

**GAZİ ÜNİVERSİTESİ**  
**TEKNOLOJİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BMT – 310 YAZILIM PROJE YÖNETİMİ**

**YONGA BASKET ANALYSIS**

**TEST PLANI**

**161816057 Şükriye ÇAVDAR**

**171816046 Şeyma Sultan SÖZEN**

**161816060 Okan KOCATÜRK**

**161816076 Emre TARI**

## 1. GİRİŞ

Yazılım test süreci önceden planlanan, sonra icra edilip bildirilen ve resmi olarak belgelendirilen bir grup eylemden oluşur. Bu süreç, geliştirilen yazılımın içerisindeki olası hataların varlığına odaklanır. Ancak testler hatanın hangi kod parçalarından kaynaklandığını saptamayı amaçlamazlar. Hatanın kodun neresinden kaynaklandığının belirlenmesi işlemi hata ayıklama ile ilişkilidir. Bu da izlenebilirlik ve hata ayıklama ile gerçekleştirilir.

Resim 1: Yazılım Test Süreci



## 2. ANALİZ

Projemizi marketler ve E-Ticaret siteleri kullanabilecektir. Projemizde, müşterilerin alışverişleri kayıt altında tutularak bundan sonraki alışverişlerinde buna göre ürün önerilecektir. Bu ürün önerme işlemi, müşterinin aldığı ürüne karşılık en çok aldığı ürün indirimli olarak önerilecektir. Böyle hem müşteri hem de satıcı kazanç sağlayacaktır.

Projemiz için gerekli yazılım gereksinimleri java için Eclipse, python için PyCharm ve verilerin tutulması için Oracle Veri Tabanıdır. Donanım olarak sistem gereksinimleri aşağıdaki gibi olmalıdır:

**Tablo 1:** En düşük sistem gereksinimleri ile önerilen sistem gereksinimlerini belirtir tablo

Gereksinimler	En Düşük	Önerilen
RAM	5,3 GB boş RAM	8 GB toplam sistem RAM'i
Disk Alanı	4,4 GB	En az 8 GB SSD sürücü
İşlemci	Pentium 2 266 MHz işlemci	Intel Core i5 4460 3.2GHz
İşletim Sistemi	Microsoft Windows 8 veya üzeri, macOS 10.13 veya üstü, Gnome, KDE veya Unity DE'yi destekleyen herhangi bir Linux dağıtımı	Windows, macOS veya Linux'un en son 64 bit sürümü

### 3. TEST STRATEJİSİ

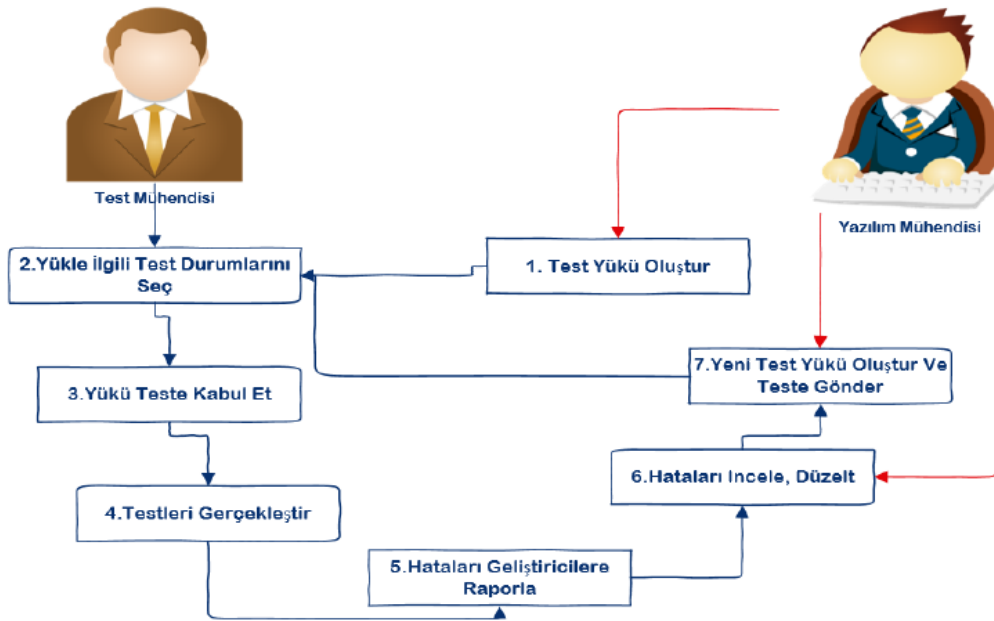
Öncelikle geliştirdiğimiz projeyi özelliklerine göre (risk, sonuç vb.) seviyelere ayrıldı. Daha sonra bu özelliklere göre proje modüllerine önem derecesi tanımlandı. Bu önem derecesi kavramı temel alınarak, test sürecindeki dokümantasyon ve faaliyetlerin derinliği belirlendi.

