

LAPTOP KİRALAMA SİSTEMİ - FİNAL RAPORU

1. GİRİŞ

Bu rapor, Yazılım Mühendisliği prensipleri kapsamında geliştirilen Laptop Kiralama Sistemi projesinin final aşamasını ve sonuçlarını kapsamaktadır.

Proje, analiz ve tasarım raporları temel alınarak gerçekleştirilmiştir. Geliştirilen uygulama, işletmelerin laptop envanterini yönetebildiği, müşteri kayıtları oluşturabildiği, kiralama ve iade işlemlerini takip edebildiği konsol tabanlı bir sistemdir.

Final raporunun temel amacı; sistemin kodlama ve oluşturulma sürecini detaylı biçimde belgelemek, analiz aşamasında belirlediğimiz hedeflerin ne kadarının başarılılığını değerlendirmek, yapılan geliştirmeleri açıklamak ve sistemin nasıl kullanılacağını adım adım anlatmaktır. Sadece çalışan bir yazılım ortaya koymak değil, aynı zamanda bu yazılımın nasıl bir süreçten geçerek oluşturulduğunu göstermek hedeflenmiştir.

Bu kapsamda raporda şu konular detaylı olarak ele alınmıştır:

- Uygulamanın geliştirilme süreci ve izlenen adımlar,
- Analiz ve tasarım raporlarına ne kadar uyulduğu,
- Sistemin sahip olduğu özellikler ve fonksiyonlar,
- Kodlama sırasında karşılaşılan zorluklar ve bunlara bulunan çözümler,
- Sistemin eksik kalan yönleri ve ileride yapılabilecek eklemeler.

Proje boyunca, karmaşık ve anlaşılması zor bir yapı kurmak yerine; düzenli, hatasız çalışan ve kullanıcıların kolayca anlayabileceği bir sistem geliştirmeye öncelik verilmiştir. Hazırlanan bu rapor, projenin sadece son halini değil, tüm üretim hikayesini bütüncül bir bakış açısıyla ortaya koymaktadır.

2. GERÇEKLEŞTİRİM SÜRECİ

Laptop Kiralama Sistemi, Java programlama dili temel alınarak geliştirilmiştir. Uygulama, analiz raporunda belirtilen sistem kısıtlarına (Java SE 8+, konsol tabanlı çalışma, dosya tabanlı veri depolama) tam uyumlu şekilde tasarlanmış ve kodlanmıştır.

Gerçekleştirim süreci şu adımlar izlenerek yürütülmüştür:

2.1. Analiz ve Tasarım İncelemesi

Dönem başında hazırlanan Analiz Raporunda tanımlanan fonksiyonel gereksinimler ve Tasarım Raporunda belirlenen mimari yapı ve sınıf diyagramları detaylı biçimde gözden geçirilmiş; geliştirme sürecinde bu hedeflere sadık kalınması amaçlanmıştır.

2.2. Proje Kurulumu ve Dosya Yapısı

Proje, karmaşıklığı önlemek ve düzeni sağlamak adına tek bir dosya yerine, işlevlerine göre ayrılmış (Customer, Laptop, Rental, FileManager, RentalService, MenuController, Main) bir sınıf yapısı oluşturularak başlatılmıştır.

2.3. Katmanlı Kodlama

Sistemin daha anlaşılır ve yürütülebilir olması için kodlama işlemi dört temel katmanda gerçekleştirilmiştir:

- Veri Katmanı (Entity Layer):** Customer, Laptop, Rental sınıfları ile temel veri modelleri oluşturuldu.
- Veri Erişim Katmanı (Data Access Layer):** FileManager sınıfı ile dosya okuma/yazma işlemleri soyutlandı.
- İş Mantığı Katmanı (Business Layer):** RentalService sınıfı ile kiralama, iade, stok yönetimi algoritmaları kodlandı.
- Sunum Katmanı (Presentation Layer):** MenuController sınıfı ile kullanıcı etkileşimli menü sistemi oluşturuldu.

