



T.C.
ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
Nesneye Yönelik Analiz ve Tasarım Dersi

Proje : Hava Yolu Şirket Sistemi

Proje Grubu
090401024 Serhat Rifat Demircan
090401029 Ahmet Can Kepenek

18/05/2012

VİZYON

Havayolu şirket sistemi, havayolu şirketine gerekli olan tüm gereksinimleri karşılayacak, başta müşteriler olmak üzere personeline yararlanacağı bir sistem olmayı amaçlamaktadır.

Sistem, şirket açısından personel atama, müşteriye bilet işlemi gerçekleştirme, uçak atama, yeni liman ekleme gibi işlemlerini yürütür. Müşteri açısından bilet alma, rezervasyon işlemlerini gerçekleştirme ve kargo gönderimi sırasında kargonun durumu hakkında bilgi sahibi olacaktır.

Sistemin yetkili bir kişisi olacaktır. Sistem yöneticisi yetkili kişi olarak sisteme uçak ekleme, liman ekleme, bünyesindeki uçaklara sefer atama, seferlerine mürettebat ekleme gibi görevler yaparken sistemi kullanacak personeli ekler ve bunların yetkilerini düzenler.

Sistemi kullanacak diğer şirket personeller, başta müşteri işlemlerini gerçekleştirmek üzere, uçak ve mürettebatın durumunu, kargo hakkında bilgileri sisteme girer.

Müşteri bilet ve kargo işlemlerini gerçekleştirebilir. Yolcu uçağı için bilet alabilir ya da rezervasyon yaptırabilir. Ayrıca check-in işlemlerini de sistemi kullanarak yapabilirler. Müşteriler öğrenci, çocuk ve sivil olmak üzere üç kategoride değerlendirilir. Çocuklar (2-11 yaş) %25 indirimli, Yetişkinler (12+) normal tarife, Öğrenciler(13-24) %10 indirimli giderler. Ayrıca gidiş-dönüş bilet almak mümkündür. Ayrıca buisness, first class ve ekonomi sınıfı bilet alabilecektir.

ACTOR-GOAL

Sistem Yöneticisi	<ul style="list-style-type: none">• Pilotların bilgilerini girer ve günceller.• Uçaklara sefer atar.• Kabin memurlarının bilgilerini girer ve günceller.• Şirkete personel ekler ve bilgilerini günceller.• Sisteme yeni uçak bilgilerini ekler, günceller.• Uçaklara sefer atar, sefer bilgilerini günceller.• Seferlerin ücret tarifelerini sisteme girer ve günceller.• Yolcu ve kargo uçaklarının yük için kapasitelerini girer.• Seferlere mürettebat atar.• Sistemdeki kişilerin sistemi kullanması için yetkilerini belirler.
Pilot	<ul style="list-style-type: none">• Uçağın eksikliğini, sorunlu ve sorunsuz oluşunu sisteme not düşer.
Hava Limanı Personeli	<ul style="list-style-type: none">• Uçağın iniş-kalkışlarla ilgili durum bilgisini girer.• Mürettebatın durumlarını sisteme girer.
Satış Personeli	<ul style="list-style-type: none">• Müşteriye uygun sefer için sorgulama yapar.• Müşteri için satış ve rezervasyon işlemlerini gerçekleştirir.

Müşteri	<ul style="list-style-type: none">• Uygun bileti güzergahına göre satın alır ve rezervasyon yaptırır.• Sistemde check-in olabilir.• Gönderdiği kargoyu takip edebilir.
---------	--