Analizler incelendiğinde gerekli test yöntemleri birim test, entegrasyon testi ve sistem testi olarak belirlenmiştir. Test sürecinin ilerleyişine ait aşamalar aşağıdadır:

- Test durumlarını, test prosedürlerini belirlemek
- Testin tamamlanması için yapılacakları adım adım belirtmek
- Testler için gerekli test verilerini üretmek ve doğrulamak
- Test durumunda belirtilen tüm adımların gerçekleştirilmesinden sonra hangi sonuç ile karşılaşılacağına belirtilmesi
- Birim testlerin gerçekleştirilmesini izlemek ve sonuçlarını denetlemek
- Hata bildirim yapısını tanımlamak ve işlerliğini denemek
- Testler için gerekli test ortamını tanımlamak

- Test durumları koşturulduktan sonra gerçekleştirilecek olan eylemleri yani sonlandırma koşullarını belirlemek
- Test gerçekleştirildikten sonra ortaya çıkan sonuç ile beklenen sonucun karşılaştırılması ve testin geçip kaldığının belirlenmesi
- Bulunan hataları önem dereceleri, hataların kaynağı ve yapılan düzenlemeleri ile raporlamak
- Bulunan hataların düzeltildiğini denetlemek ve yineleme testlerini gerçekleştirmek
- Testler gerçekleştirildikten sonra elde edilen test verilerini raporlamak, analiz etmek ve değerlendirmek

**Resim 2: Test akışındaki roller**



#### 4. TEST HEDEFLERİ

Test planlamanın en önemli adımı testlerle neyin amaçlandığının belirlenmesidir. Test amaçlarının belirlenmesi aşamasında testten beklenenler, test için kritik başarı faktörleri, test faaliyetlerindeki karşılaşılabilecek kısıtlar, testin kapsamı, test sonrası sonuçların analiz ve değerlendirilmesinin nasıl yapılacağı gibi konular tanımlanır.

Projemize ait test edilecek yazılım özellikleri işlevsellik, performans, doğruluktur. Test hedefleri şunlardır:

- **Durum 1:** Veriseti geliştirilecek yazılım için uygun mu?
- **Durum 2:** Seçilen birliktelik analizi algoritması geliştirilecek yazılım için uygun mu?
- **Durum 3:** Veriseti doğru okunuyor mu?
- **Durum 4:** Veriseti ilgisiz değerler içeriyor mu?
- **Durum 5:** Verisetinde null değerler bulunuyor mu?
- **Durum 6:** Veriseti yapılacak işlemler için uygun şekilde sınıflandırılmış mı?
- **Durum 7:** Veritabanı bağlantısı başarılı mı?
- **Durum 8:** Yeni müşteriye önerilecek ürünler ve alışverişi olan müşterilere önerilecek ürünler birbirinden ayrı tablolarda tutuluyor mu?
- **Durum 9:** Yeni müşteri veritabanına kayıt ediliyor mu?
- **Durum 10:** Girilen ID'ye ait bir müşteri var mı?
- **Durum 11:** Bu müşteriye öneri yapabilmek için ürün listesi dolu mu, alışveriş yapılmış mı?
- **Durum 12:** Müşterinin aldığı ürüne önerilen ürün doğru mu?
- **Durum 13:** Müşterinin ilk alışverişine öneri yapılmış mı?
- **Durum 14:** KVKK hükümlerine aykırılık bulunuyor mu?

## 5. TEST ÖLÇÜTLERİ

**Durum 1:** Seçilen veriseti fizibilite raporunda belirtilen kriterlere göre incelenmeli, uygun bulunmazsa veriseti değiştirilmelidir.

