

KARABÜK ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM426 YAZILIM MÜHENDİSLİĞİ – OTEL OTOMASYONU YAZILIM SÜRECİ

TOPIC : 4

BİLGİSAYAR MÜHENDİSLİĞİ %30 (II)

DANIŞMAN

Arş.Gör.Dr. Oğuzhan MENEMENCİOĞLU

Corresponding Member Student ID :2014010206076

ID : 20

HAZIRLAYANLAR

2014010206076 AHMET TALHA ÇAKIROĞLU

2014010206059 KUBİLAY ÖZ

2014010206067 MERT FAİK DELİ

KARABÜK 2019

İÇİNDEKİLER

1. GİRİŞ	3
2. GEREKSİNİM ANALİZİ	4
3. AKTÖRLER	4
4. USE CASE DİYAGRAM	5
5. SIRA DİYAGRAMI	6
6. DURUM MAKİNASI DİYAGRAMI	7
7. PLANLAMA VE SÜREÇ MODELİ	9
8. TASARIM – CLASS DİYAGRAMI	9
9. SİSTEMİN KODU	10
9.1 Oda.cs	10
9.2 Anlasma.cs	10
9.3 Otel.cs	11
9.4 FirmaHesabı.cs	12
9.5 FirmaMüşterileri.cs	12
9.6 Hesap.cs	13
9.7 Muhasebe.cs	13
9.8 Musteri.cs	13
9.9 OthersMusteri.cs	14
9.10 Lokanta.cs	14
9.11 Yiyecek.cs	14
9.12 İçecek.cs	15
9.13 Hizmetler.cs	15
9.14 OdaServisi.cs	15
9.15 KuruTemizleme.cs	16
9.16 Firma.cs	16
9.17 SeyahatAcenta.cs	16
9.18 TicariFirma.cs	16
10. MALİYET	17
11. VERİTABANI (Bonus)	18
12. SONUÇ	19
13. KAYNAKÇA	19

1.GİRİŞ

Proje konusu olarak bize 4. Konu olan otel otomasyonu veri tabanı tasarımının yapılması istendi.

Projemizi tasarlarken izlediğimiz yolu kabaca özetlemek gerekirse; ilk önce GEREKSİNİM ANALİZİ ve bunların çözüm yolu oluşturuldu. Bu gereksinimlere göre AKTÖR'lerimizi belirledik. Bu aktörlerin USE-CASE senarosunu oluşturup diyagramını çizdik. Yine gereksinim analizine göre bu aktörlerin SIRA ve STATE MACHINE diyagramlarını çizdik. Tasarım kısmında ise CLASS diyagramı ve bunlara ek olarak veritabanını tasarladık(ERD).

Son olarak projemizin PLANLANMASI, SÜREÇ MODELİ ve MALİYET HESABI yapılarak tamamlanmıştır.

Bu amaç doğrultusunda veri tabanının kısaca açıklaması şu şekildedir;

- Otelde odalar vardır. Her odanın numarası, yatak sayısı ve gecelik fiyat verileri bulunmaktadır.
- Otelde belirli hizmetler (gömlek yıkama, gömlek ütöleme, elbise ütöleme, ..vb) ve oda servisleri (çay, kahve, ..vb) verilmektedir. Bunların her birinin belirli bir adı ve ücreti vardır.
- Otelin lokantasında yenilip içilen belirli yemekler ve içecekler; bunların her birinin adı ve ücreti vardır.
- Otel odalarında müşteriler kalır. Her oda bir müşteriye verilir ve muhasebe kayıtları bu müşteri için tutulur. Eğer odada birden çok kişi kalırsa, bunlardan biri müşteri, diğerleri ise müşterinin beraberindekilerdir. Otel işletmesi için müşterinin adı, soyadı, cinsiyeti ve adresi ile odaya geliş ve odadan ayrılış tarihlerine ihtiyaç duyulmaktadır. Müşteri beraberindeki kişilerin her birinin adı ve soyadı ile müşteri ile yakınlık derecesinin de saklanması gerekmektedir. Ayrıca her odada kalan müşteri için bir hesap açılması ve odada kalan kişiler tarafından yapılan tüm harcamalarının hizmetler, oda servisi, yemek ve içecekler vb) bu hesaba yazılması istenmektedir.
- Otelin ilişkili olduğu firmalar (ticari şirketler ve seyahat acenteleri) vardır. Bunların her biri ile bir anlaşma yapılmıştır. Bu firmaların ad. ve adresleri ile firmaya anlaşmayla sağlanan indirim oranlarının saklanması gereklidir. Bu firmalardan her biri için de otelde bir hesap açılır. Odalarda kalan müşteriler (ve beraberindekiler) bireysel müşteri ya da firma müşterisi olabilir. Bireysel müşterilerin oda ücretleri de müşteri hesaplarına yazılır. Firma müşterileri için ise, oda ücretleri, anlaşmadaki oranda indirim yapılarak firma hesabına yazılır. Firma hesabında ayrıntılar (hangi müşteri ve beraberindeki kaç kişi için, hangi tarihler arasında ne kadar ücret olduğu ve ne kadar indirim yapıldığı görülecek biçimde) yer alır. Tüm müşterilerin, oda ücreti dışındaki harcamaları müşteri hesaplarına yazılır.

