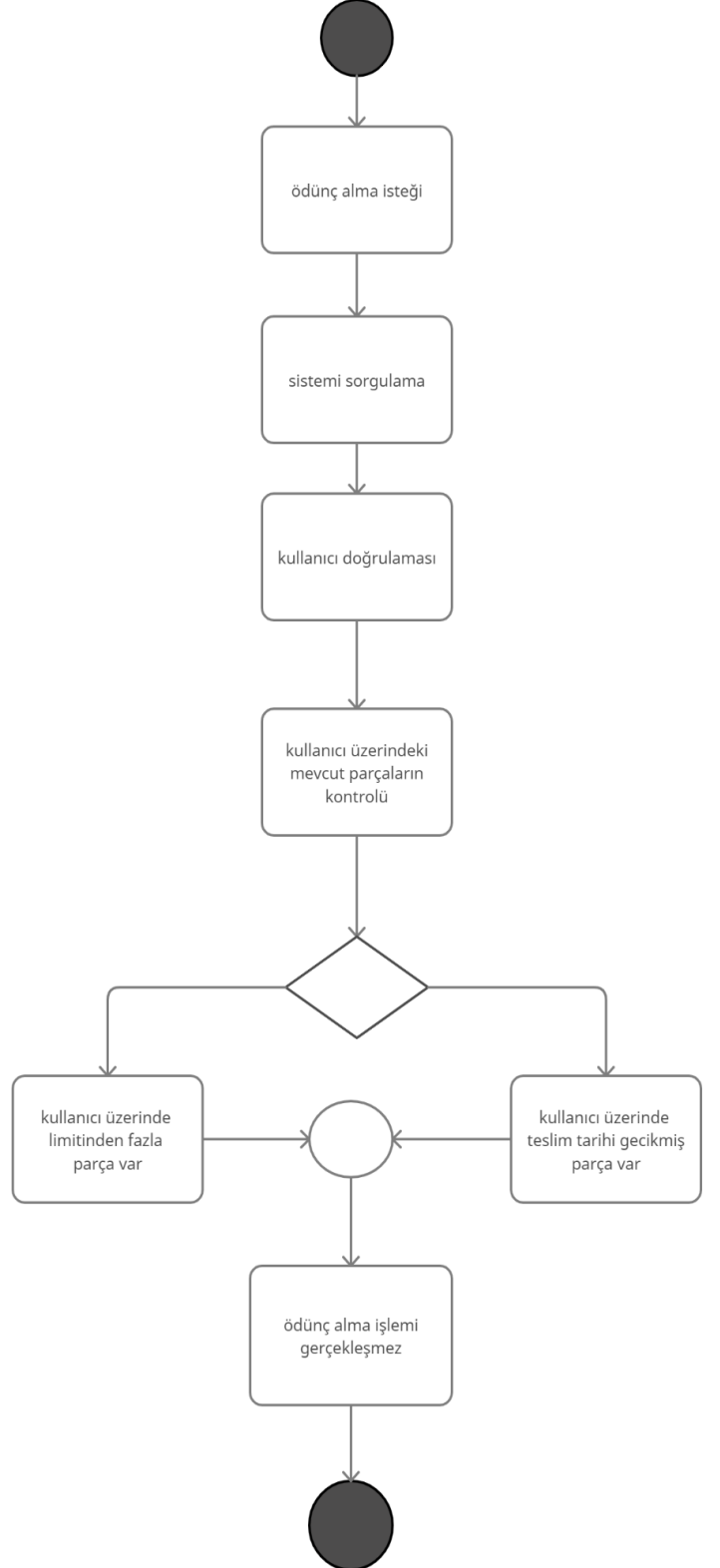


ETKİNLİK (ACTIVITY) DİYAGRAMI

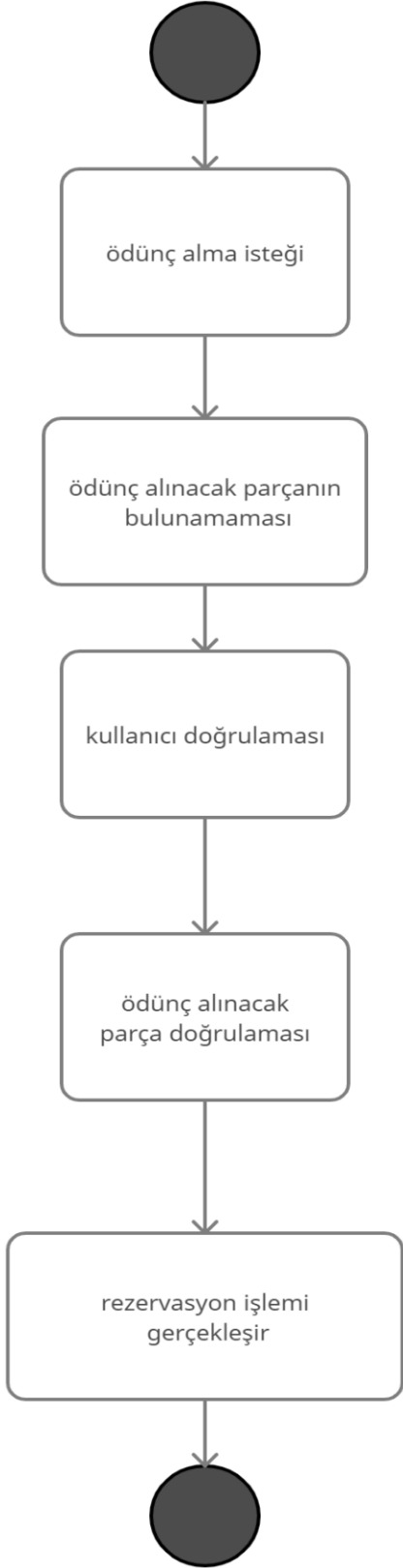
Ödünç alma senaryosu



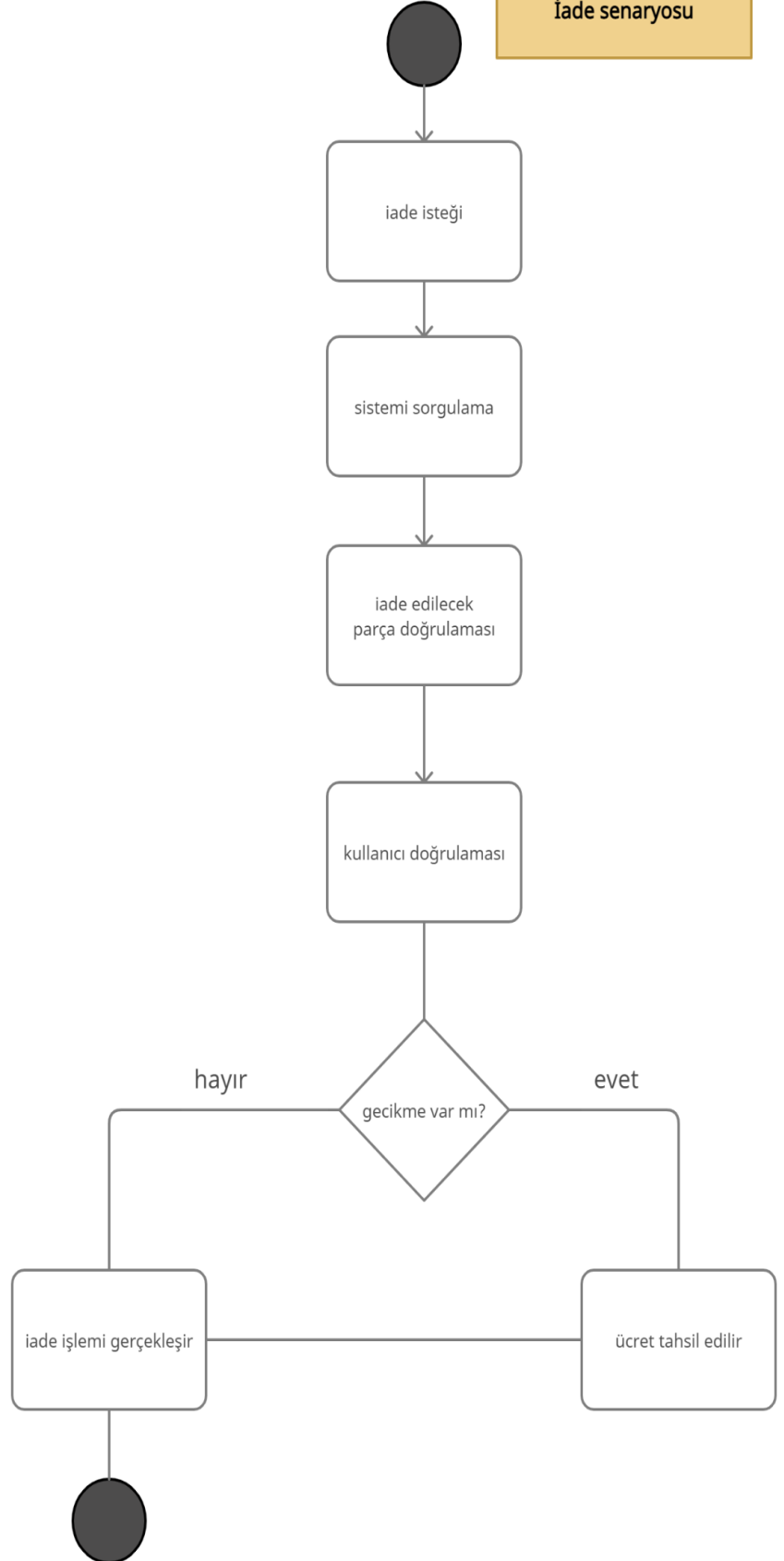
Ödünç alamama senaryosu



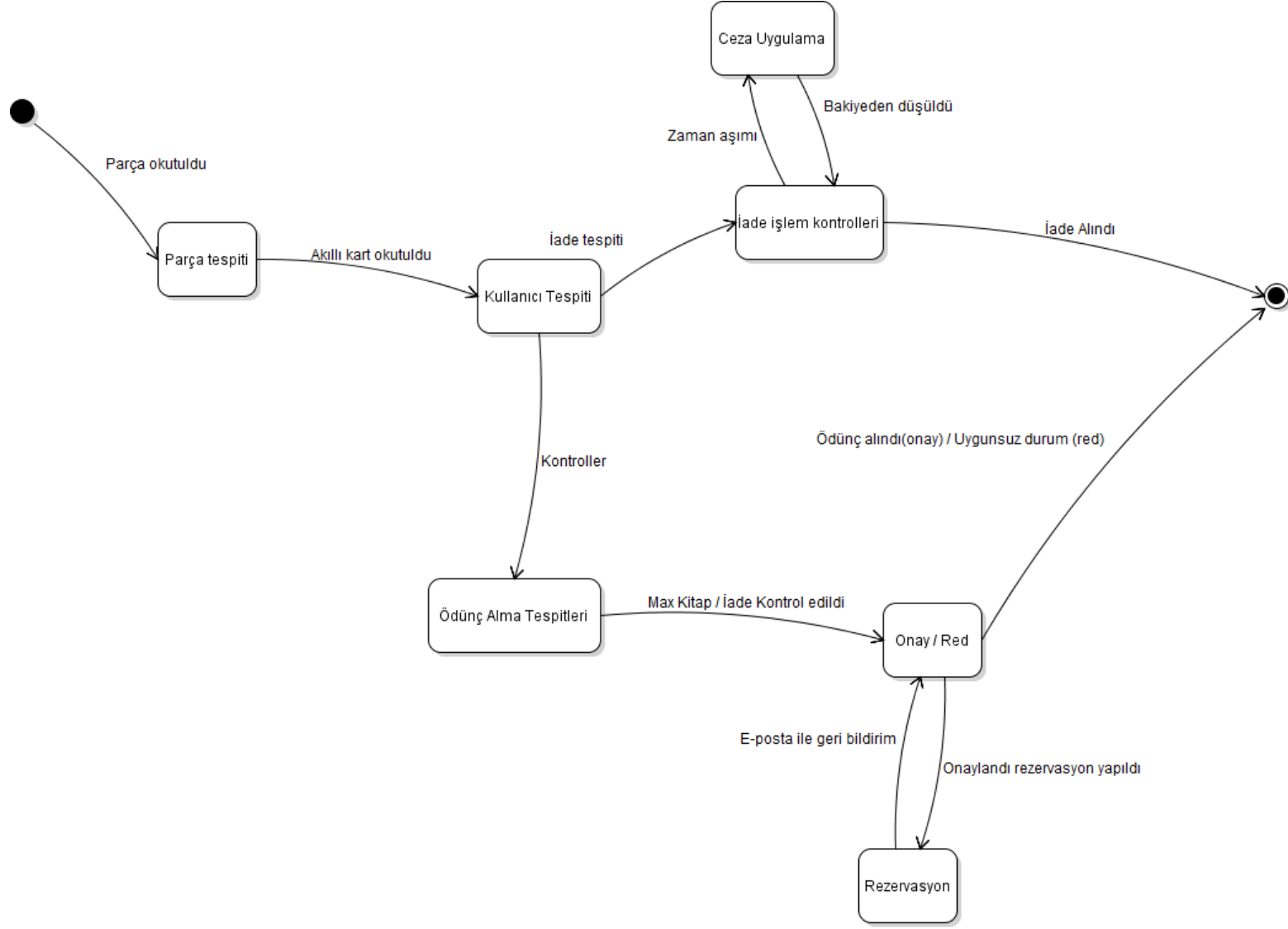
Rezervasyon senaryosu



İade senaryosu



DURUM (STATE) DİYAGRAMI



BİRİM TESTİ SINAMALARI

Test 1 : Ders Kitabı Alım Testi --Cihat İlker GENÇ - 12011068

```
@Test
void test() {
    //Ders kitabını sadece öğretim görevlisi alır