**Durum 2:** Algoritmaların çalışma mantıkları ile sistem kaynaklarını ne kadar kullandıkları karşılaştırılmalıdır. Sonuç elde etme süreleri de kıyaslandıktan sonra, yazılımın üzerinde çalışacağı sistem için en uygun algoritmaya karar verilmelidir.

**Durum 3:** Kullanılan geliştirme ortamına ait bir özellik olan Variable Explorer penceresi kontrol edilmelidir. Eğer burada bir değişken oluşmamış ise test durdurulmalıdır.

**Durum 4:** Kullanılan veriseti, geliştirilecek olan yazılım için ilgisiz değerler içeriyorsa bunlar temizlenmelidir. Bu test yazılım geliştirmenin ilk aşamasında gerekli görülmüştür ancak ilerleyen aşamalarda bu teste ihtiyaç duyulmamıştır.

**Durum 5:** Kullanılan veriseti analizi etkilememesi için boş değerlerden temizlenmelidir. Eğer verisetinde boş değer içeren satırlar bulunuyorsa test durdurulmalı ve bu değerler temizlenmeden işlemlere devam edilmemelidir.

**Durum 6:** Kullanılan veriseti nesneye yönelik programlama yapısına uygun olarak doğru sınıflara ayrılmalıdır. Sınıf yapısı ve özellikleri geliştirilecek yazılıma uygun seçilmelidir. Eğer uygun değilse sınıflar yeniden yapılandırılana kadar teste ara verilmelidir.

**Durum 7:** Veritabanı bağlantısı başarısız ise test durdurulmalıdır. Bağlantı için gerekli ortam koşulları kontrol edilmelidir.

**Durum 8:** Kişiyi özel analiz edilmiş ürünler ayrı bir tablodan alınmıyorsa test durdurulmalı ve hata verilmelidir. Tablolar yeniden düzenlenmelidir. Bu durum düzeltilene kadar test devam ettirilmemelidir.

**Durum 9:** Yeni kayıt olan müşteriye ID numarası verilmiyor ise test durdurulmalıdır. Ayrıca müşteriye ait ID numarasının müşteriler tablosunda bulunup bulunmadığı da kontrol edilmelidir. Eğer bu şartlardan bir tanesi bile sağlanmıyorsa teste devam edilmemelidir.

**Durum 10:** Bu duruma ait kontroller bir önceki test durumunda zaten kontrol edilmiş olacağı için bu durum gereksiz görülmüş ve iptal edilmiştir.

**Durum 11:** Müşteriye öneri yapabilmek için sipariş listesinin dolu olması gerekir. Müşterinin hiç ürünü bulunmazsa hata alınacaktır.

**Durum 12:** Müşteriye öneri doğru bir şekilde yapıldıysa test başarıyla sonlandırılacaktır.

**Durum 13:** Müşterinin ilk alışverişine öneri yapılmalıdır. Aksi takdirde test durdurulmalı ve bu durum çözülmelidir.

**Durum 14:** İş birliği yapılacak market ya da E-Ticaret sitesinin KVKK sözleşmesine aykırı durumların varlığı kontrol edilmelidir. Böyle bir aykırılık bulunuyorsa test durdurulmalıdır. Sözleşmeye uygun şartlar oluşturulana kadar teste devam edilmemelidir.

## 6. KAYNAK PLANLAMASI

Bu projenin gerçekleşebilmesi için yeterli çalışan ve ekipman olmalıdır. Projemizde Tasarım Mimarı, Programcı, Test Uzmanı ve Proje Yürütücüsü ile birlikte 4 çalışan bulunmaktadır. Ekipman olarak her çalışanın kendine özel bilgisayarını bulunmalıdır. Ayrıca Güvenlik Lisansları, Veri Tabanı Lisansları, İşletim Sistemi Lisansları sağlanmalıdır.

Projenin pazarlanması ve gerçekleştirilecek seyahatler için gerekli bütçe ayarlaması yapılmalıdır.

## 7. TEST ORTAMI

Günümüz teknoloji ihtiyaçlarının yazılım alanına yansımaları da hızlı yenilikler olmaktadır. Bu tempoya uyum sağlamak hızlı ve kaliteli bir yazılım geliştirme sürecini ortaya çıkarmaktadır. Bu akış içerisinde de kaliteyi ortaya çıkarabilmek için de test süreçleri ve ortamının da aynı tempoda uyumlu olması bir ihtiyaç arz etmektedir.