2. GEREKSİNİM ANALİZİ

Aktörlerimizi belirledikten sonra, sistemle ilişkili tüm senaryoları ortaya çıkarıldı. Sistemde olası tüm durumları aşağıdaki şekilde belirtildi.

No	Gereksinim Tanımı	Çözüm Yolu
1	Otelde odalar vardır ve odalar müşteriler tarafından kiralanmalı.	Bir otomasyon sistemi oluşturulabilir.
2	Odalara ait oda servisleri ve hizmetler verilmeli her birinin adı belirlenmeli.	Veri tabanında hizmet ve oda servislerine ait bilgiler tutulmalı.
3	Otelin lokantasında yenilip , içilen yemekler ve içeceklerin adı ve ücreti vardır.	Müşteriye ait lokantadaki harcamaların adı ve ücreti veritabanında tutulmalı.
4	Her oda bir müşteriye ait olmalı ve muhasebe kayıtları bu müşteri için tutulmalı.	Muhasebe kayıtları veritabanında tutulmalı.
5	Eğer odada birden çok kişi kalırsa, biri müşteri, diğerleri ise müşterinin beraberindekiler olarak belirlenmelidir.	Müşteri ve beraberindekiler eşleştirilmeli ve bu şekilde veritabanında kaydedilmeli.
6	Her odada kalan müşteri için bir hesap açılması ve odada kalan kişiler tarafından yapılan tüm harcamalarının bu hesaba yazılması istenmektedir.	Her müşteri için gerçekleştirilen harcamalar veri tabanında kayıt altında tutulmalı.
7	Otelin ilişkili olduğu firmalar ile her biri bir anlaşması olmalıdır.	Otel ile Ticari Firmalar birbiri ile ilişkilendirilmeli ve karışmamalıdır.
8	Yapılan anlaşmaya göre firmayla sağlanan indirim oranı bilinmelidir.	Veri tabanında ticari firmalara ait hotel ile yapılan indirim oranı bilgileri tutulmalıdır.
9	Bireysel müşteri veya firma müşterisi ayrı hesaplanmalıdır.	Veritabanında müşteriler bireysel ve firma olarak veri tabanında tutulmalıdır.
10	Firma müşterilerinin oda ücretleri anlaşma oranının firma hesabına hesaba yazılır.	Firma müşterileri için anlaşmadaki indirim oranında firma hesabında veri tabanında tutulmalıdır.

ŞEKİL 1: OTEL OTOMASYONU REQUIREMENTS ANALYSIS

3. AKTÖRLER

Bu aktörleri seçerken öncelikle oluşturduğumuz sistemle ilişkili objeleri belirledik. Buna göre,

