



İSTANBUL TEKNİK ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü - Bilgisayar Mühendisliği

YAZILIM MODELLEME VE TASARIMI DÖNEM PROJESİ

*HAVAYOLU FİRMALARI İÇİN
REZERVASYON VE BİLET SATIŞ SİSTEMİ*

Hazırlayan : Beycan Kahraman
Numara : 504071508
Öğr. Üyesi : Yrd. Doç. Dr. Feza Buzluca

24 Aralık 2008

İçerik

Şekiller	2
A. Ön Hazırlık	3
B. Kullanım Senaryoları.....	3
1. Aktörlerin Belirlenmesi.....	3
2. Senaryo Grupları (Use-Case)	4
a. SG1: Rezervasyon Yapma	5
b. SG2: Rezervasyon İptali	6
3. Sözleşmeler (Contracts).....	7
a. Sözleşme 1: Sistem ceza kesintisini hesaplar	7
b. Sözleşme 2: Sistem yapılacak indirimi hesaplar	7
C. Uygulama Domeni Modelinin Oluşturulması ve UML Sınıf Diyagramı.....	7
1. Kavramsal Sınıfların Belirlenmesi	7
2. Kavramsal Sınıflar Arası Bağlantıların Belirlenmesi	8
D. Nesnel Tasarım Modelinin Oluşturulması	10
• Nesnel Tasarım Modelinin Oluşturulması, UML Sınıf ve Etkileşim Diyagramları	10
a. Genel Tasarımın belirlenmesi.....	10
b. İndirim ve Ceza ile İlgili Güncelleme.....	14
c. Değişken Kısımların Güncellenmesi.....	15
E. Sonuçlar ve Eleştiriler	19

Şekiller

Şekil 1: Senaryo Grupları	4
Şekil 2: Kullanıcı, Rezervasyon, Koltuk Sınıfları	8
Şekil 3: Kullanıcı, Rezervasyon, Koltuk, Uçuş ve Kayıt Defteri Sınıfları.....	8
Şekil 4: UML Sınıf Diyagramı	9
Şekil 5: Nesnel Tasarım Modeline Giriş.....	10
Şekil 6: UML - Tasarımın Başlangıcı.....	11
Şekil 7: Kullanıcı Girişi.....	11
Şekil 8: Uçuş Taraması.....	11
Şekil 9: Koltuk Listesi	12
Şekil 10: Genel UML Sınıf Tasarımı.....	12
Şekil 11: Rezervasyon ve Ödeme	13
Şekil 12: Kayıt ve Güncelleme	13
Şekil 13: Rezervasyonun İptali.....	13
Şekil 14: Genel UML Sınıf Diyagramının Son Hali.....	14
Şekil 15: Değişken Uçuş Sınıflarının Tasarımı	15
Şekil 16: Ardışıl Etkileşim Diyagramı – Ödeme.....	15
Şekil 17: Ardışıl Etkileşim Diyagramı - Geri İade.....	16
Şekil 18: İndirim ve Ceza Stratejileri, Tekil Fabrika Tasarımı.....	17
Şekil 19: Ayrıntılı UML Sınıf Diyagramının Son Hali.....	18

A. Ön Hazırlık

Havayolu firmaları için rezervasyon ve bilet satış sisteminin hazırlanması için gelen müşteri, bizden aşağıdaki isteklerde bulunmaktadır.

- Kullanıcı güzergah ve tarih bilgisini vererek ilgili seferleri görebilir.
- Herhangi bir seferi seçtiğinde, bu uçuştaki koltukların durumlarını görebilir (boş / dolu).
- Koltuk numarası vererek rezervasyon yapabilir.
- Müşteri rezervasyonu iptal edebilir.
- Rezervasyon iptalinde çeşitli cezalar kesilebilir (DEĞİŞEBİLİR).
- Sistemde çeşitli uçuş tipleri mevcut (DEĞİŞEBİLİR).
- Sistemde çeşitli indirimler mevcut (DEĞİŞEBİLİR).
- Tüm rezervasyon ve iptal işlemlerinin kayıtları sistemde tutulur.

Bu genel isteklerden yararlanarak projenin yazılım modellenmesi ve tasarımı aşağıdaki aşamalarla açıklanmalıdır.

- İsteklerin (requirements) belirlenmesi, kullanım senaryolarının (use-case) yazılması
- Analiz: Uygulama domeninde modelleme
- Tasarım: Yazılım domeninde modelleme
 - Etkileşim diyagramları
 - Sınıf diyagramları
- Kodlama (Projenin kapsamında olmadığından yapılmayacaktır)
- Sınama (Projenin kapsamında olmadığından yapılmayacaktır)

Öncelikle, isteklerin ayrıntılı olarak belirlenmesi ve kullanım senaryolarının yazılmasıyla modellemeye başlayabiliriz.

B. Kullanım Senaryoları

Modellenecek olan sistemle ilgili genel bir bilgi elde ettikten sonra, müşterinin istekleri doğrultusunda ayrıntılı olarak kullanım senaryoları oluşturulacaktır. Böylece isteklerin anlaşılması ve ifade edilmesi sağlanacaktır.

Kullanım senaryoları, sistemle aktörler arasında gerçekleşen olaylar zinciridir. Bu sebeple öncelikle aktörlerin belirlenmesi, senaryoların hazırlanmasını kolaylaştıracaktır.

1. Aktörlerin Belirlenmesi

Projedeki istekleri incelendiğinde, ilk aşamada dört önemli aktör hemen göze çarpmaktadır:

- Müşteri: Sistemden yararlanacak olan kullanıcılar.
- Kredi kartı işlem onay merkezi: Ödemelerin sadece kredi kartıyla yapıldıkları varsayımıyla, ödeme ve iptal işlemlerinin sonucunu veren aktör.
- Veri tabanı: Yapılan tüm işlemlerin kayıtlarını tutan aktör.
- Yönetici: Sistemdeki ceza ve indirimleri değiştirme hakkına sahip kullanıcı.

Sistemin karmaşıklığını daha fazla arttırmamak ve projenin temel isteklerinin karşılanmasının yeterli olması sebebiyle için, bu dört aktörle yetinilmiştir. Vergi dairesi, müşteri temsilcileri, ... gibi aktörler gözardı edilmişlerdir.

2. Senaryo Grupları (Use-Case)

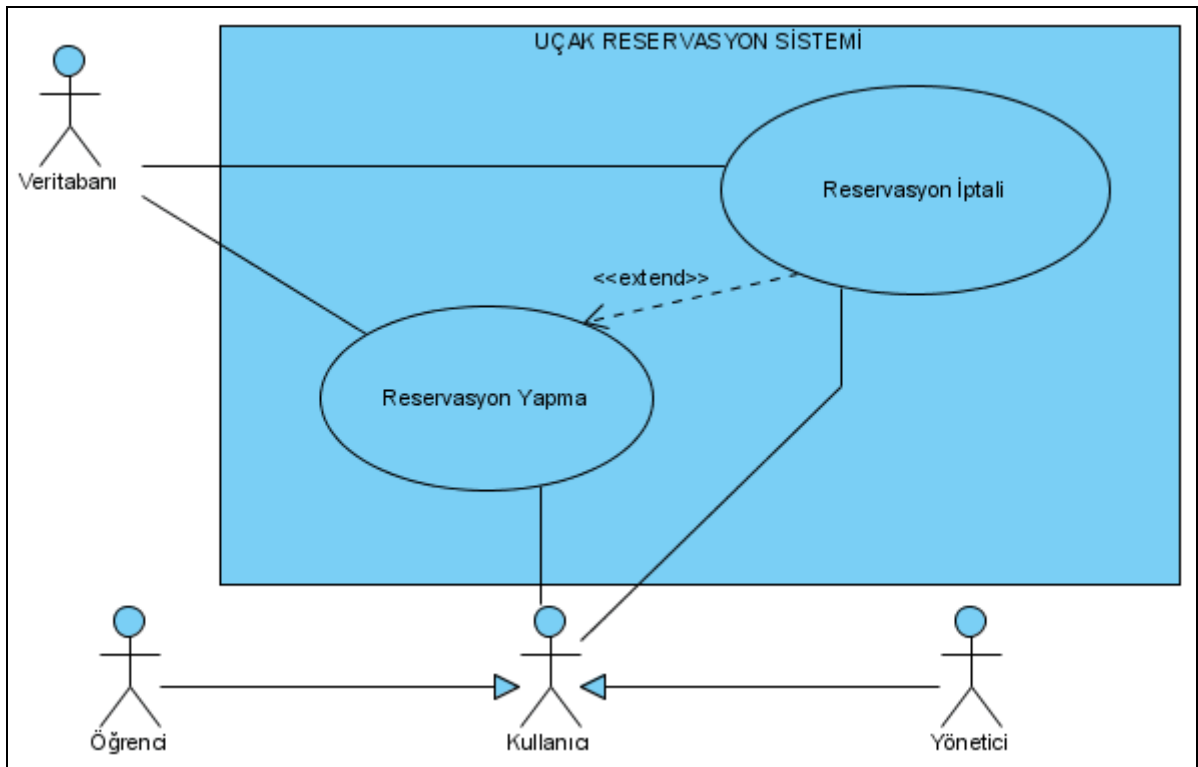
Sistemde yer alacak aktörler belirlendikten sonra, kullanım senaryolarını oluşturmaya geçebiliriz. Öncelikle, senaryo gruplarını belirlemeye çalışalım.