Fiziksel test ortamlarının uyumlu hale getirilmesi her zaman bir sorun ve zorluk olarak baş göstermiştir. Bu sorunla baş etmek için yapılan fiziksel donanım yatırımları hem pahalı hem de belli aşamadan sonra hantallaşmaya neden olmaktadır. Tam da bu noktada, test durumlarının koşturulabilmesi için geliştirme ortamlarıyla entegre çalışan yapılar kullanılacaktır. Java Programlama Dilinde JUnit, Python üzerinde UnitTest araçları kullanılacaktır.

Test yöneticisi ve sorumlu test mühendisi tarafından ihtiyaç görülürse aşağıda tanımlanan yardımcı araçlar da test sürecine dahil edilebilecektir.

**Test Verisi Üreteçleri:** Test durumlarının koşturulması için gerekli olan test girdi verilerini üreten yazılımlardır.

**Hata Ayıklayıcılar (Debugger):** Testler sırasında karşılaşılan hataların kaynak kod üzerinde bulunmasını sağlayan yazılımlardır. Hata yakalayıcılar ise sadece yazılımların gerçek çalışma ortamlarındaki çalışma zamanlarında faal olup yazılımın üzerinde meydana gelen bellek taşması, karşılaştırma hatası, yanlış boyutta değişken kullanılması gibi hataları tespit ve teşhis eder. Yani sadece hatayı değil, hatanın oluşmasına neden olan gerçek sebebin ne olabileceğine dair bilgi verir.

**Emülatör ve Simülatörler:** Yazılımların ihtiyaç duyduğu gerçek donanımları taklit eden ve testler için gerekli olan donanım verilerini sağlayan yazılımlardır.

**Koçan (Stub) ve Sürücüler:** Testler için gerekli olan sistemin işlevsel olmayan bileşenlerinin yerini tutan ufak yazılım parçacıklarıdır.

**Testlerin yapıldığı bilgisayarın sistem özellikleri şöyledir:**

- i. İşletim Sistemi: Windows 10 64bit,
- ii. Directx Versiyonu: 11
- iii. CPU: Intel Core i7-7700HQ
- iv. Anakart: AMD 770
- v. RAM: 16 GB DDR4
- vi. Ekran Kartı: Nvidia GeForce GTX 560
- vii. GPU: Nvidia GTX 1060 Max-Q Design 6 GB
- viii. Hard Disk: 1 TB SATA HDD ve 256 GB PCIe SSD
- ix. Klavye ve Mouse: Standard PS/2 Keyboard, HID-compliant Mouse



## 8. ZAMANLAMA VE TAHMİN

**Tablo 2:** Test durumlarına ait zamanlama

Durum	Başlama Tarihi	Bitiş Tarihi	
		Planlanan	Gerçekleşen
<b>1</b>	15.10.2019	16.10.2019	16.10.2019
<b>2</b>	17.10.2019	25.10.2019	25.10.2019
<b>3</b>	06.02.2020	01.04.2020	22.04.2020
<b>4</b>	24.04.2020	02.04.2020	24.04.2020
<b>5</b>	27.04.2020	04.04.2020	29.04.2020
<b>6</b>	30.04.2020	21.04.2020	11.05.2020
<b>7</b>	12.05.2020	22.04.2020	12.05.2020
<b>8</b>	12.05.2020	24.04.2020	12.05.2020
<b>9</b>	13.05.2020	28.04.2020	13.05.2020
<b>10</b>	13.05.2020	30.04.2020	13.05.2020
<b>11</b>	14.05.2020	06.05.2020	18.05.2020
<b>12</b>	19.05.2020	8.05.2020	20.05.2020
<b>13</b>	21.05.2020	13.05.2020	22.05.2020
<b>14</b>	23.05.2020	-	-

## 9. TEST ÇIKTILARI

**Tablo 3:** Yapılan testlere ait çıktıların değerlendirilmesi için referans tablo

<b>DURUM</b>	<b>AÇIKLAMA</b>
<b>Testte</b>	Testin devam ettiğini gösterir.
<b>Beklemede</b>	Testin başladığını ancak başka bir sebepten dolayı beklediğini gösterir.
<b>İptal</b>	Belli bir sebepten dolayı koşturulamayacak olan test durumunu belirtmek için kullanılır.
<b>Güncellenecek</b>	Test durumları veya senaryolarında revizyona gerekiyorsa bunu gösterir.
<b>Başarılı</b>	Koşturulan ve başarılı sonuçlanan test durumlarını veya senaryolarını belirtir.
<b>Başarısız</b>	Koşturulan ve başarısızlıkla sonuçlanan test durumlarını veya senaryolarını belirtir.