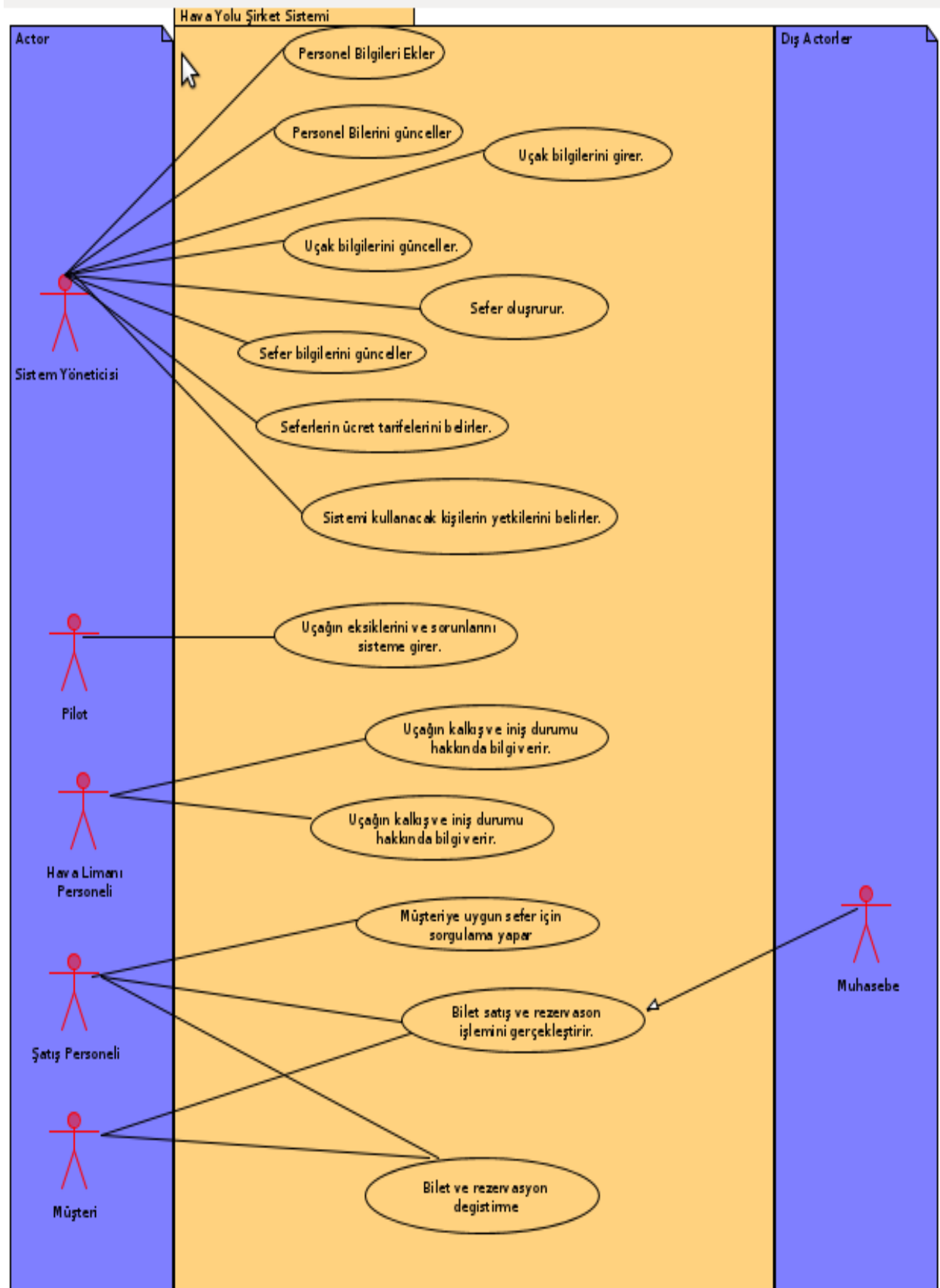
İTERASYON PLANI

1. İterasyon (0-4 Hafta)

Müşteriler bileti, satış personelinden alabilecek veya rezervasyon gerçekleştirebilecek. Ayrıca check-in işlemini yapabileceği bir sistem oluşturulması.

2. İterasyon (4-8 Hafta)

Kargo işlemleri ile ilgili kargo gönderme, alma, kargoyu takip etme gibi sistemin oluşturulması.



USE CASE 1

Primary Actor: Satış Personeli

Ön Koşul: Satış personeli sisteme giriş yapmış olmalı.

Son Koşul: Bilet satışının gerçekleşmesi.

Senaryo

1. Müşteri satış görevlisine nereden nereye gideceğini söyler.
2. Satış personeli verilen bilgileri ilgili kutulara girer ve düğmeye tıklar.
3. Gelen sonuçları saatleriyle birlikte müşteriye söyler.(Öneri sunar.)
4. Müşteri saat kaçta gideceğini, hangi koltuk sınıfından kaç tane bilet istediğini söyler.
5. Müşterinin istediği koltuklar bloke edilir.
6. Satış personeli müşteriden gerekli bilgileri ister.
7. Müşteri gerekli bilgileri verir.
8. Satış personeli müşteriye ödeme şeklini sorar.
9. Müşteri nakit ödeme yapar.
10. Bilgiler sisteme kaydedilir.
11. Satış personeli bileti keser ve bilet bilgilerini yazıcıya gönderir.
12. Satış gerçekleşmiş olur.