- Kullanıcının rezervasyon yapması
- Kullanıcının rezervasyonunu iptal etmesi

Tasarlanacak olan sistemi yukarıda belirtilen iki temel senaryoya göre modelleyeceğiz. Veri tabanımızın oldukça basit olacağı varsayıldığından, kaydetme işlemi ile ilgili senaryo yazılmayacaktır. Ayrıca, kredi kartı onay merkezinin de onayı verdiği varsayılacaktır. Aşağıda verilen diğer senaryo gruplarıyla ilgilenilmeyecektir.

- Yöneticinin indirim seçeneklerini değiştirmesi
- Yöneticinin ceza kesintilerini değiştirmesi
- Rezervasyon veya iptal işlemlerinin veritabanına kaydedilmesi
- Kredi kartı merkezinden satış onayının alınması
- Kredi kartı merkezinden iptal onayının alınması
- ...

Bu durumda elde edilen senaryo grupları aşağıdaki şekilde verilmiştir.



Şekil 1: Senaryo Grupları

a. SG1: Rezervasyon Yapma

Konu: Havayolu firmaları için rezervasyon ve bilet satış sistemi

Birincil aktör: Müşteri

İlgililer ve beklentileri:

Müşteri: Doğru ve hızlı bir şekilde rezervasyon işlemi yapabilmek

Veritabanı: Kaydedilecek bilgilerin bilinen bir yapıda gelmesi

Kredi kartı merkezi: İstenen işlemin doğru formatta bildirilmesi

Ön koşullar: (Sisteme giriş kullanılmayacaktır : İDO'da olduğu gibi giriş yapılmadan işlem no yardımıyla bütün işlemler takip edilecektir, kullanıcı iptal etmek için işlem numarasını hatırlamalı).

Son koşullar: Kredi kartı asıllama merkezinden onay alınmıştır, yapılan rezervasyon kaydedilmiştir, koltuk bilgileri güncellenmiştir.

Ana Başarılı Senaryo:

1. Müşteri tarih ve güzergah bilgisini seçer ve arama tuşuna basar
2. Sistem uygun seferlerin listesini getirir
3. Müşteri isteğine uygun seferi seçer
4. Sistem koltuk listesini getirir
5. Müşteri uygun olan boş koltukları seçer ve onaylar
6. Sistem kullanıcıdan kredi kartı bilgilerini ister
7. Müşteri bilgilerini sisteme girer
8. Sistem yapılacak indirim hesaplar
9. Sistem kredi kartı onay merkezine bilgileri gönderir
10. Kredi kartı onay merkezi bilgileri onaylar
11. Sistem, rezervasyon bilgisini veritabanına kaydeder
12. Müşteri, başarılı işlem konusunda bilgilendirilir ve müşteriye işlem numarası verilir
13. Müşteri sistemden ayrılır

Uzantılar:

2. Uygun sefer bulunamamıştır. Seçim sayfasına geri dönülür ve müşteri bilgilendirilir.
5. Boş koltuk yoktur.
 1. Sistem kullanıcıya iki seçenek sunar, ana seçim sayfası ya da arama sonuçlarına geri dön
 2. Müşteri boş koltuk kalmadığını anlar ve tepki verir.
 - 2a. Müşteri ana sayfaya gider, yeni bir arama başlatır --> 1
 - 2b. Müşteri arama sonuçlarına geri döner --> 3
 - 2ba. Uygun başka bir sefer vardır, onu seçer --> 4
 - 2bb. Uygun başka sefer yoktur --> 1
- 10a. Kredi kartı onay merkezine erişilemez
Sistem, müşteriye bilgilendirir ve işlem sonlandırılır
- 10b. Kredi kartı onay merkezi bilgilerin yanlış olduğunu bildirir
 1. Kullanıcı bilgilerinin yanlış olduğu konusunda bilgilendirilir
 2. Kullanıcıdan bilgilerini tekrar girmesi istenir
 - 2a. Kullanıcı sistemi terk eder
 - 2b. Kullanıcı bilgilerini tekrar girer --> 8

Özel İstekler:

Kredi kartı onay merkezinden cevap 30 saniye içinde gelmeli
Sistemde kullanılan renkler, sarı lacivert olmalı :)

Açık Noktalar:

Müşteri işlem numarasını unuttuysa, kredi kartıyla bulunabilsin mi?
Rezerve edilmiş koltuğun, bileti en geç ne zaman alınmalıdır ki rezervasyon düşmesin?

b. SG2: Rezervasyon İptali

Konu: Havayolu firmaları için rezervasyon ve bilet satış sistemi

Birincil aktör: Müşteri

İlgililer ve beklentileri:

Müşteri: Doğru ve hızlı bir şekilde önceden yaptığı rezervasyon işlemini iptal edebilmek

Veritabanı: Kaydedilecek bilgilerin bilinen bir yapıda gelmesi

Kredi kartı merkezi: İstenen işlemin doğru formatta bildirilmesi

Ön koşullar: Müşterinin rezerve ettiği koltuk vardır ve işlem numarasını bilmektedir.

Son koşullar: Kredi kartı asıllama merkezinden onay alınmıştır, rezervasyon iptal edilmiştir, koltuk bilgileri güncellenmiştir.

Ana Başarılı Senaryo:

1. Müşteri rezervasyon iptal kısmına giriş yapar
2. Müşteri işlem numarasını ve kredi kart numarasını girer (koruma amaçlı)
3. Sistem veritabanından gerekli işlemi ister
4. Veritabanı işlemi bulur
5. Sistem, kredi kartı numaralarını karşılaştırır ve numaralar uyuşur
6. Sistem ceza kesintisini hesaplar
7. Sistem kredi kartı onay merkezine bilgileri gönderir
8. Kredi kartı onay merkezi bilgileri onaylar
9. Sistem, rezervasyon bilgisini veritabanına kaydeder
10. Koltuk bilgileri güncellenir
11. Müşteri, başarılı işlem konusunda bilgilendirilir
12. Müşteri sistemden ayrılır

Uzantılar:

4. Veritabanı işlemi bulamaz
Kullanıcı bilgilendirilir ve işlem iptal edilir
5. Kredi kartı numaraları uyuşmaz
Kullanıcı bilgilendirilir ve işlem iptal edilir
- 9a. Kredi kartı onay merkezine erişilemez
Sistem, müşteriye bilgilendirir ve işlem sonlandırılır

Özel İstekler:

Kredi kartı onay merkezinden cevap 30 saniye içinde gelmeli

Açık Noktalar:

Müşteri işlem numarasını unuttuysa, kredi kartıyla bulunabilsin mi?

Kredi kartına para geri gönderildiği anda sistemde hata olur ve kapanırsa, koltuk bilgileri nasıl güncellenecek?

3. Sözleşmeler (Contracts)

Kullanıcı senaryoları yardımıyla sistemin büyük bir kısmı anlaşılmış ve üzerinde çalışılacak proje hakkında ayrıntılı bilgi elde edilmiştir. Yine de bazı maddelerin yeterince açık olmadığı düşünülerek, iki tanesi için sözleşme yazılmasının gerektiği varsayılacaktır.

a. Sözleşme 1: Sistem ceza kesintisini hesaplar

Operasyon: CezaKesintisi()

Karşı Referans: Kullanım Senaryoları: Rezervasyonun İptali

Ön Koşullar: Önceden belirlenmiş bir ceza prosedürü mevcuttur

Son Koşullar: Ceza prosedüründen en uygun olan seçilmiştir. Bu ceza oranı müşterinin kredi kartına yapılacak iadedden düşülmüştür.

b. Sözleşme 2: Sistem yapılacak indirim hesaplar

Operasyon: IndirimHesapla()

Karşı Referans: Kullanım Senaryoları: Rezervasyon Yapma

Ön Koşullar: Önceden belirlenmiş bir indirim prosedürü mevcuttur

Son Koşullar: İndirim prosedüründen en uygun olan seçilmiştir. Bu indirim oranı müşterinin kredi kartıyla yapacağı ödemeden düşülmüştür.

C. Uygulama Domeni Modelinin Oluşturulması ve UML Sınıf Diyagramı

Projenin karmaşıklığını azaltıp, sistemin nesnel mimarisi üzerinde yoğunlaşılması için sistemde birçok durum açık bırakılmıştır. Örneğin; yöneticinin, indirimleri ve ceza oranlarını değiştirmesi düşünülmeyecektir. Bunun sonucunda, sadece rezervasyonun yapılması ve rezervasyon iptaliyle ilgili kullanım senaryoları üzerinden analiz yapılacaktır.

1. Kavramsal Sınıfların Belirlenmesi

Öncelikle, kavramsal sınıfların neler olabileceğini belirlemeye çalışalım. Verilen isteklerden de yararlanarak akla gelen temel sınıflar aşağıda sıralanmıştır:

UÇUŞ TİPİ, KULLANICI, UÇAK, KOLTUK, REZERVASYON, BİLET, İNDİRİM, CEZA, KAYIT DEFTERİ, KAYIT.

Gereksiz sınıfları elemeye çalışarak, projenin daha sade bir analiz modelini oluşturmaya çalışalım. İlk olarak, rezervasyon yapmanın bilet almakla eşdeğer olduğu düşünülürse BİLET sınıfını yukarıdaki listeden çıkarabiliriz. Doğal sonucu olarak, rezervasyon iptali de bileti geri vermeye eşdeğer olarak kabul edilmiştir.