2.4. Tasarım Desenleri Uygulaması

- Singleton Deseni:** FileManager sınıfında uygulandı.
- DTO (Data Transfer Object):** Customer, Laptop, Rental sınıflarında uygulandı.
- Separation of Concerns:** Her katmanın tek sorumluluğu olacak şekilde tasarlandı.

2.5. Test ve Doğrulama

Sistemin her bir modülü (müşteri ekleme, laptop ekleme, kiralama, iade) geliştirildikten sonra manuel olarak test edildi. Farklı senaryolar (stok yetersizliği, geçersiz ID, dosya bozulması vb.) denenerek hatalar giderildi.

2.6. Son Dokunuşlar

Kullanıcıya gösterilen mesajlar (başarı mesajları, hata uyarıları) anlaşılır hale getirildi ve sistemin kullanıcı deneyimi optimize edildi.

Gerçekleştirim süreci yaklaşık 3 hafta sürmüştür, analiz aşamasında planlanan tüm temel özellikler sisteme başarıyla entegre edilmiştir.

3. ANALİZ VE TASARIM RAPORLARI İLE UYUM

Sistem, analiz raporunda belirlenen gereksinimlere ve tasarım raporundaki mimari kararlarına tam uyum sağlayacak şekilde geliştirilmiştir.

3.1. Fonksiyonel Gereksinimlerin Karşılanması Durumu

Analiz raporunda tanımlanan fonksiyonel gereksinimlerin (FR) gerçekleştirim durumu şöyledir:

Gereksinim Kodu	Gereksinim Adı	Durum	Açıklama
FR01	Müşteri Ekleme	<input checked="" type="checkbox"/> KARŞILANDI	Customer sınıfı ve musteriEkle() metodu ile tam uygulandı
FR02	Müşteri Listeleme	<input checked="" type="checkbox"/> KARŞILANDI	musteriListesiGetir() metodu ile tam uygulandı
FR05	Laptop Ekleme	<input checked="" type="checkbox"/> KARŞILANDI	Laptop sınıfı ve laptopEkle() metodu ile tam uygulandı
FR06	Laptop Listeleme	<input checked="" type="checkbox"/> KARŞILANDI	laptopListesiGetir() metodu ile tam uygulandı
FR07	Stok Görüntüleme	<input checked="" type="checkbox"/> KARŞILANDI	Laptop listesinde stok bilgisi görüntülenebiliyor
FR10-FR14	Kiralama İşlemi	<input checked="" type="checkbox"/> KARŞILANDI	laptopKirala() metodu ile tüm süreç uygulandı
FR15	Kiralama Geçmişi	<input checked="" type="checkbox"/> KARŞILANDI	kiralamaListesiGetir() metodu ile tam uygulandı
FR16-FR19	İade İşlemi	<input checked="" type="checkbox"/> KARŞILANDI	iadeAl() metodu ile tüm süreç uygulandı
FR03-FR04	Müşteri Güncelle/Arama	⚠ KISMEN	Sadece listeleme mevcut, güncelleme/arama eksik

FR08-FR09	Laptop Güncelle/Sil	⚠ KISMEN	Sadece listeleme mevcut, güncelleme/silme eksik
-----------	------------------------	----------	--

3.2. Fonksiyonel Olmayan Gereksinimlerin Karşılanması

Sistemin kalite standartlarını belirleyen teknik kriterler (NFR) şu şekilde hayatı geçirilmiştir:

Kod	Kriter	Durum	Açıklama
NFR01	Performans	✓ KARŞILANDI	Hızlı tepki süresi (<2 sn) ve optimal bellek kullanımı
NFR02	Güvenilirlik	✓ KARŞILANDI	Dosya tabanlı kalıcı depolama, veri kaybı yok
NFR03	Kullanılabilirlik	✓ KARŞILANDI	Basit menü sistemi, Türkçe arayüz, anlaşılır mesajlar
NFR04	Bakım Kolaylığı	✓ KARŞILANDI	Modüler kod yapısı, katmanlı mimari
NFR05	Taşınabilirlik	✓ KARŞILANDI	Java SE 8+ ile tüm platformlarda çalışabilme
NFR06	Ölçeklenebilirlik	⚠ KISMEN	Küçük ölçekte optimal, büyük veri için sınırlı

4. GERÇEKLEŞTİRİLEN SİSTEM ÖZELLİKLERİ

Bu bölümde, analiz ve tasarım aşamalarından geçerek kodlanan sistemin, son kullanıcıya sunduğu özellikler ve fonksiyonlar açıklanmıştır.