Alternatif Senaryolar

*A- Sistem herhangi bir anda çökerse,

1. Sistem yeniden başlatılır.
2. İşlem 2. adımdan devam eder.

3A1. Sistem müşterinin söylediği güzergahta sefer olmayabilir. Bu durumda satış personeli müşteriye varacağı yere en yakın yeri sistemden bulup söyler.

3A2. Müşteri kabul ederse 4 adımdan işlem devam eder.

1. Müşteri kabul etmez ve satış sonlanır.

3B1. Sistem müşterinin söylediği güzergahta sefer olmayabilir. Bu durumda satış personeli müşteriye eğer mümkünse aktarmalı olarak nasıl gidebileceğini söyler.

3B2. Bu durumda müşteri kabul ederse işlem 4. adımdan devam eder.

1. Müşteri kabul etmez ve satış sonlanır.

4A1. Müşterinin istediği sınıfta yeteri kadar koltuk olmayabilir. Bu durumda eğer yer varsa müşteriye başka sınıflardan yer önerilebilir.

4A2. Müşteri kabul ederse 5. adımdan devam eder.

1. Müşteri kabul etmez ve satış sonlanır.

8A1. Müşteri kredi kartı ile ödemek ister.

8A2. Müşteri kredi kartı bilgilerini sisteme girer.

8A3. 9. adımdan devam edilir.

USE CASE 2

Primay Actor: Satış Personeli

Ön Koşul: Satış personeli sisteme giriş yapmış olmalı.

Son Koşıl: Rezervasyon işleminin gerçekleşmiş olması.

Senaryo

1. Müşteri satış görevlisine nereden nereye gideceğini söyler.
2. Satış personeli verilen bilgileri ilgili kutulara girer ve düğmeye tıklar.
3. Gelen sonuçları saatleriyle birlikte müşteriye söyler.(Öneri sunar.)
4. Müşteri saat kaçta gideceğini, hangi koltuk sınıfından kaç tane bilet istediğini söyler.
5. Müşterinin istediği koltuklar bloke edilir.
6. Satış personeli müşteriden gerekli bilgileri ister.
7. Müşteri gerekli bilgileri verir.
8. Koltuklar rezerve edilir.
9. Bilgiler sisteme kaydedilir.
10. Rezerve işlemi gerçekleşmiş olur.

Alternatif Senaryolar

*A- Sistem herhangi bir anda çökerse,

1. Sistem yeniden başlatılır.
2. İşlem 2. adımdan devam eder.

3A1. Sistem müşterinin söylediği güzergahta sefer olmayabilir. Bu durumda satış personeli müşteriye varacağı yere en yakın yeri sistemden bulup söyler.

3A2. Müşteri kabul ederse 4 adımdan işlem devam eder.

1. Müşteri kabul etmez ve işlem sonlanır.

3B1. Sistem müşterinin söylediği güzergahta sefer olmayabilir. Bu durumda satış personeli müşteriye eğer mümkünse aktarmalı olarak nasıl gidebileceğini söyler.

3B2. Bu durumda müşteri kabul ederse işlem 4. adımdan devam eder.

1. Müşteri kabul etmez ve satış sonlanır.

4A1. Müşterinin istediği sınıfta yeteri kadar koltuk olmayabilir. Bu durumda eğer yer varsa müşteriye başka sınıflardan yer önerilebilir.

4A2. Müşteri kabul ederse 5. adımdan devam eder.

1. Müşteri kabul etmez ve satış sonlanır.

USE CASE 3

Primary Actor: Satış Görevlisi

Ön Koşul: Yolcunun bilet satın almış olması.

Son Koşul: Check in işleminin gerçekleşmesi.

Seneryo

1. Müşteri check in işlemi için gelir.
2. Görevliye biletini ve kimliğini verir.
3. Görevli bilet ve kimlik ile bilgileri doğrular.
4. Müşteri koltuk numarasını belirler.
5. Eğer varsa müşteri bagajını teslim eder ve bagaj fişini.
6. Check in işlemi tamamlanır.