Uçak ile ilgili özel bilgiler bizi ilgilendirmediğinden, UÇUŞ TİPİ ve UÇAK sınıflarını tek bir UÇUŞ sınıfı altında toplarsak güzel bir sadeleştirme yapmış oluruz. Son durumda elde edilen kavramsal sınıflar:

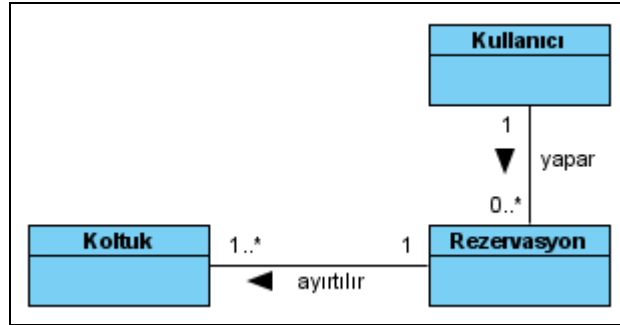
UÇUŞ, KULLANICI, KOLTUK, REZERVASYON, İNDİRİM, CEZA, KAYIT DEFTERİ ve KAYIT'tır.

Betimleme sınıfları ile ilgili başlangıç aşamasında herhangi bir kestirme yapılamamıştır. Analiz aşamasında gerekli olduğu takdirde eklemeler yapılacaktır.

2. Kavramsal Sınıflar Arası Bağlantıların Belirlenmesi

En temel kısımlarıyla kavramsal sınıflar arasındaki bağlantıları belirlemeye çalışalım.

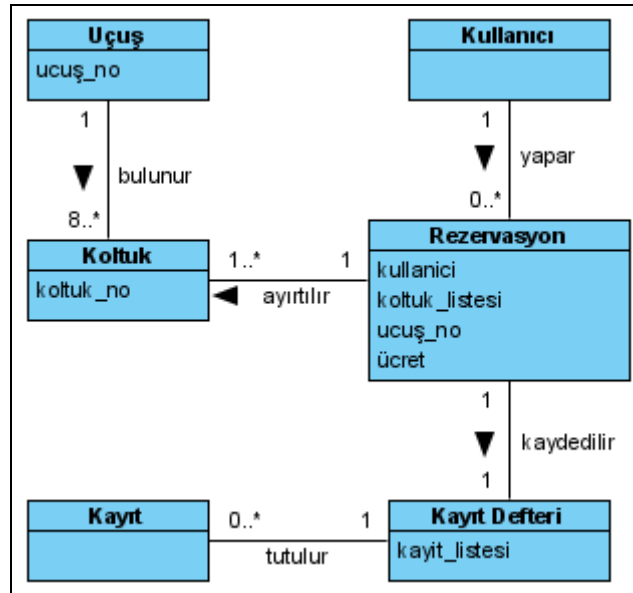
İlk aşamada kullanıcıların rezervasyon yaptığı bilindiğinden bu işlem sınıf listesine eklenecektir. Bununla birlikte, her rezervasyonda bazı koltukların ayrıldığı bilindiğinden bu bağlantı da sınıf listesine eklenebilir.



Şekil 2: Kullanıcı, Rezervasyon, Koltuk Sınıfları

Gelinen aşamada, kayıt defteri ve uçuş nesnelerini sınıf diyagramına nasıl belirleneceği üzerinde durulacaktır.

Her uçuşta koltukların bulunduğu düşünülürse, uçuşu koltuğa bağlayabiliriz. Kayıt defteri ise yapılan rezervasyonları ve iptal işlemlerinin kaydı için kullanılacaktır. Bu durumda, kayıt defterini rezervasyona bağlamamız uygun olacaktır.



Şekil 3: Kullanıcı, Rezervasyon, Koltuk, Uçuş ve Kayıt Defteri Sınıfları

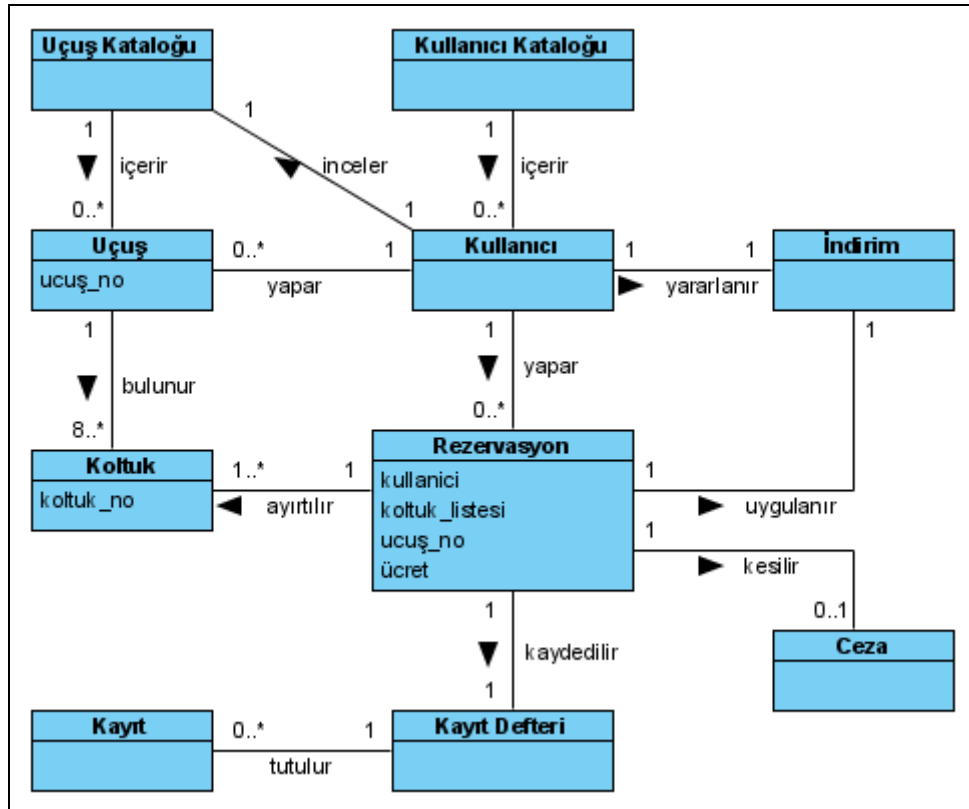
Böylece, hazırlanan tasarımda temel 6 kavramsal sınıf arası bağlantılar gerçekleşmiştir. Son olarak karar verilmesi gereken iki sınıf kalmıştır: İndirim ve Ceza sınıfları. Analizin bu kısmında biraz tasarımsal birikimlerimizden de yararlanacağız.

Uçuş tiplerine, kullanıcı çeşitlerine, kişinin önceden yaptığı uçuş bilgilerine ve belli saat aralıklarına göre farklı indirimler uygulanabilmektedir. Bu durumda indirim sınıfı; uçuş, rezervasyon, kayıt defteri ve kullanıcı kavramsal bilgilerinden her biriyle ilişkilendirilebilir. Bu seçimdeki temel kriterimiz karmaşıklığı azaltmaya çalışmak olacaktır. Rezervasyon sınıfının tüm bu bilgileri içereceği düşünüldüğünden indirim sınıfının bu sınıfa bağlanması uygun görülmektedir. Benzer şekilde, ceza sınıfı da rezervasyon sınıfına bağlanmalıdır.

Elde edilen son UML sınıf diyagramında, bazı sınıflarla ilgili gerekli olacağı düşünülen nitelikler de eklenmiştir. Bu aşamada bazı betimleme sınıflarının gerekli olduğu görülmektedir. Örneğin tüm uçuşların ve tüm kullanıcıların bulunduğu birer kataloğa ihtiyaç duyulmaktadır. Bu durumda, iki kavramsal sınıf daha eklenmelidir. Bu durumda eklenecek kataloglar aşağıda verilmiştir:

UÇUŞ KATALOĞU ve KULLANICI KATALOĞU.

Bu iki sınıfı da aşağıdaki şekilde gösterildiği gibi UML sınıf diyagramına ekleyebiliriz.



Şekil 4: UML Sınıf Diyagramı

Uçuş ve kullanıcı kataloglarının sistem çalıştırıldığında doldurulduğu varsayılmaktadır. Ayrıca zamanı geldiğinde, eklemelerin otomatik olarak yapıldıkları varsayılacaktır. Bir sonraki adımda artık tasarım aşamalarına geçebiliriz.