1-Bireysel Müşteri

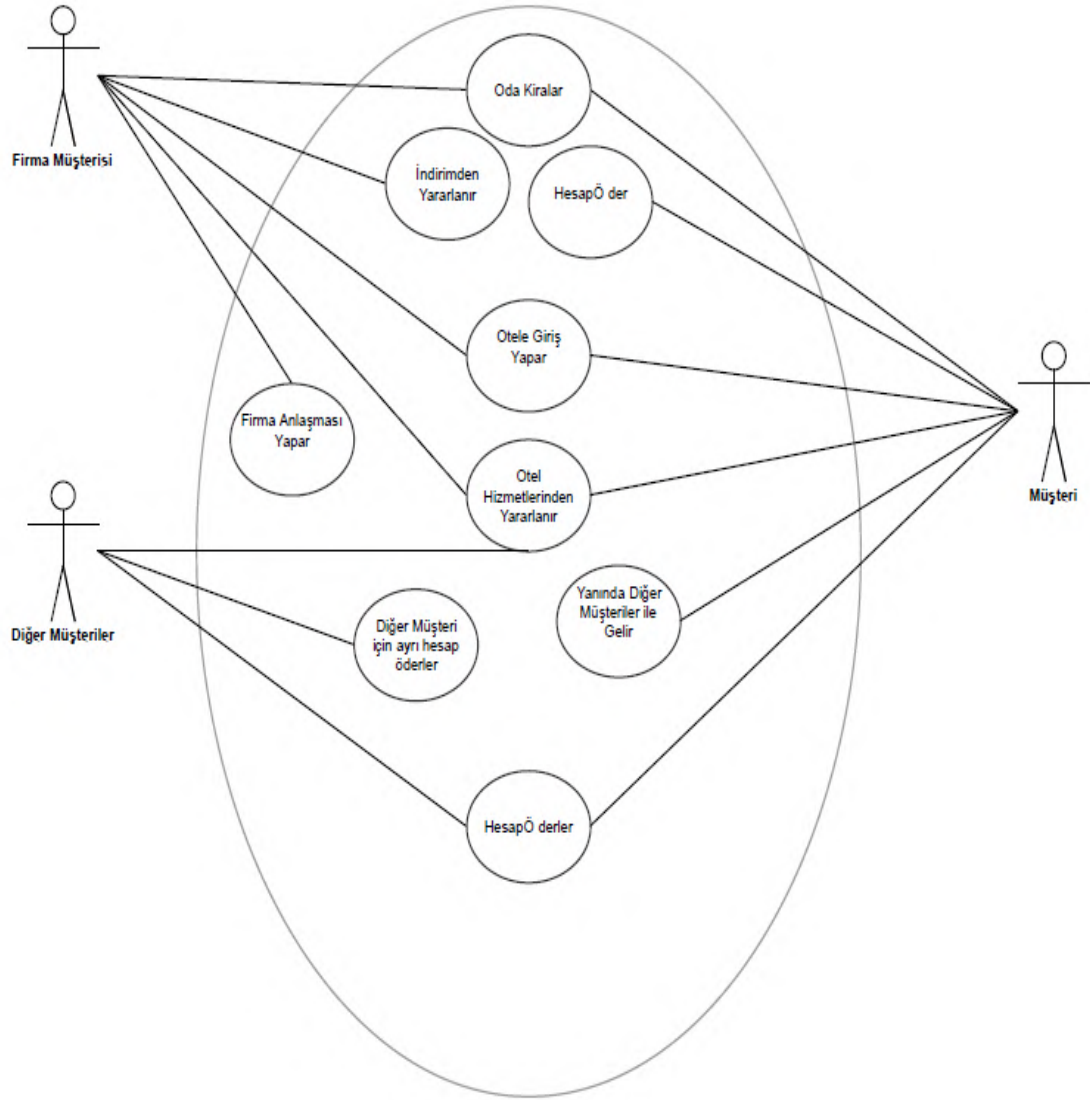
2-Firma Müşterileri

3- Diğer Müşteri (müşteri yakını)

4. USE-CASE DİYAGRAM

Bu diyagramda, oluşturduğumuz aktörlerin sistem içerisinde erişebildikleri tüm olasılıklar göz önüne alınarak hazırlandı.