    Kullanici m=new Memur("aaa", "ahmet", "as", "asdfg", 1, 100);
    Yayin k=new DersKitabi("d", "bok", "asd", 1, false);
    assertEquals(false, kiosk.DersKitabiKontrol(m,k));
}
```

Test 2 : Getirilmemiş Kitap Testi (Memur için) --Cihat İlker GENÇ - 12011068

```
@Test
void test2() {
    //Getirmediği kitap varsa true. Memur olduğu için 30 gün
    kiosk.kullaniciListe = new ArrayList<>();
    kiosk.islemListe = new ArrayList<>();
    kiosk.kullaniciListe.add(new Memur("aaa", "ahmet", "as", "asdfg", 1, 100));
    kiosk.islemListe.add(new Islem("0", "aaa", "d", new Date(121, 2, 10), new Date(121, 3, 10)));
    LocalDateTime now = LocalDateTime.now();
    kiosk.currentDate=new Date(now.getYear()-1900,now.getMonthValue()-1,now.getDayOfMonth());

    assertEquals(true, kiosk.GetirmedigiVarMi(kiosk.kullaniciListe.get(0)));
}
```

Test 3 : Getirilmemiş Kitap Testi (Öğretim Üyesi için) --Okan ŞEREF - 17011042

```
@Test
void test3() {
    //Getirmediği kitap varsa true. Öğretim üyesi olduğu için 90 gün
    kiosk.kullaniciListe = new ArrayList<>();
    kiosk.islemListe = new ArrayList<>();
    kiosk.kullaniciListe.add(new OgretimUyesi("aaa", "ahmet", "as", "asdfg", 1, 100));
    kiosk.islemListe.add(new Islem("0", "aaa", "d", new Date(121, 3, 10), new Date(121, 4, 18)));
    LocalDateTime now = LocalDateTime.now();
    kiosk.currentDate=new Date(now.getYear()-1900,now.getMonthValue()-1,now.getDayOfMonth());

    assertEquals(false, kiosk.GetirmedigiVarMi(kiosk.kullaniciListe.get(0)));
}
```