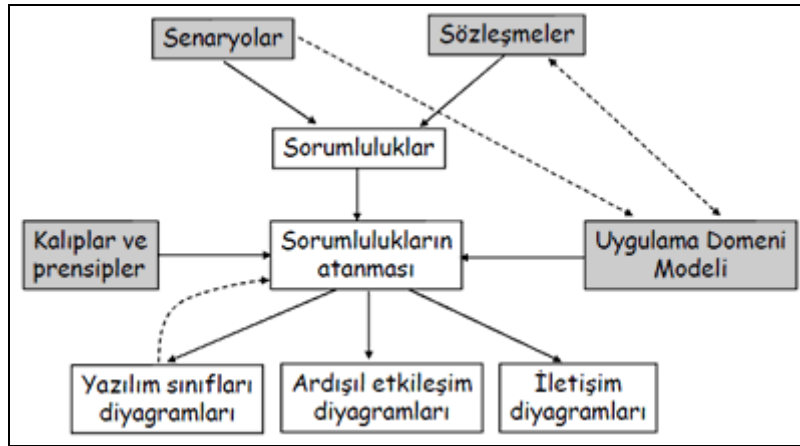
Sistemin analizi aşamasında değişebilir olarak öngörülen uçuş, indirim ve ceza ile ilgili bilgiler mevcuttur. Başlangıçta, sistemin genel davranışı ifade edecek bir tasarım yapılması hedeflenmektedir. Hemen ardından, değişebilecek kısımların kalıplar dahilinde sisteme en uygun şekilde nasıl monte edilebileceği incelenecektir.

D. Nesnel Tasarım Modelinin Oluşturulması

Uygulama domeni modeli ve kavramsal sınıflar için hazırlanan UML sınıf diyagramı elimizde olduğuna göre, artık nesneye dayalı yöntemle göre problemlerin mantıksal çözümlerini bulmaya geçebiliriz. Hazırlanacak olan tasarım modelinde, yazılım sınıfları ve aralarındaki işbirlikleri belirlenecektir.

Yukarıda anlatılanlar gerçekleşirken, sınıflar arası sorumluluklar paylaşılacaktır. Bu aşamada, GRASP kalıpları (bilginin uzmanı, yaratıcı, az bağımlılık, denetçi, iyi uyum, çok şekillilik, yapay sınıf, dolaylılık, değişimlerden korunma) ve GoF (adaptör, fabrika, tekil nesne, strateji, bileşik nesne, ön yüz, gözlemci, köprü, dekoratör, ...) kalıplarından yararlanılacaktır.

Gelinen son aşamada, aşağıdaki şekilde gri renkle belirtilenler (senaryolar, sözleşmeler, uygulama domeni modeli, kalıplar ve prensipler) elimizde olduğundan, öncelikle senaryolar ve sözleşmelerden yararlanılarak sorumluluklar belirlenecektir. Ardından; uygulama domeni modeli, kalıplar ve prensiplerden yararlanılarak sorumluluklar atanacaktır. Atanan bu sorumluluklar da yazılım sınıfları, ardışıl etkileşim diyagramları ve iletişim diyagramları yardımıyla açıklanacaktır.



Şekil 5: Nesnel Tasarım Modeline Giriş

• Nesnel Tasarım Modelinin Oluşturulması, UML Sınıf ve Etkileşim Diyagramları

Uçuş ve kullanıcı kataloglarının önceden doldurulmuş olduklarını varsaymaktayız. Bu durumda, müşterilerin inceleyeceği uçuşlar ve koltukların durumları da sistemde yer almaktadır. Bu bilgiler elimizde olduğuna göre, analiz aşamasında elde ettiğimiz kavramsal sınıf diyagramlarını tasarım modelinde de kullanabiliriz. Benzer şekilde, proje kapsamında kullanıcı kataloğu ve kullanıcıların da önceden sistemde oluşturulduğu varsayılmıştır.

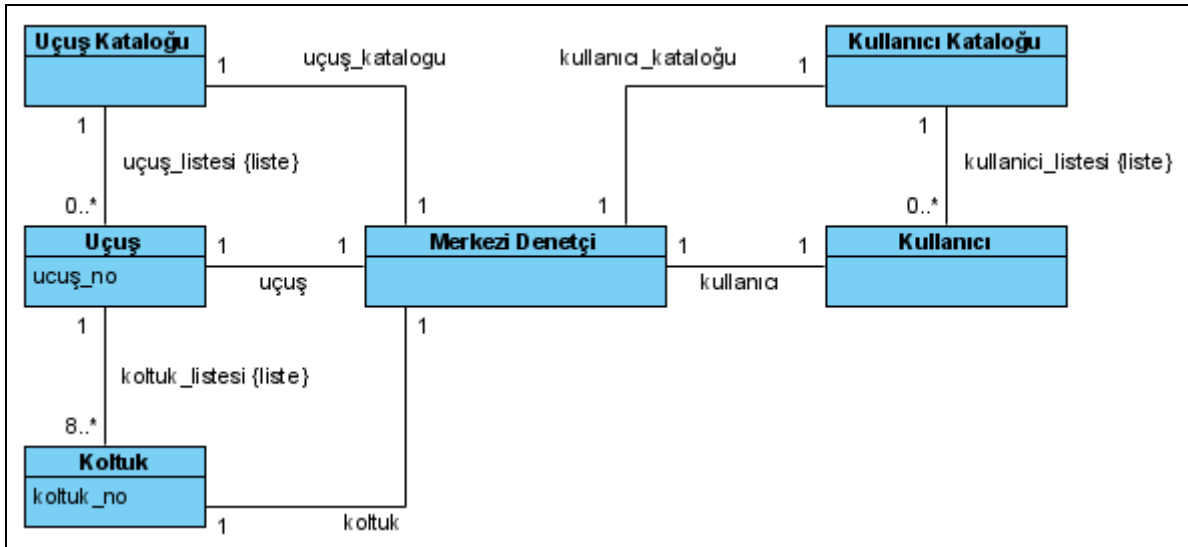
a. Genel Tasarımın belirlenmesi

Şekil-3'teki temel kavramsal sınıflar incelendiğinde, kullanıcının yaptığı işlemleri sorgulayan ve gerçekleyen bir denetçiye ihtiyaç olduğu görülmektedir.

Eklenecek olan sistem denetçisinin yapması gereken temel işlevler aşağıdaki gibi sıralanabilir:

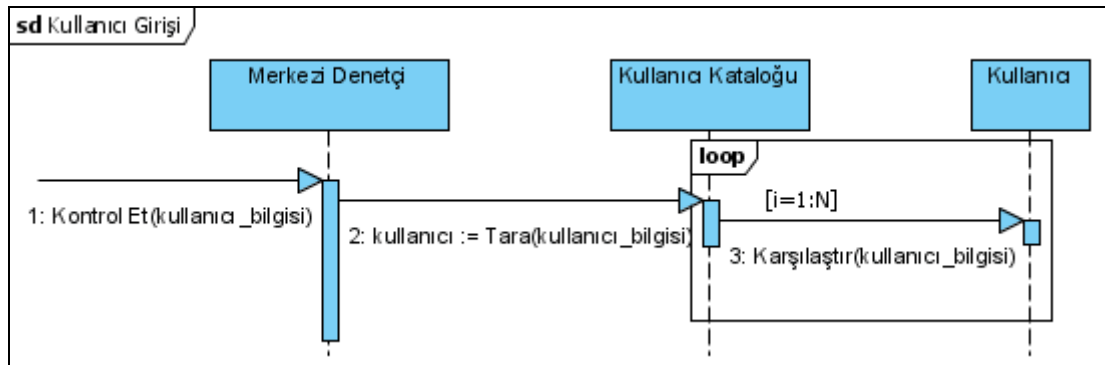
- Sisteme hangi kullanıcının geldiğini, kullanıcı kataloğunu tarayarak elde etmek
- Kullanıcı istediğinde, uçuş kataloğundan belli tarih ve güzergahtaki listeyi çekmek
- Kullanıcı istediğinde, istenen uçuştaki koltukların durumunu almak
- Kullanıcı için istenen rezervasyonu tutmak, gerektiğinde kayıt defterinden çekmek

Bu durumda, analiz aşamasındaki modeli MERKEZİ denetçinin etrafında şekillendirmiş oluruz.



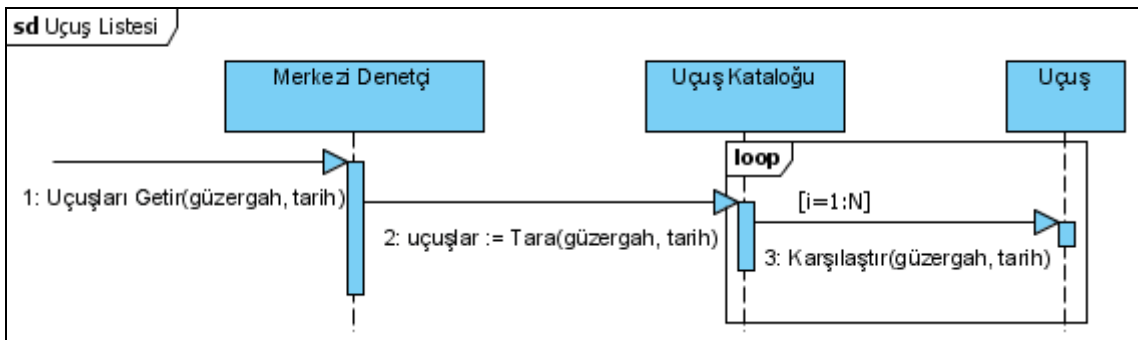
Şekil 6: UML - Tasarımın Başlangıcı

Görüldüğü üzere tüm bilgi alışverişleri merkezi denetçinin üzerinden yapılacaktır. Ayrıca, müşteri ile yapılan görüşmenin ardından her rezervasyonda tek bir koltuğun ayrılması yeterli görüldüğünden tasarımda bu kısım güncellenmiştir. Ayrıca, merkezi denetçiye kullanıcı ve uçuş kataloğu bilgilerinin verildiği varsayılacaktır (bu bilgilerin nasıl aktarıldığı ileride açıklanacaktır). Tasarımın bu ana kadarki kısmıyla ilgili etkileşim diyagramları aşağıda verilmiştir.