### 9.1. TEST CASE

**Tablo 4:** Test çıktıları

				TEST DURUMU					
DURUM	DURUM ADI	SONUÇ	EK AÇIKLAMA	Testte	Beklemede	İptal	Güncellenecek	Başarılı	Başarısız
1	Veriseti geliştirilecek yazılım için uygun mu?	Yazılım tasarımının isterlerini karşılayan bir veriseti olduğu doğrulanmıştır.						✓	
2	Seçilen birliktelik analizi algoritması geliştirilecek yazılım için uygun mu?	Diğer birliktelik analizi algoritmaları ile karşılaştırıldığında en yüksek güven değeri veren algoritma, Apriori Algoritması olarak belirlenmiştir.						✓	
3	Veriseti doğru okunuyor mu?	Veriseti isterler doğrultusunda okunabilmektedir.						✓	
4	Veriseti ilgisiz değerler içeriyor mu?		Yazılım testi sırasında bu duruma ihtiyaç duyulmamıştır.			✓			
5	Verisetinde null değerler bulunuyor mu?	Verisetinde rastlanan boş değerleri içeren satırlar verisetinden temizlenmiştir.					✓		
6	Veriseti yapılacak işlemler için uygun şekilde sınıflandırılmış mı?	Verisetindeki veriler müşteriler ve ürünler olarak sınıflandırıldığından yazılım tasarımına uygun olarak görülmüştür.						✓	
7	Veritabanı bağlantısı başarılı mı?	Java üzerinden Oracle veritabanına başarılı bağlantı gerçekleştirilmiştir.						✓	

8	Önerilecek ürünler ve satın alınmak istenilen ürünler birbirinden ayrı tablolarda tutuluyor mu?	Müşterilerin yaptığı alışverişler kayıt altına alınarak farklı tablolarda tutulmaktadır ve doğrulanmıştır.						✓	
9	Yeni müşteri veritabanına kayıt ediliyor mu?	Yeni müşteriye, yeni bir müşteri numarası verilerek müşteriler tablosuna kaydedilmektedir.						✓	
10	Girilen ID'ye ait bir müşteri var mı?		Bu duruma ait kontroller gereksiz görülmüş ve iptal edilmiştir.			✓			
11	Bu müşteriye öneri yapabilmek için ürün listesi dolu mu, alışveriş yapılmış mı?	Müşteriye öneri doğru bir şekilde yapıldıysa test başarıyla sonlandırılacaktır.						✓	
12	Müşterinin aldığı ürüne önerilen ürün doğru mu?	Müşterinin almış olduğu ürünün yanında daha önce, bu ürün ile birlikte aldığı bir ürün önerilmiş ise doğru bir öneri yapmış olduğu varsayılmıştır.						✓	
13	Müşterinin ilk alışverişine öneri yapılmış mı?	Müşterinin ilk alışverişi olduğunda mağazada en çok satılan ilk 5 üründen biri önerilmiştir.						✓	
14	KVKK hükümlerine aykırılık bulunuyor mu?		Hizmet verilecek kurumla anlaşma sağlandığında kontrol edilecektir.		✓				

## 9.2. BİRİM TEST

Birim testler metot, modül veya prosedür gibi bir kod parçasının kendisinden beklenen işlevselliği doğru olarak yerine getirdiğini ve hata içermediğini göstermek için gerçekleştirilen testlerdir.