Alternatif Senaryolar

*A- Sistem herhangi bir anda çökerse,

- 1.Sistem yeniden başlatılır.
- 2.İşlem 2. adımdan devam eder.

2A. Eğer müşterinin bileti veya kimliği yoksa check in işlemi yapılamaz. İşlem sonlanır.

3A. Kişinin bilet ve kimlik bilgileri doğrulanmaz ise check in işlemi gerçekleştirilemez. Eğer kimlik bilgileri tutarsızsa güvenlik görevlisi çağırılır. İşlem sonlanır.

Supplementary Specification

Fonksiyonellik

Çok fazla kullanım durumu olduğunda fonksiyonel olmalı.

Güvenlik

Yetkisiz kişilerce işlem yapılmasını engelleme ve yetkili kişileri güvenli bir şekilde sistemi kulanması gerekir.

Uygulama Arayüzü

Uygulama arayüzü, sistemi kullanan kişiler için etkin olmalı.

Yedekleme-Kurtarma

Sistemin çökmesi durumunda, sistemin tekrar ayağa kaldırılıp eski işleyişiyle devam etmeli, veri kaybı yaşanmamalıdır.

Fonksiyonel Olmayan Gereksinimler

Zaman kısıtlaması gibi fonksiyonel olmayan gereksinimler yer almalı.

Performans

Sistemi kullananlar, sistemi performanslı bir şekilde kullanmaları gerekir.

Yapılandırılabilirlik

Sisteme yeni donanım eklendiğinde, entegrasyon işlemi kolay olmalı.

Loglama

Sistemin işleyişi hatalarıyla birlikte bir yerde tutulmalı gerektiğinde hata işleme yapılmalıdır.

Business Rules

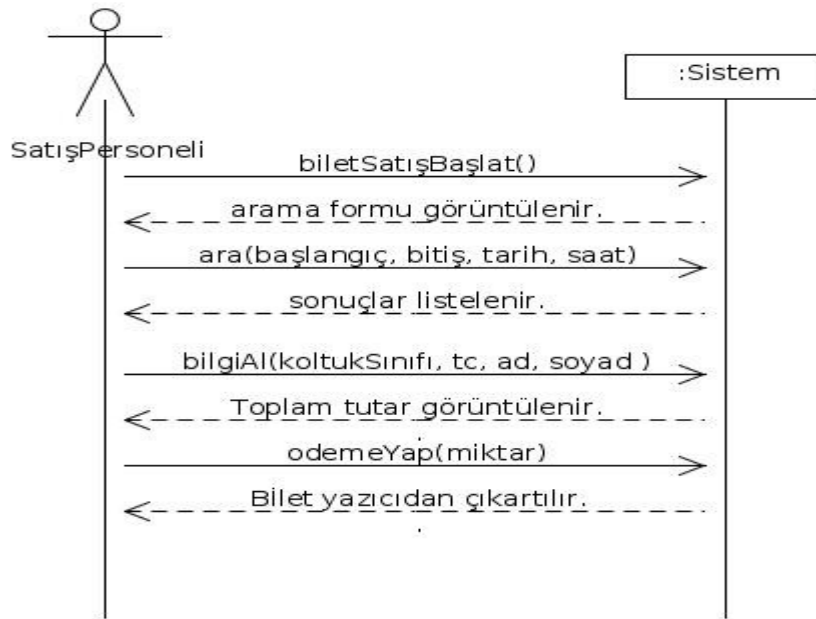
ID	Rule	Changeability	Source
Kural 1	Müşteriler bilet alırken, kendi durumuna göre indirim kazanır.	Yüksek. İndirim kazanma şekilleri değişebilir.	Müşteri politikası
Kural 2	Müşteriler uçuş için farklı sınıflar seçebilirler.	Yüksek. Uçuş sınıfları uçak tasarımına bağlı olarak değişebilir.	Uçuş politikası
Kural 3	7 yaşından küçük olanlar ancak bir büyük ile birlikte bilet alabilir.	Düşük.	Uçuş politikası