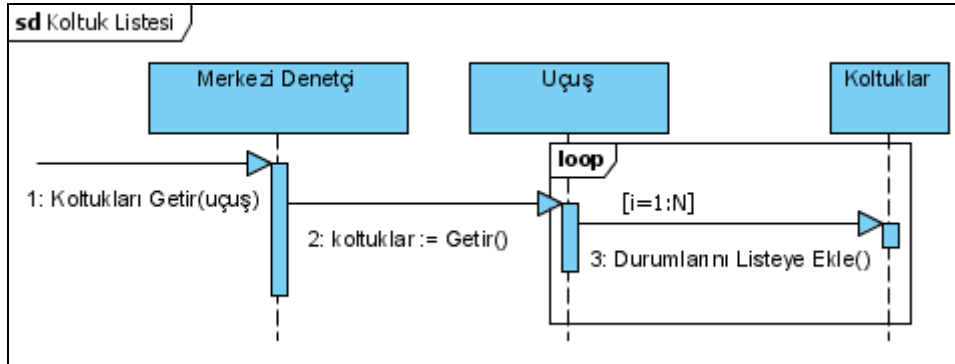
Şekil 7: Kullanıcı Girişi

Her yeni kullanıcı gelişinde yeni bir merkezi denetçi oluşturulduğu varsayılmaktadır. Bu denetçi kendisine verilen yeni kullanıcı bilgisini (kullanıcı adı, şifre, ... ?) kullanıcı kataloğuna taratır ve kullanıcının sistemde kayıtlı olup olmadığını öğrenir. Böylece kullanıcı niteliğini günceller.



Şekil 8: Uçuş Taraması

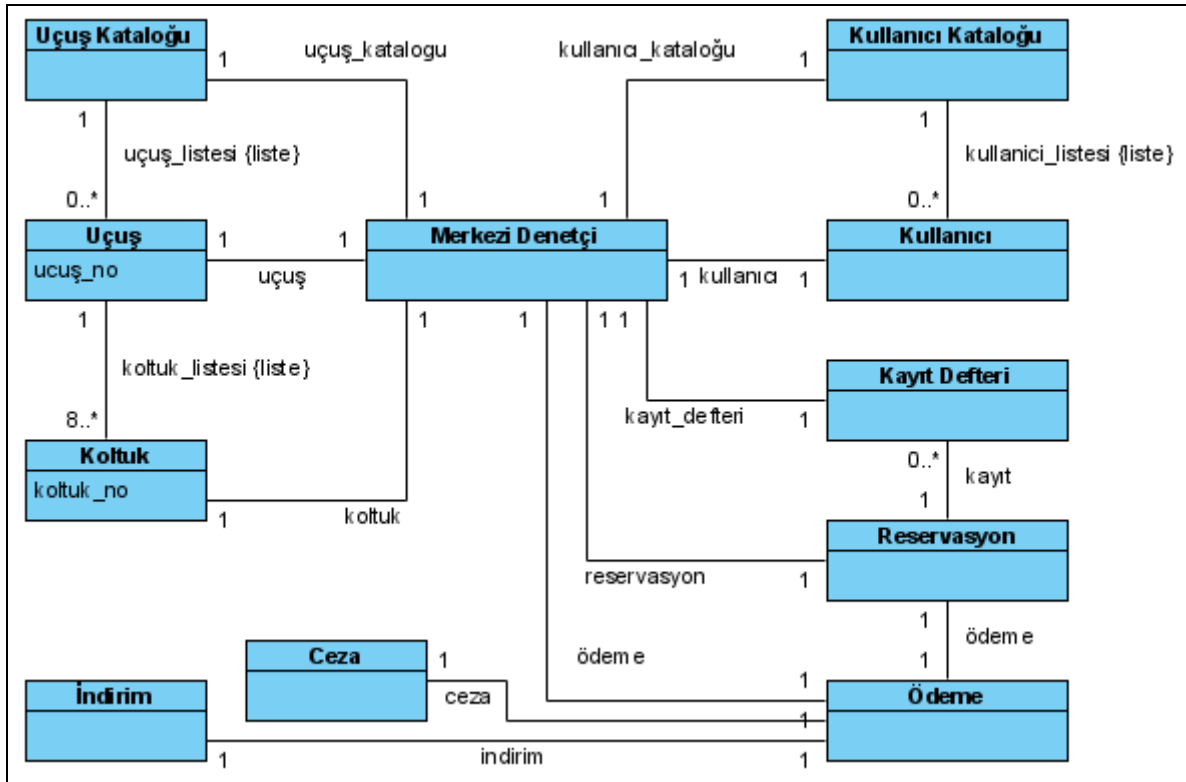
Yukarıdaki ardışıl diyagramda da görüldüğü üzere, kullanıcı uçuş listesini belli güzergah ve tarihe göre taramak istediğinde sistem uyarılır. Merkezi denetçi bu görevi uçuş kataloğuna verir ve uygun uçuş listesini alır.



Şekil 9: Koltuk Listesi

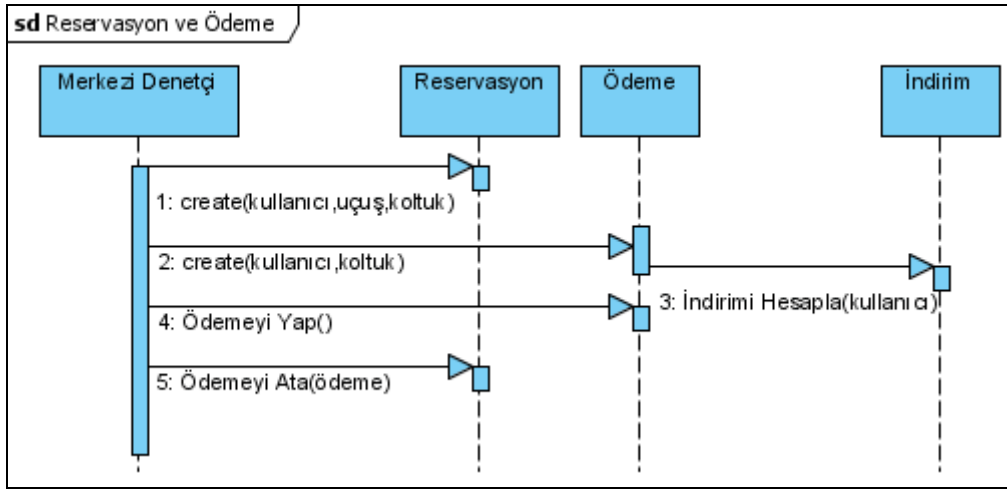
Önceki ardışıl diyagramlara benzer şekilde, merkezi denetçi uyarıldığında, koltuk listesini önceden seçilen uçuştan öğrenir. Öğrendiği bu liste ekrana çıkarılır ve sonuçta müşteri istediği boş koltuğu seçerek işlemine devam edebilir.

Merkezi denetçi seçilen koltukla ilgili bilgilendirildikten sonra, rezervasyonun yapılması istenecektir. Bu durumda merkezi denetçi elindeki kullanıcı, uçuş, koltuk verilerine göre önce indirim hesaplatmalı (benzer şekilde iptal için ceza hesaplanmalı) ve rezervasyonu oluşturmalıdır. Paranın tahsilinin onayını aldıktan sonra, rezervasyonu kayıt defterine kaydetmeli ve koltuk listesindeki gerekli güncellemeleri yapmalıdır. Bunların sağlanması için sınıf diyagramına yapılacak gerekli eklemeler aşağıda verilmiştir.



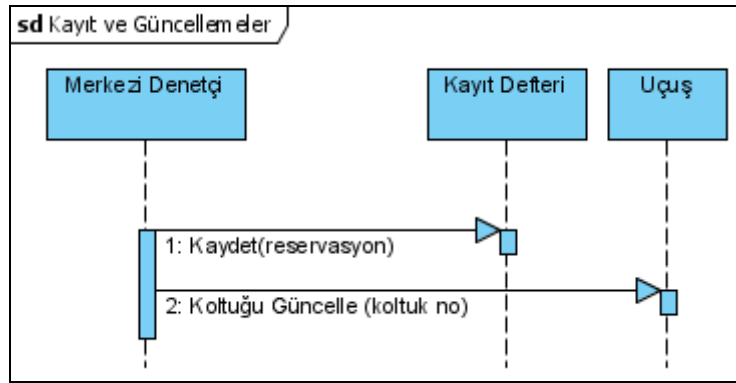
Şekil 10: Genel UML Sınıf Tasarımı

Yeni bir rezervasyon yapma işlemi, aşağıdaki ardışıl diyagramda açıklanmıştır.