Resim 3: Python Testleri

```
136
137 ▶ class MyTestCase(unittest.TestCase):
138
139     customers2 = []
140
141 ▶ def test_musteriSayisi(self):
142     customers = readObjectFromFile("customer.pkl")
143     for i in range(0, len(customers)):
144         customer: Customer = customers[i]
145
146         if len(customer.siparisList) > 9:
147             self.customers2.append(customer)
148     self.assertEqual(537, len(self.customers2))
149
150 ▶ def test_urunSayisi(self):
151     count = 0
152     for customer in self.customers2:
153         for order in customer.siparisList:
154             self.assertNotEqual(len(order.productsList), 0)
155
156
157 ▶ if __name__ == '__main__':
158     unittest.main()
```

```
..
-----
Ran 2 tests in 0.383s

OK

Process finished with exit code 0
```

Resim 4: Markette en çok satılan ürünlere ait Oracle tablosu

Columns	Data	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	De
URUN_ID	URUNLER	SIKLIK						
1	(null) TOY TIDY SPACEBOY	206						
2	(null) SUKI SHOULDER BAG	206						
3	(null) ANTIQUE GLASS DRESSING TABLE POT	206						
4	(null) SET/20 STRAWBERRY PAPER NAPKINS	205						
5	(null) SET 3 PAPER VINTAGE CHICK PAPER EGG	205						
6	(null) SET OF 5 PANCAKE DAY MAGNETS	205						
7	(null) SET 2 TEA TOWELS I LOVE LONDON	205						
8	(null) PACK OF 12 COLOURED PENCILS	204						
9	(null) JINGLE BELL HEART ANTIQUE SILVER	204						
10	(null) PARISIENNE CURIO CABINET	203						
11	(null) GIN AND TONIC MUG	203						
12	(null) COWBOYS AND INDIANS BIRTHDAY CARD	203						
13	(null) PLAYING CARDS JUBILEE UNION JACK	203						
14	(null) BULL DOG BOTTLE OPENER	203						

Columns	Data	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	De
URUN_ID	URUNLER	SIKLIK						
15	(null) RECORD FRAME 7" SINGLE SIZE	203						
16	(null) BAG 250g SWIRLY MARBLES	201						
17	(null) FAIRY CAKE FLANNEL ASSORTED COLOUR	200						
18	(null) 12 MESSAGE CARDS WITH ENVELOPES	200						
19	(null) HERB MARKER MINT	199						
20	(null) TREASURE TII BUFFALO BILL	199						
21	(null) ZINC HERB GARDEN CONTAINER	199						
22	(null) VINTAGE CHRISTMAS GIFT SACK	199						
23	(null) ABC TREASURE BOOK BOX	199						
24	(null) HERB MARKER ROSEMARY	199						
25	(null) KNEELING MAT HOUSEWORK DESIGN	198						
26	(null) SET OF 36 DOILIES PANTRY DESIGN	198						
27	(null) SMALL PURPLE BABUSHKA NOTEBOOK	198						
28	(null) FUNKY WASHING UP GLOVES ASSORTED	198						
29	(null) ENCHANTED BIRD COATHANGER 5 HOOK	197						