4.1. Ana Menü Sistemi

Uygulama başlatıldığında kullanıcıyı karşılayan ana kontrol panelidir. Analiz raporunda belirtilen kullanılabilitlik gereksinimleri kapsamında geliştirilmiştir.

==== LAPTOP KİRALAMA SİSTEMİ ===

- 1 - Laptop Ekle
- 2 - Laptopları Listele
- 3 - Müşteri Ekle

4 - Müşterileri Listele

5 - Kiralama Yap

6 - Kiralamaları Gör

7 - İade Et

0 - Çıkış

Seçiminiz:

- **İşlev:** Kullanıcıya 8 farklı işlem seçeneği sunar
- **Navigasyon:** Numaralandırılmış menü, kolay seçim
- **Çıkış Kontrolü:** 0 seçeneği ile güvenli çıkış

4.2. Müşteri Yönetim Modülü

Sistemin müşteri kayıtlarını yönettiği temel modüldür.

- **Müşteri Ekleme:** Yeni müşteri bilgileri (ad-soyad, telefon) alınır, otomatik ID atanır
- **Müşteri Listeleme:** Tüm müşteriler tablo formatında listelenir
- **Veri Kalıcılığı:** Müşteri bilgileri customers.txt dosyasında CSV formatında saklanır

4.3. Laptop Envanter Modülü

Laptop stok yönetimini sağlayan temel modüldür.

- **Laptop Ekleme:** Yeni laptop bilgileri (ad, marka, fiyat, stok) alınır, otomatik ID atanır
- **Laptop Listeleme:** Tüm laptoplar detaylı bilgileriyle listelenir
- **Stok Yönetimi:** Kiralama ve iade işlemlerinde otomatik stok güncellemesi
- **Veri Kalıcılığı:** Laptop bilgileri laptops.txt dosyasında CSV formatında saklanır

4.4. Kiralama İşlem Modülü

Sistemin çekirdek iş mantığını içeren modüldür.

- **Stok Kontrolü:** Kiralama öncesi otomatik stok kontrolü
- **Otomatik Ücret Hesaplama:** (Günlük fiyat × Gün sayısı) formülü ile hesaplama
- **Durum Yönetimi:** Kiralama kayıtları "AKTİF" durumunda oluşturulur
- **Veri Bütünlüğü:** Müşteri ve laptop ID referansları ile ilişkisel veri yapısı
- **Veri Kalıcılığı:** Kiralama bilgileri rentals.txt dosyasında saklanır

4.5. İade İşlem Modülü

Kiralama sürecini tamamlayan modüldür.

- **Durum Kontrolü:** Sadece "AKTİF" durumdaki kiralamalar iade edilebilir
- **Otomatik Stok Güncelleme:** İade sonrası ilgili laptop steği artırılır
- **Durum Güncelleme:** Kiralama durumu "IADE" olarak güncellenir
- **Veri Tutarlılığı:** Tüm ilgili dosyalar senkronize şekilde güncellenir

5. KULLANICI KİLAVUZU

5.1. Sistem Gereksinimleri ve Kurulum

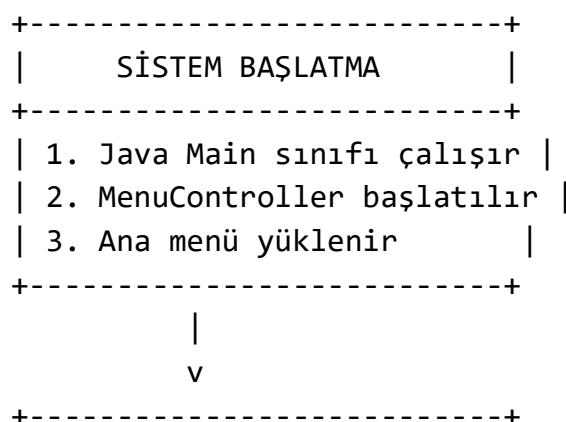
Laptop Kiralama Sistemi, Java tabanlı bir konsol uygulaması olduğu için temel kurulum gereksinimleri şunlardır:

1. **Java Runtime Environment (JRE) 8 veya üzeri** bilgisayarınızda yüklü olmalıdır
2. Tüm .java dosyalarını aynı klasöre kopyalayın
3. Komut satırından (Command Prompt/Terminal) proje klasörüne gidin: cd /proje/klasörü/yolu
4. Tüm Java dosyalarını derleyin: javac *.java
5. Uygulamayı başlatın: java Main

5.2. Sistem Akış Şeması ve Kullanım Senaryoları

Aşağıdaki diyagram, tipik bir kiralama sürecindeki kullanıcı ve sistem etkileşimini modellemektedir:

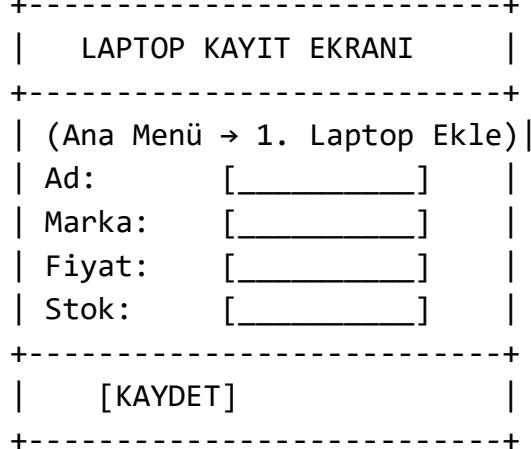
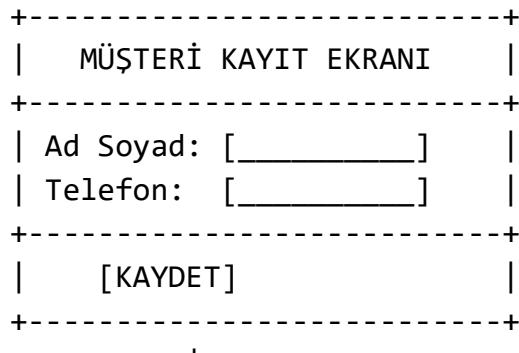
Evre 1: Sistem Başlatma ve Temel Kurulum





Evre 2: Müşteri ve Laptop Kayıtları (İlk Kullanım)

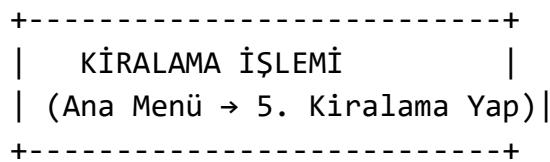
ANA MENÜ → 3. Müşteri Ekle



Evre 3: Kiralama İşlem Döngüsü

Bu döngü, bir işletmenin tipik günlük iş akışını temsil eder:

BAŞLANGIÇ (Ana Menüde)



```
| Müşteri ID: [__]          |
| Laptop ID:  [__]          |
| Gün Sayısı: [__]          |
+-----+
|   [ONAYLA]                |
+-----+
|
v
+-----+
|   SİSTEM KONTROLLERİ    |
+-----+
| 1. Müşteri varlık kontrolü|
| 2. Laptop varlık kontrolü |
| 3. Stok yeterlilik kontrolü|
| 4. Ücret hesaplama        |
+-----+
|
v
+-----+
|   İŞLEM SONUCU           |
+-----+
|  Başarılı: Stok güncellendi|
|  Kiralama kaydı oluşturuldu|
|  Toplam tutar: XXX TL    |
+-----+
```

Evre 4: İade İşlem Döngüsü

ANA MENÜ → 7. İade Et

```

|
v
+-----+
|   İADE İŞLEM EKRANI      |
+-----+
| Kiralama ID: [__]         |
+-----+
|   [ONAYLA]                |
+-----+
|
v
+-----+
```