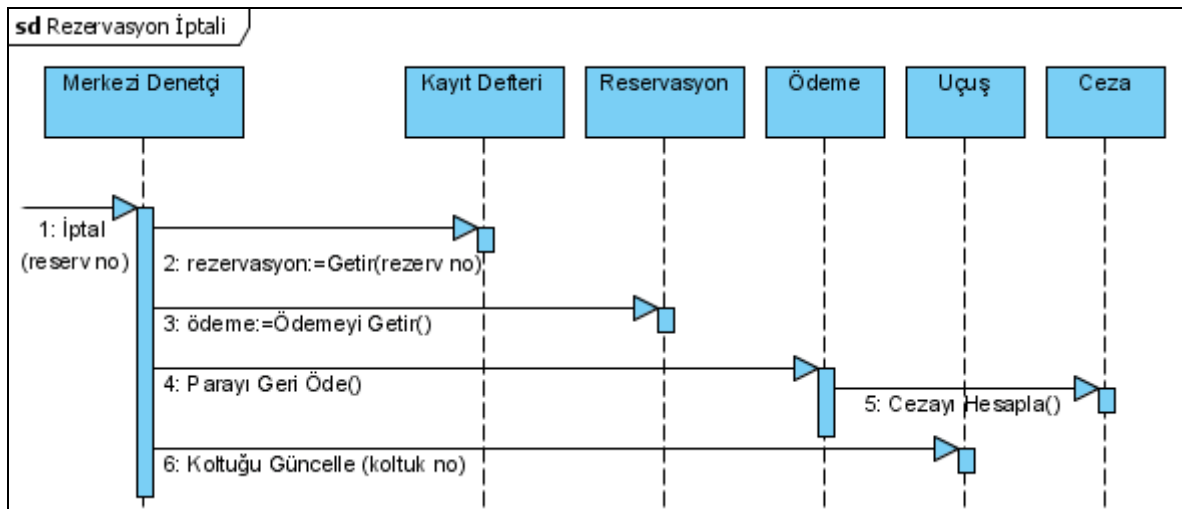
Şekil 11: Rezervasyon ve Ödeme

Rezervasyonu ve ödemeyi oluşturma işlemi görüldüğü gibi bilginin uzmanı olarak merkezi denetçiye verilmiştir. Görüldüğü üzere ödeme nesnesini sonradan değişmeyeceği varsayılmış ve tüm kredi kartıyla ilgili bilgileri gerçekleştirebildiği varsayılmıştır. Rezervasyon ve ödeme işlemi gerçekleştirildikten sonra, rezervasyonun kaydı ve koltuk bilgileri ile ilgili gerekli güncellemelerin yapılması gerekmektedir. Bunlarla ilgili olan ardışıl akış diyagramı aşağıda verilmiştir.



Şekil 12: Kayıt ve Güncelleme

Bu aşamanın ardından, rezervasyon ile ilgili bir bilginin ekrana verilerek kullanıcının kaydetmesi istenecektir. Böylece bu bilgiyle rezervasyon iptal işlemi gerçekleştirilebilir.



Şekil 13: Rezervasyonun İptali

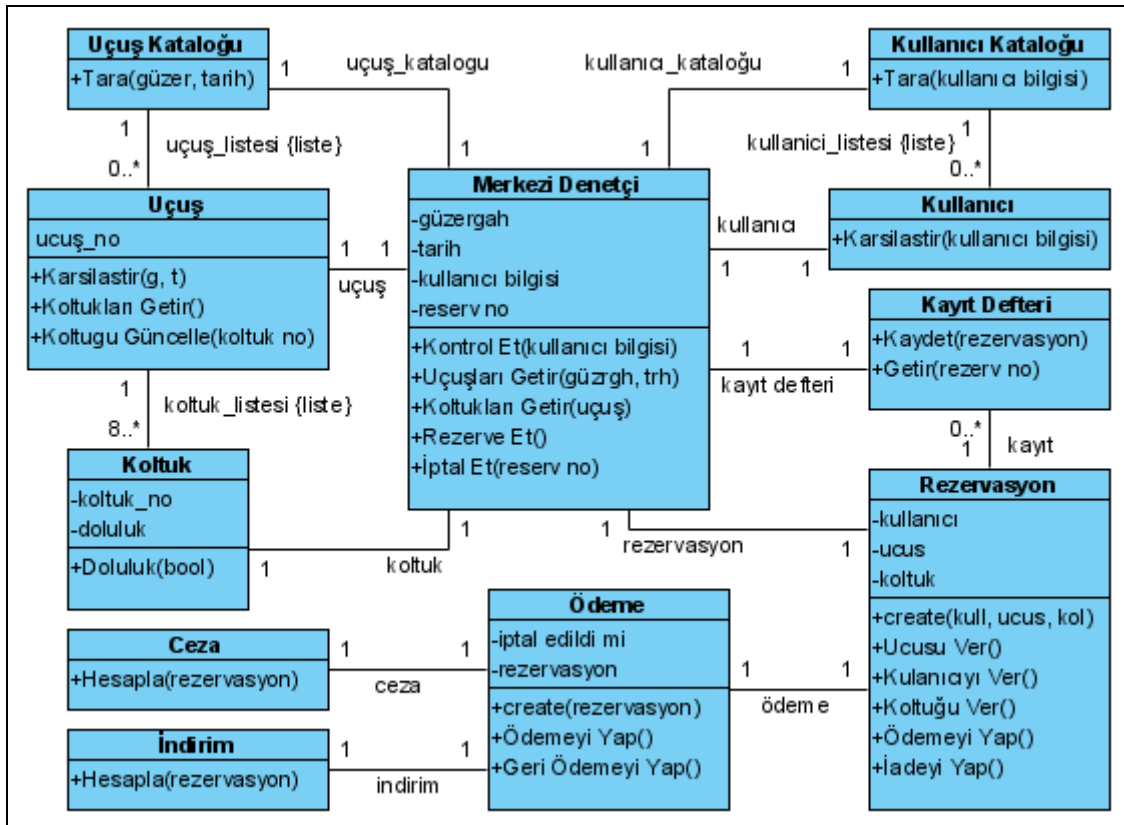
b. İndirim ve Ceza ile İlgili Güncelleme

Yukarıdaki tasarımda indirim ve cezanın ÖDEMEye bağlandığı kabul edilmiştir. Oysa, indirim ve cezanın uçuş tipine, kullanıcı tipine, saate ve önceden yapılan uçuşlara göre değişmesi olasıdır. Bu durumda ÖDEME bazı bilgilerin alınması konusunda eksik kalmaktadır.

Az bağımlılık ve işin uzmanı ilkeleri arasında seçim yapmak zorundayız. Ya ödemeye ilgisi olmayan rezervasyon, indirimi elindeki bilgilerle hesaplayacak; ya da uçuş tipiyle ilgili olmayan ödemeye gerekli bilgiler de verilecektir.

İndirim ve cezanın rezervasyonla ilgili her türlü bilgiye sahip olması olasıdır. Bu durumda indirim ve ceza hesaplanırken rezervasyon bilgisini almaları gerekecektir.

Sonuç olarak, ödemenin oluşturulması ve gerekli bilgileri hesaplaması görevi merkezi denetçiden alınmış ve rezervasyona verilmiştir. Böylece, ödeme oluşturulurken rezervasyon kendi bilgisinin tamamını buraya aktarabilir. UML sınıf diyagramında bir bağlantı (merkezi denetçi - ödeme) eksilir.



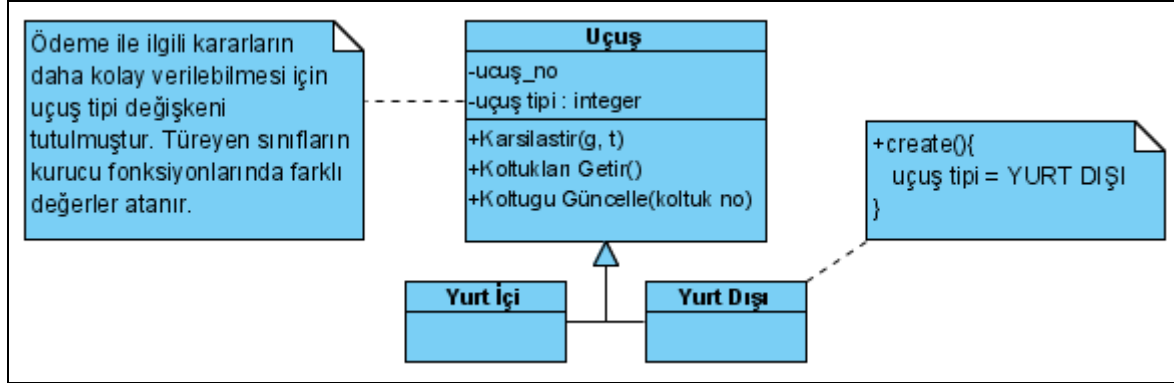
Şekil 14: Genel UML Sınıf Diyagramının Son Hali

Gelinen son durumu kısaca açıklamak gerekirse; öncelikle merkezi denetçi kullanıcı kataloğundan kullanıcıyı çeker. Ardından, istenen güzergah ve tarih bilgilerine göre uçuş kataloğundan uygun uçuşları alır. Sonra da istenen koltuk bilgilerini alır. Bu durumda, merkezi denetçi yapılacak rezervasyonla ilgili tüm bilgileri elde etmiştir. Son aşamada, merkezi denetçi rezervasyonu oluşturur ve ödemenin yapılmasını bekler. Ödeme yapılırken indirim hesaplatılır. Merkezi denetçi son olarak rezervasyonu kayıt defterine kaydeder.