Glossary

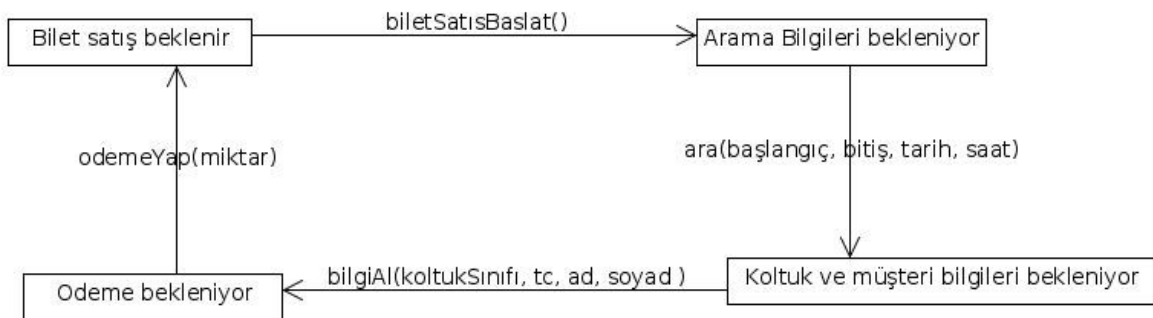
Terim	Açıklama
Uçuş	Uçağın bir havalimanından başka bir havalimanına gitmesiyle gerçekleşen eylem.
Bilet	Uçuşa katılma imkanı veren belge.
Tarife	Fiyat gösteren çizelge.
Kargo Uçağı	Yük taşımak amacıyla tasarlanmış uçak.

Yolcu Uçağı	Yolcu taşımak amacıyla tasarlanmış uçak.
Rezervasyon	Bilet için yer ayırma işlemi.

Bilet Satış SSD



Bilet Satış SMD



Bilet Satış Operational Contracts

CO1:biletSatışBaşlat

operation: biletSatışBaşlat()

cope: Use case – Bilet Satış

precondition: none

postcondition:

- bsatis isminde bilet satış nesnesi oluşturulmuş olmalı.
- bsatis nesnesi kayıtçı ile ilişkilendirilmiş olmalı.
- bsatis nesnesinin özellikleri başlatılmış olmalıdır.

CO2:ara

operation: ara(başlangıç, bitiş, tarih, saat)

scope: Use case – Bilet Satış

precondition: Bilet Satış nesnesi yaratılmış olmalı.

Postcondition:

- uc isminde uçuş nesnesi başlangıç, bitiş, tarih ve saatten sistemde çekilmiş olmalı.
- uc nesnesinin özellikleri başlatılmış olmalıdır.

CO3:bilgiAl

operation: bilgiAl(koltukSınıfı, tcKimlik, ad, soyad)

scope: Use case – Bilet Satış

precondition: Uçuş nesnesi sistemde çekilmiş olmalı.

Postcondition:

- koltukSınıfına göre koltuks isimli KoltukSınıfı nesnesi yaratılmış olmalı.
- mus isminde müşteri nesnesi oluşturulmalı, tcKimlik, ad, soyad özellikleri set edilmiş olmalıdır.
- bil isminde bilet nesnesi oluşturulur.
- bil nesnesi ile uc nesnesi ve müşteri nesnesi ilişkilendirilir.
- koltuks nesnesi ile bil nesnesi ilişkilendirilmelidir.
- bil nesnesi ile bsatis nesnesi ilişkilendirilmelidir.

CO4:odeme

operation: odeme(miktar)