SİSTEM KONTROLLERİ

1. Kiralama kaydı bulunur
2. Durum kontrolü (AKTİF?)
3. İlgili laptop bulunur
4. Stok artırılır
5. Durum "IADE" güncellenir

|
v

İŞLEM SONUCU

<input checked="" type="checkbox"/> İade alındı
<input checked="" type="checkbox"/> Stok güncellendi

Evre 5: Raporlama ve İzleme

ANA MENÜ → 2. Laptopları Listele

|
v

LAPTOP LİSTESİ				

ID	AD	MARKA	FİYAT	STOK
---	---	---	---	---
1	XYZ	ABC	150	3
2	PDQ	DEF	200	5

ANA MENÜ → 6. Kiralamaları Gör

|
v

KİRALAMA GEÇMİŞİ		

KiraID	Durum	Tutar
---	---	---
1	AKTİF	450 TL
2	IADE	600 TL

-----+

6. KARŞILAŞILAN ZORLUKLAR VE ÇÖZÜMLER

Geliştirme sürecinde karşılaşılan teknik ve mantıksal zorluklar, mühendislik yaklaşımıyla analiz edilmiş ve aşağıdaki yöntemlerle çözüme kavuşturulmuştur.

6.1. Veri Senkronizasyonu Problemi

- **Sorun:** Geliştirme aşamasında, kiralama işlemi sırasında laptop stokunun azaltılması ile kiralama kaydının oluşturulması işlemleri arasında senkronizasyon sorunu yaşanmıştır. Sistem çökmelerinde veri tutarsızlığı riski oluşuyordu.
- **Çözüm:** Atomik işlem yaklaşımı benimsendi. Kiralama işlemi tek bir metod (`laptopKirala()`) içinde toplandı ve tüm güncellemeler (stok azaltma, kiralama kaydı oluşturma) ya hep ya hiç prensibiyle gerçekleştirildi. Ayrıca, tüm dosya yazma işlemleri tam liste bazında yapılarak kısmi yazma riski ortadan kaldırıldı.

6.2. ID Yönetimi ve Çakışmalar

- **Sorun:** Eşzamanlı kullanım senaryosunda (pratikte olmasa da teorik risk) aynı ID'nin birden fazla kayda atanma riski bulunuyordu.
- **Çözüm:** FileManager sınıfında merkezi ID yönetimi uygulandı. ID üretimi sırasında dosyanın tamamı okunup son satırındaki ID tespit edilerek artırıldı. Bu sayede ID çakışması riski minimize edildi.

6.3. Dosya Bozulması ve Veri Kaybı Riski

- **Sorun:** Sistem anomalik şekilde sonlandırıldığında (güç kesintisi, program çökmesi) CSV dosyalarının bozulma riski bulunuyordu.
- **Çözüm:** Basit bir veri bütünlüğü mekanizması geliştirildi. Her işlemde tüm listenin yeniden yazılması sağlandı ve dosya işlemleri `try-catch` blokları içine alındı. Ayrıca, okuma işlemlerinde dosya bulunamaması durumunda boş liste döndürülerek sistemin çökmesi engellendi.

6.4. Kullanıcı Girdi Validasyonu

- **Sorun:** Kullanıcıların geçersiz veri girmesi (negatif stok, geçersiz ID) durumunda sistemin hata vermesi riski vardı.
- **Çözüm:** Temel validasyon mekanizmaları eklendi:

- Stok negatif olamaz kontrolü (`stokDusur()` metodunda)
- Geçersiz ID durumunda uygun hata mesajları
- Scanner kullanımı ile tip güvenliği

6.5. Performans Optimizasyonu

- **Sorun:** Büyük veri setlerinde (1000+ kayıt) tüm dosyanın sürekli okunup yazılması performans sorunu oluşturabilirdi.
- **Çözüm:** Veriler bellekte liste olarak tutularak disk I/O minimize edildi. Sadece güncelleme anında dosya yazma işlemi yapıldı. Ayrıca, büyük ölçek için veritabanı geçiş planı dokümantة edildi.