**Resim 5:** Müşterilerin birlikte aldığı ürünlere ait Oracle tablosu

Columns   Data   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL			
Sort..   Filter:			
ID	URUN1	URUN2	
1	1 HOME BUILDING BLOCK WORD	LOVE BUILDING BLOCK WORD	
2	1 PINK CREAM FELT CRAFT TRINKET BOX	SET OF 3 CAKE TINS PANTRY DESIGN	
3	2 ALARM CLOCK BAKELIKE PINK	ALARM CLOCK BAKELIKE GREEN	
4	2 ALARM CLOCK BAKELIKE GREEN	ALARM CLOCK BAKELIKE RED	
5	2 ALARM CLOCK BAKELIKE PINK	ALARM CLOCK BAKELIKE RED	
6	2 SET/6 RED SPOTTY PAPER CUPS	SET/20 RED RETROSPOT PAPER NAPKINS	
7	2 SET/6 RED SPOTTY PAPER PLATES	SET/20 RED RETROSPOT PAPER NAPKINS	
8	2 SET/6 RED SPOTTY PAPER CUPS	SET/6 RED SPOTTY PAPER PLATES	
9	2 ALARM CLOCK BAKELIKE GREEN	POSTAGE	
10	2 ALARM CLOCK BAKELIKE PINK	POSTAGE	
11	2 SET/6 RED SPOTTY PAPER CUPS	POSTAGE	
12	2 POSTAGE	SET/6 RED SPOTTY PAPER PLATES	
13	3 BLUE STRIPE CERAMIC DRAWER KNOB	RED STRIPE CERAMIC DRAWER KNOB	
14	4 CHARLOTTE BAG PINK POLKADOT	CHARLOTTE BAG SUKI DESIGN	
15	4 KEY FOB , SHED	KEY FOB , BACK DOOR	
16	4 RED RETROSPOT PICNIC BAG	TOILET SIGN OCCUPIED OR VACANT	
17	4 SUKI SHOULDER BAG	SKULL SHOULDER BAG	
18	5 CHRISTMAS CRAFT WHITE FAIRY	CHRISTMAS CRAFT TREE TOP ANGEL	
19	5 CHRISTMAS CRAFT TREE TOP ANGEL	GARDENERS KNEELING PAD CUP OF TEA	
20	5 ROTATING SILVER ANGELS T-LIGHT HLDR	CHRISTMAS CRAFT TREE TOP ANGEL	
21	5 CHRISTMAS CRAFT WHITE FAIRY	GARDENERS KNEELING PAD CUP OF TEA	
22	5 ROTATING SILVER ANGELS T-LIGHT HLDR	CHRISTMAS CRAFT WHITE FAIRY	
23	5 DOORMAT FANCY FONT HOME SWEET HOME	IVORY KITCHEN SCALES	
24	5 ROTATING SILVER ANGELS T-LIGHT HLDR	GARDENERS KNEELING PAD CUP OF TEA	
25	5 PAPER CHAIN KIT VINTAGE CHRISTMAS	PACK OF 6 BIRDY GIFT TAGS	
26	5 PACK OF 6 BIRDY GIFT TAGS	SET OF 20 VINTAGE CHRISTMAS NAPKINS	
27	5 PAPER CHAIN KIT VINTAGE CHRISTMAS	SET OF 20 VINTAGE CHRISTMAS NAPKINS	

## Resim 6: Java Testleri

eclipse-workspace - Apriori\_Sunum/src/apriori/Unit\_Tests.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

```
12 import org.junit.Before;
13 import org.junit.BeforeClass;
14 import org.junit.Test;
15
16 public class Unit_Tests {
17     DAO islemmler = new DAO();
18     AprioriConnection con= new AprioriConnection();
19     ArrayList<String> musteriler = new ArrayList<String>();
20     ArrayList<String> yeni = new ArrayList<String>();
21
22
23     @Test
24     public void connectionTest() {
25         assertEquals("oracle.jdbc.driver.T4CConnection@548ad73b",con.getCon());
26     }
27
28     @Test
29     public void girisTest() {
30         Musteriler musteriler =islemmler.getInstance().getMusteriler("şükriye", "1965");
31     }
32     @Test
33     public void kayitOlTest() {
34         assertEquals(true,DAO.getInstance().kayitOl("1", "1"));
35     }
36     @Test
37     public void yeniKayit_IDTest() {
38
39         assertEquals(9,DAO.getInstance().IDbul("1", "1"));
40     }
41     @Test
42     public void urun_bulTest() {
43         musteriler.add("LOVE BUILDING BLOCK WORD");
44         assertEquals(musteriler,DAO.getInstance().urunBul(1, "HOME BUILDING BLOCK WORD"));
45     }
46
47     @Test
48     public void urun_onerTest() {
49         yeni.add("TOY TIDY SPACEBOY ");
50         yeni.add("SUKI  SHOULDER BAG");
51         yeni.add("ANTIQUUE GLASS DRESSING TABLE POT");
52         yeni.add("SET/20 STRAWBERRY PAPER NAPKINS ");
53         yeni.add("SET 3 PAPER VINTAGE CHICK PAPER EGG");
54         assertEquals(yeni, DAO.getInstance().urunOner());
55     }
56 }
```

JUnit 4

Finished after 1.141 seconds

Runs: 6/6 Errors: 0 Failures: 0

apriori.Unit\_Tests (Runner: JUnit 4) (0.991 s)

girisTest (0.910 s)

urun\_bulTest (0.014 s)

urun\_onerTest (0.016 s)

kayitOlTest (0.016 s)

yeniKayit\_IDTest (0.019 s)

connectionTest (0.015 s)