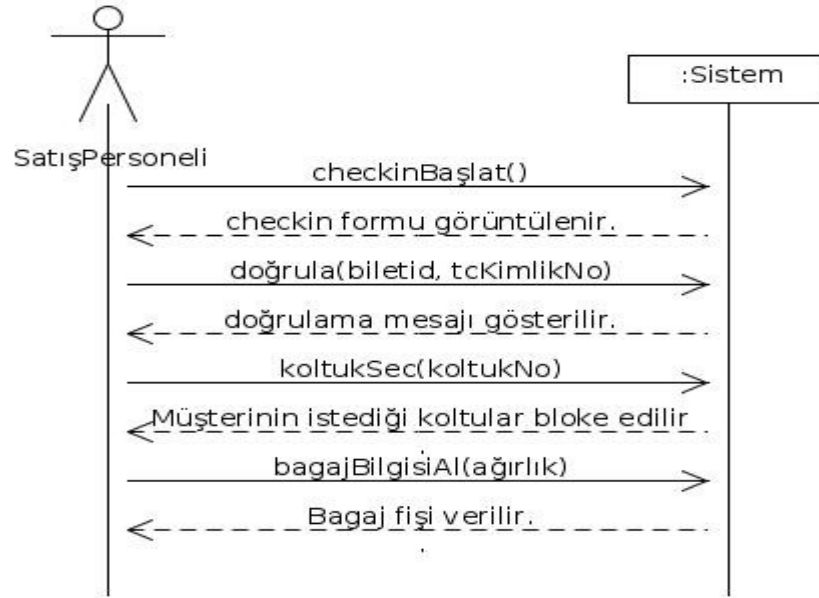
scope: Use case – Bilet Satış

precondition: Bilet nesnesi yaratılmış olmalıdır.

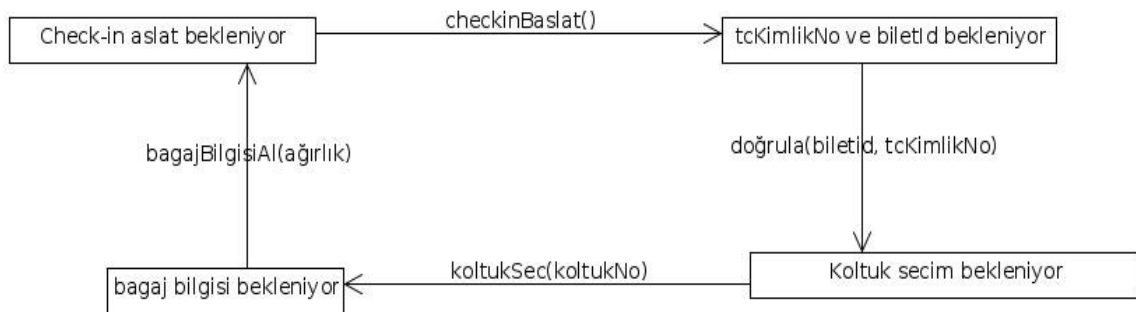
Postcondition:

- nodeme isminde NakitOdeme nesnesi oluşturulmalıdır ve miktar özelliği miktar ile set edilmelidir.
- nodeme nesnesi batis nesnesi ile ilişkilendirilmelidir.

Checkin SSD



Checkin SMD



Checkin Operational Contracts

CO1:checkinİşlemiBaşlat

operation: checkinBaşlat

scope: Use case – Check In

precondition: none

Postcondition

- ci isminde checkinİşlemi nesnesi yaratılmış olmalıdır.
- ci kayıtçı nesnesi ile ilişkilendirilmiş olmalıdır.
- ci nin özellikleri başlatılmalıdır.

CO2:doğrula

operation: doğrula(biletId, tcKimlikNo)

scope: Use case – Check In

precondition: checkinİşlemi nesnesi yaratılmış olmalıdır.

Postcondition:

- biletId özelliğine sahip b isminde ki bilet nesnesine ile tcKimlikNo özelliğine sahip mus ismindeki Müşteri nesnesi sistemden çekilir.

CO3:koltukSeç

operation: koltukSeç(koltukNo)

scope: Use case – Check In

precondition: Bilet nesnesi yaratılmış olmalıdır.

Postcondition:

- set edilir.
- Biletin koltuk sınıfına göre koltuk isimli nesne Koltuk sınıfından oluşturulur ve koltuk numarası koltukNo ile
 - koltuk müşteri ile ilişkilendirilir
 - koltuk ci nesnesi ile ilişkilendirilir.