Rezervasyon iptal edilmesi aşamasında ise, merkezi denetçi rezervasyon no ile kayıt defterinden rezervasyonu çeker ve ödemeyi iptal ettirir. Ödeme iptal edilirken, ceza hesaplatılır.

c. Değişken Kısımların Güncellenmesi

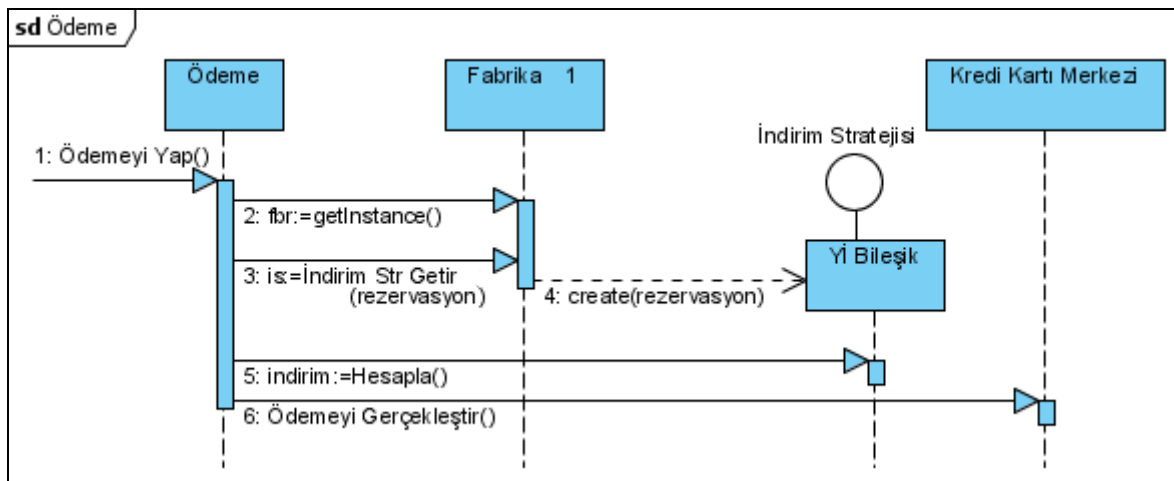
Üç önemli değişebilirlik kısmı tespit edilmiştir. Bunlardan ilki uçuştur. Şimdilik yurt içi ve yurt dışı olmak üzere iki tip uçuş mevcuttur. Ancak, ileride yenileri de eklenebilir. Bu uçuşların fiyatları, indirimleri ve cezaları farklı olarak hesaplanacağından çok şekillilikten (adaptör kalıbı) yararlanılarak tasarlanabilir.



Şekil 15: Değişken Uçuş Sınıflarının Tasarımı

Değişebilecek kısımlardan ikincisi indirim sınıfıdır. O halde, indirim sınıfını değişimlere açık bir yapıda tasarlamak gerekir. Ayrıca, aynı anda birkaç indirim uygulanabileceği için, tasarımında strateji kalıbı kullanılacaktır. Birden fazla stratejinin aynı anda uygulanabilmesi doğal olarak, bileşik nesne kalıbı ihtiyacını doğurmaktadır. Doğal olarak, stratejik nesnelere hangisinin kullanılacağını belirleyen tekil fabrika nesnesine ihtiyaç bulunmaktadır. Bu durumda hangi indirim veya cezayı kullanacağını bilmeyen bir sınıf, fabrikadan bu bilgiyi alabilir.

Ödeme öncesi kısım, önceden açıklanan genel kısımın aynısıdır ve herhangi bir değişikliğe gerek yoktur. Ancak ödemenin yapılması ve geri iadenin gerçekleştirilmesi, ayrıca indirim ve cezaların hesaplanması ile ilgili aşağıdaki tasarımlar değiştirilmiştir. Öncelikle bu iki ardışıl etkileşim diyagramını inceleyelim, hemen ardından ceza ve ödeme sınıflarının nasıl tasarlandığını UML sınıf diyagramında inceleyelim.

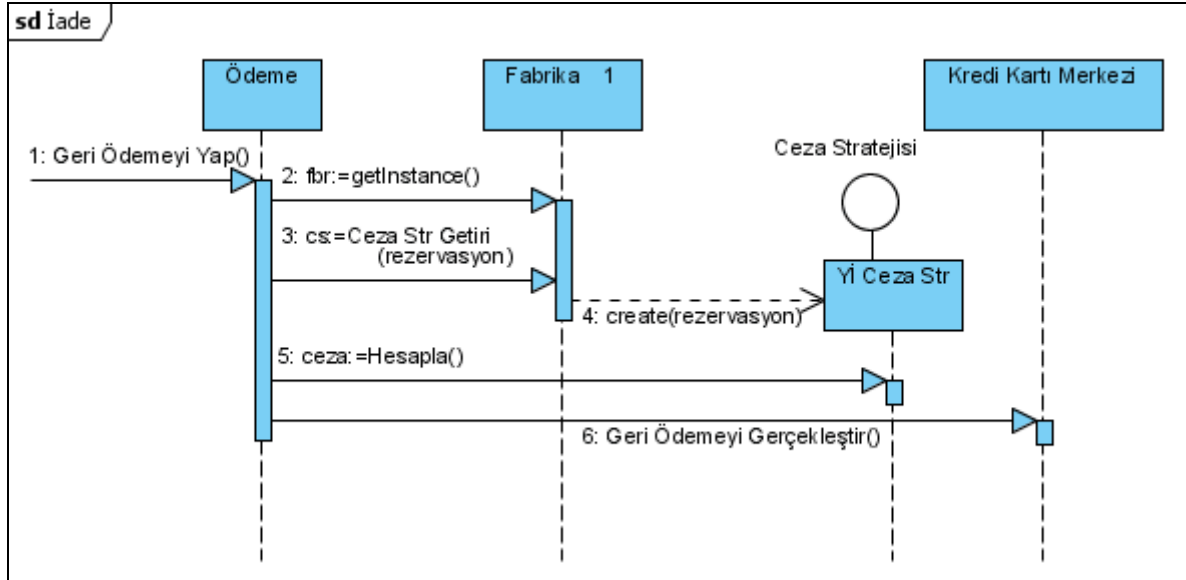


Şekil 16: Ardışıl Etkileşim Diyagramı – Ödeme

Yukarıdaki ardışıl etkileşim diyagramında ödemenin yapılması sırasında strateji kalıbından ve tekil fabrikadan nasıl yararlandığı gösterilmiştir. Fabrikadan indirim stratejisi istenmiş ve bu strateji yardımıyla rezervasyona yapılacak indirim belirlenmiştir.

Önceden de belirtildiği gibi, ödemenin gerçekleştirilmesi durumunda merkezi denetçi bilgilendirilecektir. Bilgilendirilen merkezi denetçi de koltuk bilgilerini güncelleyip, rezervasyonu kayıt defterine ekleyecektir.

Kayıt defterinin veri tabanını etkili bir şekilde kullandığı varsayılmaktadır. Projenin senaryolarına alınmadığından bu işlem ayrıntılarıyla gerçekleştirilmemiştir.



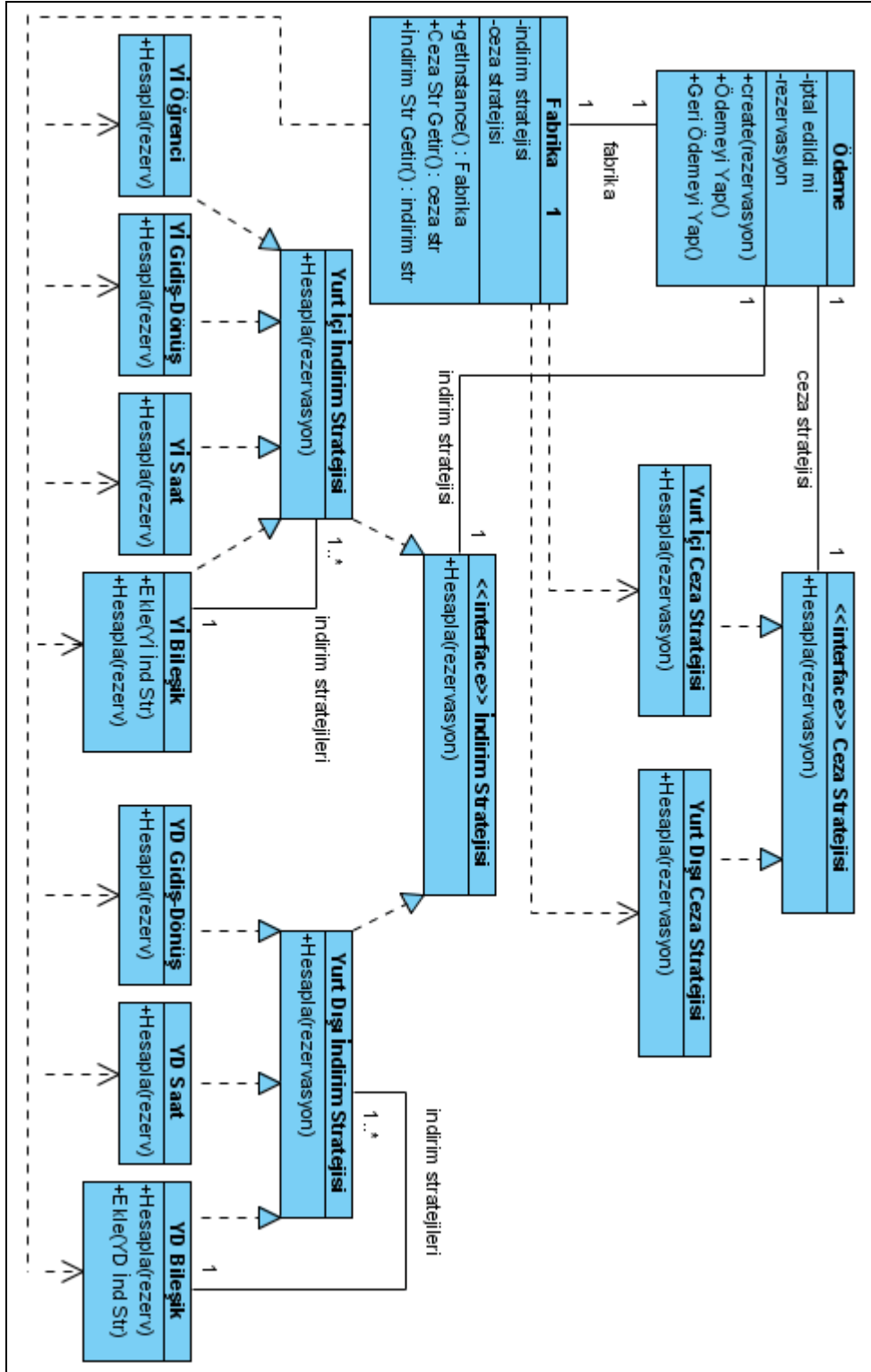
Şekil 17: Ardışıl Etkileşim Diyagramı - Geri İade