Test 4 : Kitap ID döndürme testi --Okan ŞEREF - 17011042

```
@Test
void test4() {
    //Elindeki kitapsa işlem ID döndürür
    kiosk.islemListe = new ArrayList<>();
    Kullanici k=new OgretimUyesi("aaa", "ahmet", "as", "asdfg", 1, 100);
    Yayin y=new DersKitabi("d", "hey", "asd", 1, false);
    kiosk.islemListe.add(new Islem("0", "aaa", "d", new Date(121, 3, 10), new Date(121, 4, 18)));

    assertEquals("0", kiosk.ElindekiKitapMi(k,y));
}
```

Test 5 : Limit Kontrol Testi -- Uğur ALTINDAL - 17011043

```
@Test
void test5() {
    //Limit kontrolü 3 ve 6
    kiosk.islemliste = new ArrayList<>();
    Kullanici k=new OgretimUyesi("aaa", "ahmet", "as", "asdfg", 5, 100);
    Kullanici k2=new Memur("aaa", "ahmet", "as", "asdfg", 2, 100);
    Kullanici k3=new Ogrenci("aaa", "ahmet", "as", "asdfg", 3, 100);

    assertEquals(true, kiosk.LimitCheck(k));
    assertEquals(true, kiosk.LimitCheck(k2));
    assertEquals(false, kiosk.LimitCheck(k3));
}
```

Test 6 : Kitap İadesi Uygunluk Durum Testi --Uğur ALTINDAL - 17011043

```
@Test
void test6() {
    //Kitabı geri verdik ve müsait hale geldi
    kiosk.kullaniciListe = new ArrayList<>();
    kiosk.islemliste = new ArrayList<>();
    kiosk.rezervListe = new ArrayList<>();
    kiosk.yayinListe = new ArrayList<>();

    kiosk.yayinListe.add(new Kitap("d", "hey", "asd", 1, false));
    kiosk.kullaniciListe.add(new Memur("aaa", "ahmet", "as", "asdfg", 1, 100));
    kiosk.islemliste.add(new Islem("0", "aaa", "d", new Date(121, 3, 10), new Date(121, 4, 18)));
    LocalDateTime now = LocalDateTime.now();
    kiosk.currentDate=new Date(now.getYear()-1900,now.getMonthValue()-1,now.getDayOfMonth());

    kiosk.Okuma("d",true);
    kiosk.Okuma("aaa",true);

    assertEquals(true, kiosk.MusaitMi(kiosk.yayinListe.get(0)));
}
```

Test 7 : İade Alınan Kitabın Tekrar Ödünç Alınma Testi --Şeyma ÇINAR – 17011047

```
@Test
void test7() {
    //Kitabı geri verdik ve baskası aldı müsait mi
    kiosk.kullaniciListe = new ArrayList<>();
    kiosk.islemliste = new ArrayList<>();
    kiosk.rezervListe = new ArrayList<>();
    kiosk.yayinListe = new ArrayList<>();

    kiosk.yayinListe.add(new Kitap("d", "hey", "asd", 1, false));
    kiosk.kullaniciListe.add(new Memur("aaa", "ahmet", "as", "asdfg", 1, 100));
    kiosk.kullaniciListe.add(new Ogrenci("bbb", "memet", "as", "asdfg", 0, 100));

    kiosk.islemliste.add(new Islem("0", "aaa", "d", new Date(121, 3, 10), new Date(121, 4, 18)));
    LocalDateTime now = LocalDateTime.now();
    kiosk.currentDate=new Date(now.getYear()-1900,now.getMonthValue()-1,now.getDayOfMonth());

    kiosk.Okuma("d",true);
    kiosk.Okuma("aaa",true);
    kiosk.Okuma("d",true);
    kiosk.Okuma("bbb",true);
    assertEquals(false, kiosk.MusaitMi(kiosk.yayinListe.get(0)));
}
```