CO4:bagajBilgisiAl

operation: koltukSeç(ağırlık)

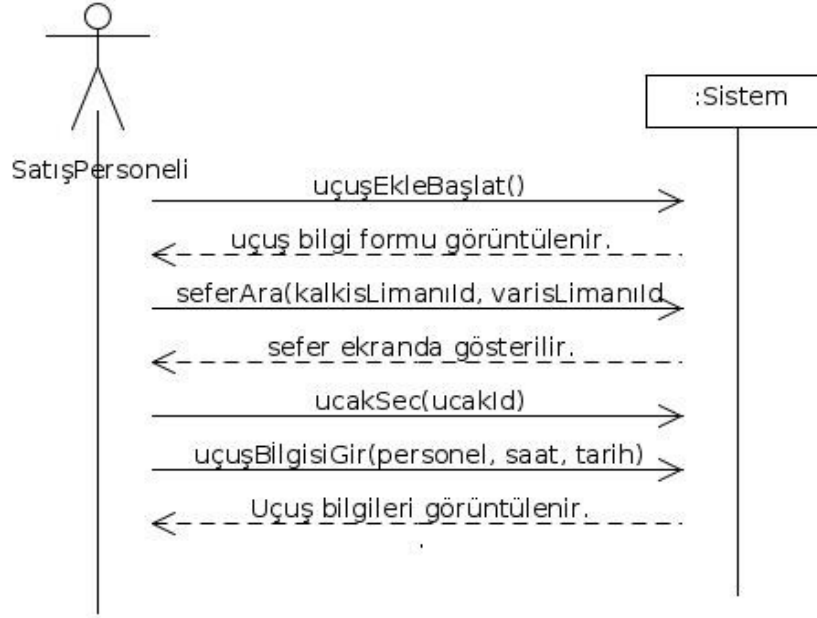
scope: Use case – Check In

precondition: b ismindeki bilet nesnesi sistemden çekilmiş olmalı.

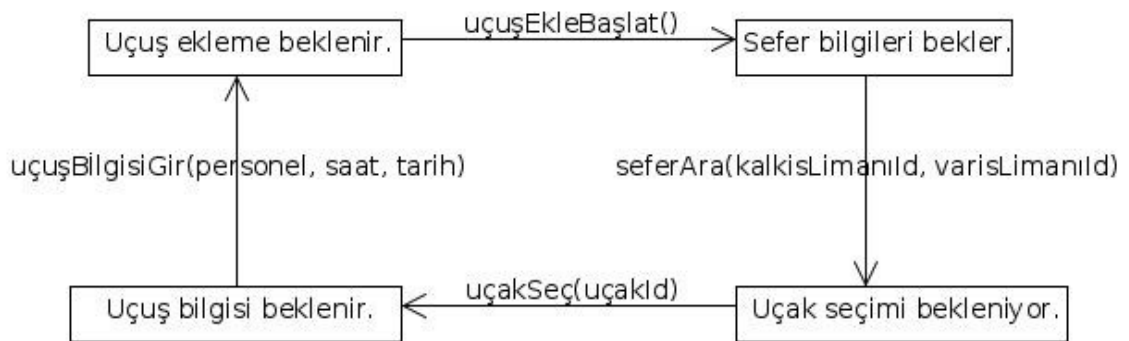
Postcondition:

- bag isimli nesne Bagaj sınıfından türetilmeli. Ağırlık özelliği ağırlık ile set edilmelidir.
- bag nesnesi müşteri ile ilişkilendirilmelidir.
- bf nesnesi BagajFişi sınıfından türetilmelidir. Bf nesnesi bagaj ve checkin işlemi ile ilişkilendirilmelidir.

Uçuş Oluşturma SSD



Uçuş Oluşturma SMD



Uçuş Oluşturma Operational Contracts

CO1:uçuşEkleBaşlat

operation: uçuşEkleBaşlat()

scope: Use case – Uçuş Ekle

precondition: none

Postcondition:

- ue isimli nesne UçuşEkle sınıfından türetilmelidir.
- ue nesnesi kayıtçı ile ilişkilendirilmelidir.
- ue nesnesinin özellikleri başlatılmalıdır.

CO3:seferAra

operation: seferAra(kalkisLimanıId, varisLimanıId)

scope: Use case – Uçuş Ekle

precondition: Uçuş ekle sınıfının nesnesi yaratılmış olmalıdır.

Postcondition:

- kh isimli nesne HavalimanıKatalogunda kalkisLimanıId ile çekilir.
- vh isimli nesne HavalimanıKatalogunda varisLimanıId ile çekilir.
- sistemde sefer isimli nesne kh ve vh nesnelere göre SeferKatalogundan çekilmelidir.

CO4:uçakSeç

operation: uçakSeç(uçakId)

scope: Use case – Uçuş Ekle

precondition: Sefer nesnesi yaratılmış olmalıdır.