Yukarıdaki şekilde gösterilen ardışıl etkileşim diyagramında, ödeme nesnesinde para iade işlemini nasıl gerçekleştirileceği anlatılmaktadır. Öncelikle tekil fabrikadan ceza stratejisi alınmakta, ardından gerekli kesinti yapılarak kredi kartı merkezine para iade edilmektedir.

Tasarımda görüldüğü üzere, ödeme nesnesinin değişmeyeceği dolayısıyla, birçok bilgiyi içinde barındırdığı varsayılmıştır. Dış bağlantılar için ayrıca yeni sınıflar oluşturulmamıştır.

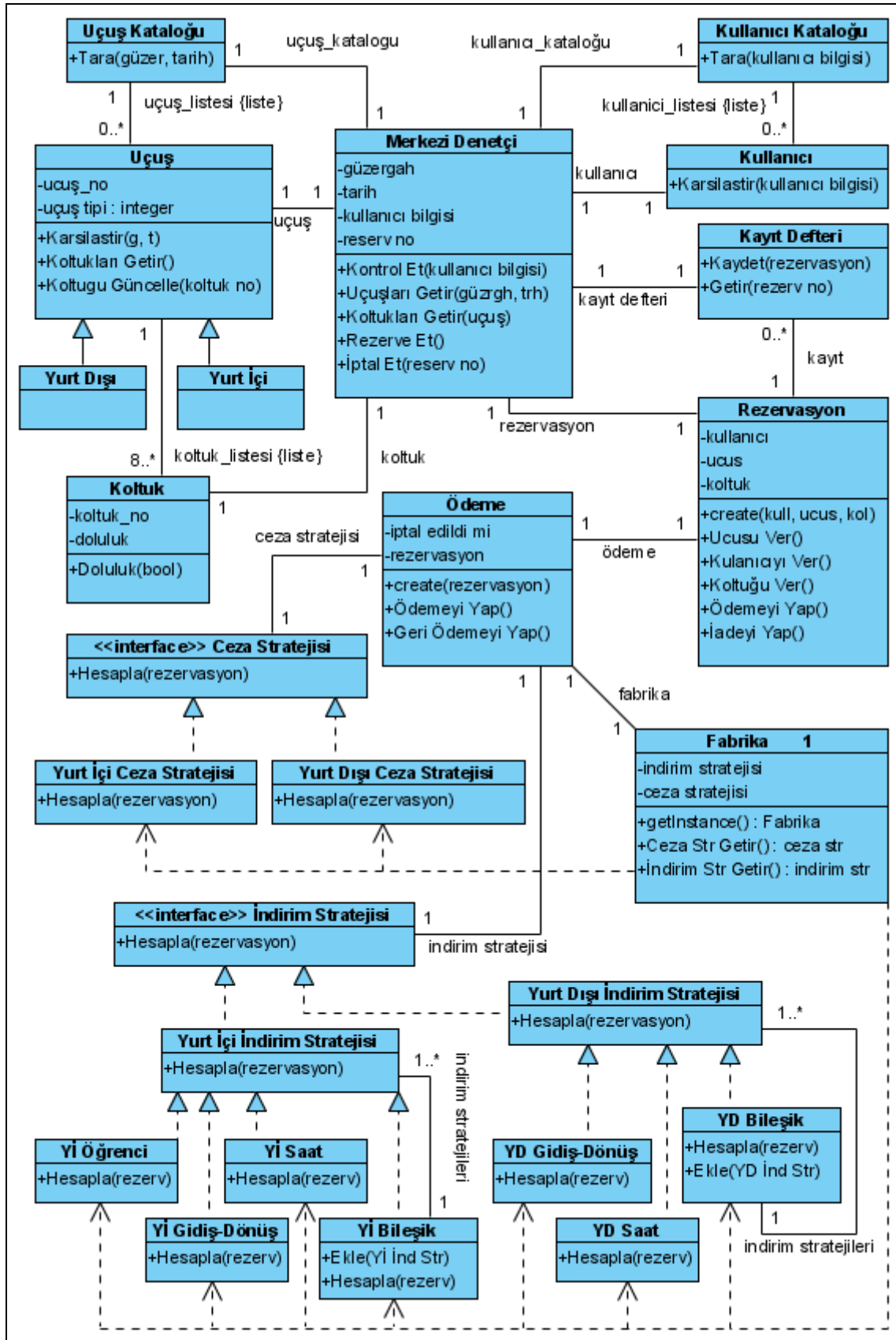
İadenin başarılı bir şekilde gerçekleştirilmesi durumunda merkezi denetçi bilgilendirilecek ve koltuk bilgisini güncelleyecektir.

Bu tasarımı gerçekleyecek olan UML sınıf diyagramı tasarımı aşağıda verilmiştir.



Şekil 18: İndirim ve Ceza Stratejileri, Tekil Fabrika Tasarımı

Son olarak, yukarıda verilen indirim ve ceza tasarımını, ayrıntılı UML sınıf diyagramına eklersek aşağıda verilen son haline ulaşmış oluruz. Böylece istenen tasarımı, prensip ve kalıplardan yararlanarak etkili çalışacak şekilde gerçekleştirmiş olduk.



Şekil 19: Ayrıntılı UML Sınıf Diyagramının Son Hali

E. Sonular ve Eleřtiriler

Proje dahilinde havayolu firmaları iin rezervasyon ve bilet satıř sisteminin altyapısı tasarlanmıřtır. Tasarımda ařađıdakiler gereklenmiřtir.

- Kullanıcı guzergah ve tarih bilgisini vererek ilgili seferleri gorebilir.
- Herhangi bir seferi setiđinde, bu uuřtaki koltukların durumlarını gorebilir (boř / dolu).
- Koltuk numarası vererek rezervasyon yapabilir.
- Mřteri rezervasyonu iptal edebilir.
- Rezervasyon iptalinde eřitli cezalar kesilebilir (DEĐİŐEBİLİR).
- Sistemde eřitli uuř tipleri mevcut (DEĐİŐEBİLİR).
- Sistemde eřitli indirimler mevcut (DEĐİŐEBİLİR).
- Tm rezervasyon ve iptal iřlemlerinin kayıtları sistemde tutulur.

Özellikle deđiřebileceđi dřnlen  kısım zerinde yođun bir alıřma yapılarak, ileriki ařamalarda en az zaman kaybıyla gncellenebilmeleri sađlanmaya alıřılmıřtır. Bunun iin; adaptr, strateji, tekil nesne ve fabrika kalıplarından yararlanılmıřtır. Ayrıca, sistemin geri kalan tasarımında bilginin uzmanı, az bađımlılık, deneti ve iyi uyum prensipleri gzetilmeye alıřılmıřtır.

Yapılan tasarımın dođal sonucu olarak, ileriki ařamalarda deđiřmesi beklenen kısımlara daha kolay mdahale edilebilecektir. alıřmanın bazı ařamalarındaki ayrıntılar atlanmıř, genel tasarımın ardından deđiřmesi beklenen kısımlara daha fazla ađırlık verilmeye alıřılmıřtır.

Bu alıřma, geniř kapsamlı ilk proje tasarımım olduđu iin, tasarımın ilerleme ařamalarında da rahatlıkla grlebileceđi zere birok deđiřim olmuřtur. Bunun bir bařka sebebi de, konuya olduka uzak olmamdır. Ancak, yapılan bu alıřma gstermiřtir ki, byk projelerde yapılacak alıřmalar iin kalıplara ve prensiplere gerekten ihtiya dođmaktadır. nceden hazırlanan sınıfların tekrar kullanılabilirliđi, gncellemelerin kolaylařması, karıřıklıđın azalması ve yeni gelen proje ekiplerinin sisteme daha abuk uyum sađlayabilmeleri iin yazılım modelleme ve tasarım ařamalarının ne denli faydalı olabileceđini grmř oldum.