Console

<terminated> Unit\_Tests [JUnit] C:\Program Files\Java\jre1.8.0\_241\bin\javaw.exe (21 Mar 2018 14:00:00)

connection is succesfull

connection is succesfull

connection is succesfull

connection is succesfull

connection is succesfull

connection is succesfull

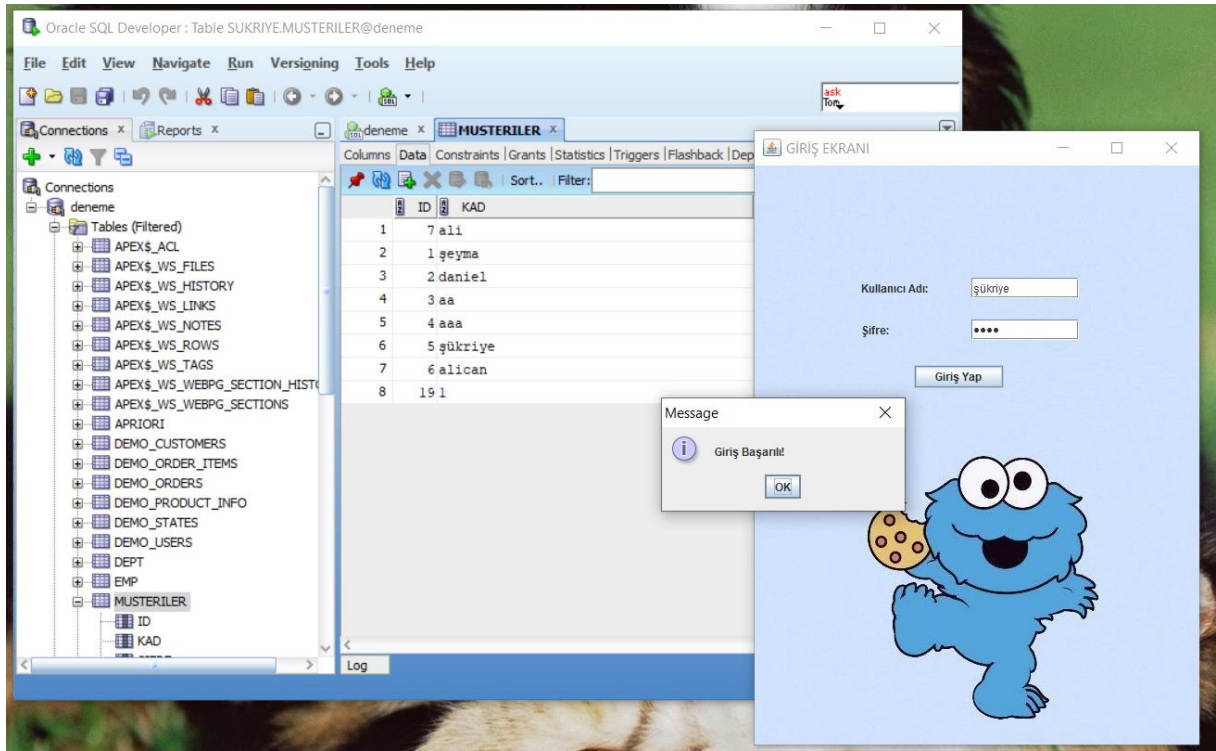
connection is succesfull

### 9.3. ENTEGRASYON TESTİ

Entegrasyon testi, yazılım modüllerinin mantıksal olarak entegre edildiği ve grup olarak test edildiği bir test türü olarak tanımlanır. Entegrasyon testinin amacı mantıksal olarak birbirine entegre edilmiş yazılım modülleri arasındaki etkileşimdeki hataları ortaya çıkarmaktır.

Projemizde artımlı yaklaşımla, mantıksal olarak ilişkili iki veya daha fazla modül birleştirilerek test edilmiştir. Ardından diğer ilgili modül eklenerek doğru çalışıp çalışmadığı test edilerek tüm modüller başarıyla birleşmesi amaçlanmıştır.

Resim 7: Oracle ve JAVA entegrasyonu





#### 9.4. SİSTEM TESTİ

Sistem testi, eksiksiz ve tam entegre yazılım ürününü doğrulayan bir test seviyesidir.

Bir sistem testinin amacı, uçtan uca sistem özelliklerini değerlendirmektir.

Bu noktada bileşenlerin birbirleriyle ve bir bütün olarak sistemle nasıl etkileşime girdiğini kontrol etmek için harici çevre birimleri de dahil olmak üzere tam entegre uygulamaların test edilmesi (end-to-end test scenario) gerçekleştirilmiştir.

**Resim 8:** Python – Oracle – JAVA entegrasyonu