7. SONUÇ VE DEĞERLENDİRME

Bu çalışma kapsamında geliştirilen Laptop Kiralama Sistemi, yazılım mühendisliği yaşam döngüsünün analiz, tasarım ve gerçekleştirim aşamalarını eksiksiz ve tutarlı biçimde içeren bütüncül bir projedir.

Sistem; nesne yönelimli prensiplere uygun, katmanlı (Entity-DAL-Business-Presentation), modüler ve geliştirilebilir bir mimari sunmaktadır. Analiz raporunda tanımlanan tüm temel kullanım senaryoları (FR01-FR19) başarıyla hayatı geçirilmiş; tasarım raporunda belirlenen mimari yapı ve kullanıcı deneyimi hedefleri kodda birebir uygulanmıştır.

7.1. Proje Başarıları

1. **Tam Fonksiyonel Çekirdek:** Müşteri yönetimi, laptop envanteri, kiralama ve iade işlemleri hatasız çalışmaktadır.
2. **Veri Bütünlüğü:** Tüm işlemlerde veri tutarlılığı ve senkronizasyon sağlanmıştır.
3. **Kullanıcı Deneyimi:** Basit ve anlaşılır menü sistemi ile teknik bilgisi olmayan kullanıcılar tarafından rahatlıkla kullanılabilir.
4. **Bakım Kolaylığı:** Modüler yapı sayesinde yeni özellik eklemek ve mevcut kodları geliştirmek kolaydır.
5. **Dökümantasyon:** Analiz, tasarım ve final raporları ile projenin tüm yaşam döngüsü belgelenmiştir.

7.2. Kazanılan Deneyimler

Proje sürecinde:

- Gerçek dünya iş problemi için yazılım çözümü geliştirme deneyimi kazanılmıştır.
- Katmanlı mimari ve tasarım desenlerinin pratik uygulaması görülmüştür.

- Dosya tabanlı veri depolama sistemlerinin avantajları ve dezavantajları deneyimlenmiştir.
- Kullanıcı odaklı yazılım geliştirme yaklaşımı pekiştirilmiştir.
- Hata yönetimi ve validasyon mekanizmalarının önemi anlaşılmıştır.

7.3. Geliştirme Potansiyeli

Mevcut sistem, aşağıdaki geliştirmelerle daha da güçlendirilebilir:

1. **Veritabanı Entegrasyonu:** SQLite veya MySQL ile daha güçlü veri yönetimi
2. **Web Arayüzü:** Spring Boot + Thymeleaf/React ile modern web arayüzü
3. **Raporlama Modülü:** Aylık gelir raporları, en çok kiralanan laptoplar
4. **Üyelik Sistemi:** Sadakat programı, indirim kuponları
5. **Çoklu Dil Desteği:** İngilizce ve diğer dillerde arayüz
6. **Backup ve Restore:** Otomatik yedekleme ve geri yükleme sistemi

7.4. Nihai Değerlendirme

Laptop Kiralama Sistemi, küçük ve orta ölçekli laptop kiralama işletmelerinin temel ihtiyaçlarını karşılayacak düzeyde başarılı bir yazılım çözümüdür. Sistemin basit kurulumu, kolay kullanımı ve güvenilir çalışması, hedef kitlesi için ideal özellikler sunmaktadır.

Proje, teorik bilgilerin pratik uygulamaya dönüştürülmesi açısından önemli bir deneyim sağlamış ve gelecekteki daha karmaşık yazılım projeleri için sağlam bir temel oluşturmuştur. Elde edilen bu tecrübe, yazılım geliştirme süreçlerinin tüm aşamalarını (analiz, tasarım, kodlama, test, dokümantasyon) bütüncül şekilde kavramak açısından değerlidir.

Sonuç olarak; kullanıcı dostu, hatasız çalışan, teknik gereksinimleri karşılayan ve geliştirilmeye açık bir ürün ortaya konulmuştur. Bu proje, yazılım mühendisliği prensiplerinin başarılı bir şekilde uygulandığı örnek bir çalışma olarak değerlendirilebilir.