Postcondition:

- yu isimli yolcu uçağı nesnesi UçakKatalogundan id ile çekilir.

CO5: uçuşBilgisiGir

operation: uçuşBilgisiGir(personelIdleri, saat, tarih)

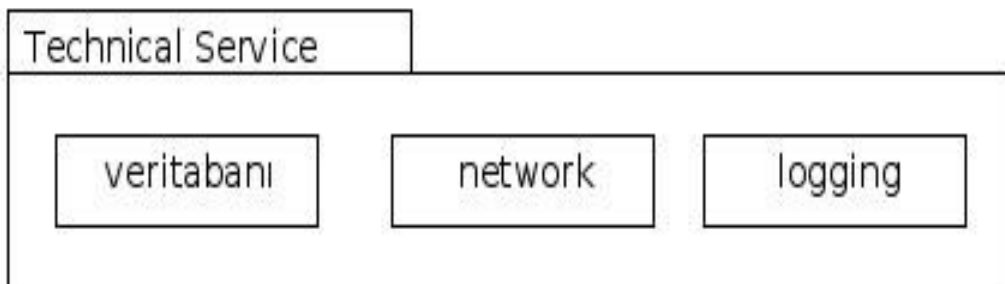
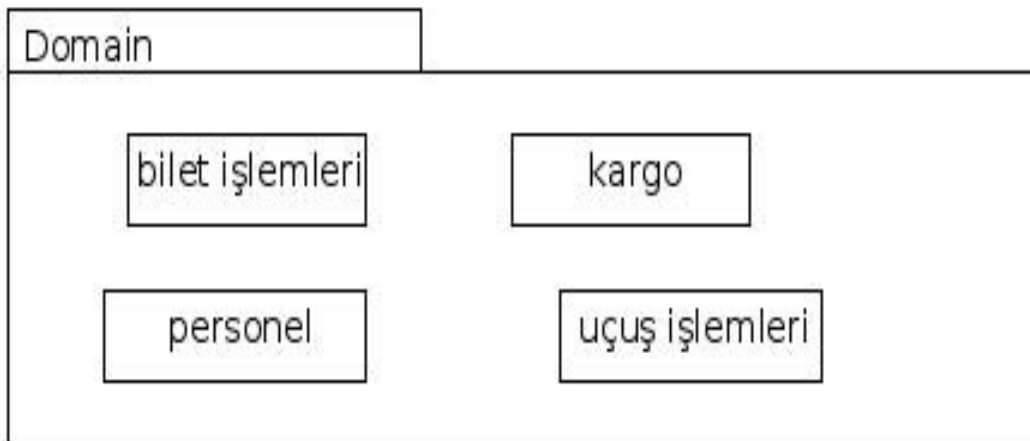
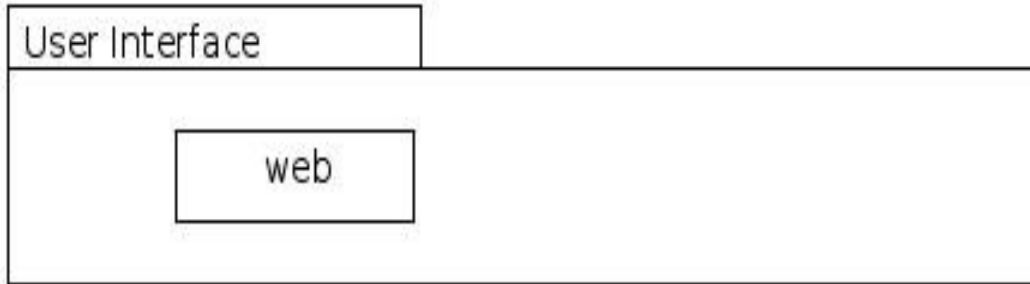
scope: Use case – Uçuş Ekle

precondition: Uçak nesnesi ve sefer nesnesi yaratılmış olmalıdır.

Postcondition:

- personeller sistemden PersonelKatalogundan idleri ile çekilirler.
- uçuş nesnesi saat ve tarih bilgileriyle Uçuş sınıfından türetilir.
- personeller, yu, sefer uçuş ile ilişkilendirilir.
- ue ile uçuş nesnesi ilişkilendirilmelidir.

MANTIKSAL MİMARİ



COMMUNICATION DIAGRAM