Test 8 : Teslim Tarihi Geçmiş İade Testi --Şeyma ÇINAR - 17011047

```
@Test
void test8() {
    //Kitabı vaktinde getirmeyen biri
    kiosk.kullaniciListe = new ArrayList<>();
    kiosk.islemListe = new ArrayList<>();
    kiosk.kullaniciListe.add(new Memur("aaa", "ahmet", "as", "asdfg", 1, 100));
    Islem i = new Islem("0", "aaa", "d", new Date(121, 0, 10), new Date(121, 2, 18));
    kiosk.islemListe.add(i);
    assertEquals(true, kiosk.SiniriGectiMi(kiosk.kullaniciListe.get(0),i.getBitis()));
}
```

Test 9 : Teslim Tarihi Geçmiş İade Testi (tarihli kontrol)

--Ömer Buğrahan ÇALIŞKAN - 17011076

```
@Test
void test9() {
    //Kitabı vaktinde getirmeyen biri
    kiosk.kullaniciListe = new ArrayList<>();
    kiosk.islemListe = new ArrayList<>();
    kiosk.yayinListe = new ArrayList<>();
    kiosk.kullaniciListe.add(new Memur("aaa", "ahmet", "as", "asdfg", 1, 100));
    kiosk.yayinListe.add(new Kitap("d", "hey", "asd", 1, false));
    LocalDateTime now = LocalDateTime.now();
    kiosk.currentDate=new Date(now.getYear()-1900,now.getMonthValue()-1,now.getDayOfMonth());
    Islem i = new Islem("0", "aaa", "d", new Date(121, 0, 10), new Date(121, 4, 18));
    kiosk.islemListe.add(i);
    kiosk.Okuma("d", true);
    kiosk.Okuma("aaa", true);
    kiosk.Okuma("d", true);
    kiosk.Okuma("aaa", true);
    assertEquals(false, kiosk.SiniriGectiMi(kiosk.kullaniciListe.get(0),i.getBitis()));
}
```

Test 10 : Tarihe Göre Ceza Testi --Ömer Buğrahan ÇALIŞKAN - 17011076

```
@Test
void test10() {
    //Kitabı vaktinde getirmeyen biri tarihli ceza
    kiosk.kullaniciListe = new ArrayList<>();
    kiosk.islemListe = new ArrayList<>();
    kiosk.yayinListe = new ArrayList<>();
    kiosk.rezervListe = new ArrayList<>();
    kiosk.kullaniciListe.add(new Memur("aaa", "ahmet", "as", "asdfg", 1, 100));
    kiosk.yayinListe.add(new Kitap("d", "hey", "asd", 1, false));
    LocalDateTime now = LocalDateTime.now();
    kiosk.currentDate=new Date(now.getYear()-1900,now.getMonthValue()-1,now.getDayOfMonth());
    Islem i = new Islem("0", "aaa", "d", new Date(121, 1, 10), new Date(121, 2, 10));
    kiosk.islemListe.add(i);
    kiosk.Okuma("d", true);
    kiosk.Okuma("aaa", true);
    assertEquals(true,kiosk.kullaniciListe.get(0).getBakiye()<50);
}
```


Finished after 0,13 seconds

Runs: 10/10 ❌ Errors: 0 ❌ Failures: 0

testCase [Runner: JUnit 5] (0,000 s)

- test100 (0,000 s)
- test() (0,000 s)
- test20 (0,000 s)
- test30 (0,000 s)
- test40 (0,000 s)
- test50 (0,000 s)
- test60 (0,000 s)
- test70 (0,000 s)
- test80 (0,000 s)
- test90 (0,000 s)

Failure Trace

Finished after 0,13 seconds

Runs: 10/10 ❌ Errors: 0 ❌ Failures: 1

testCase [Runner: JUnit 5] (0,056 s)

- test100 (0,031 s)
- test() (0,001 s)
- test20 (0,015 s)
- test30 (0,000 s)
- test40 (0,001 s)
- test50 (0,001 s)
- test60 (0,000 s)
- test70 (0,004 s)
- test80 (0,000 s)
- test90 (0,000 s)

Failure Trace

! org.opentest4j.AssertionFailedError: expected: <false> but was: <true>
at testCase.test2(testCase.java:33)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)