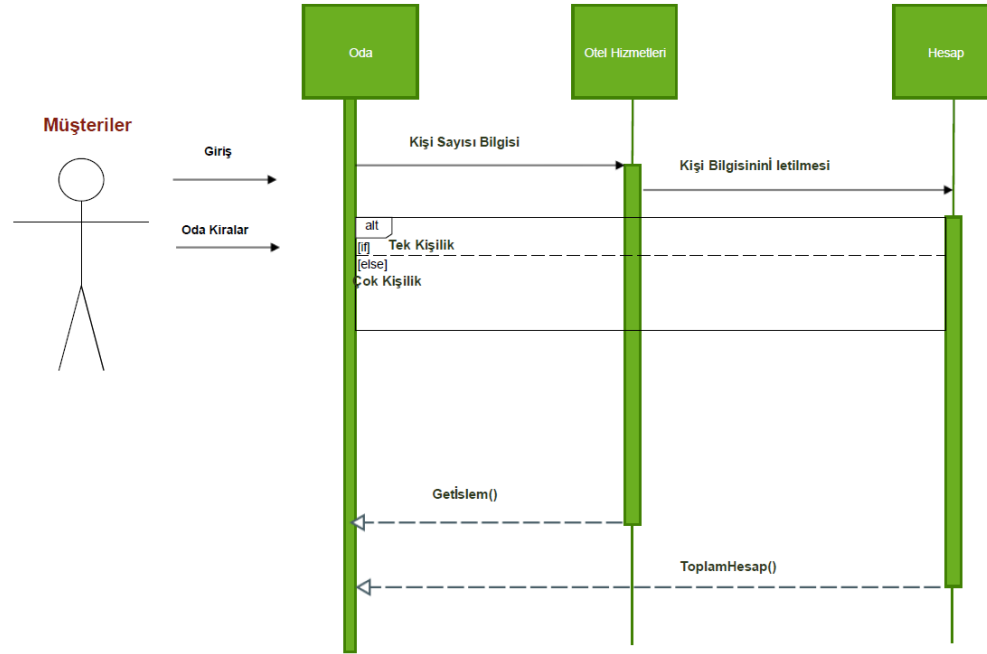
USE-CASE DIAGRAM



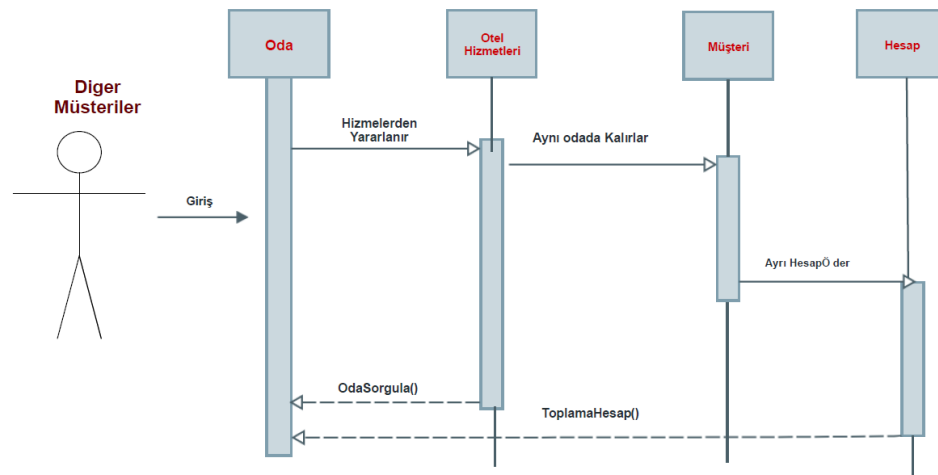
ŞEKİL 2: OTEL OTOMASYONU USE-CASE SENARYOSU

5. SIRA DİYAGRAMLARI

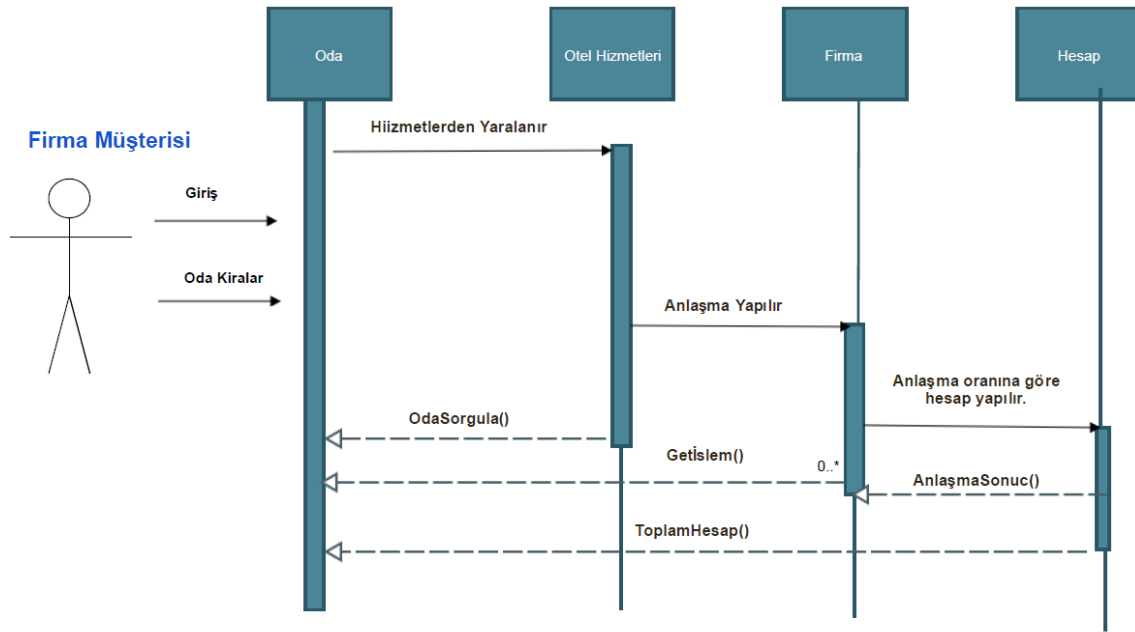
Gereksinim analizinde belirlenen aktörlerin zaman içerisinde sistemde hangi nesneler ile etkileşime gireceğine karar vermek için USE-CASE diyagramından yararlanılarak oluşturuldu. Buna göre;



ŞEKİL 4: OTEL OTOMASYONU BİREYSEL MÜŞTERİ SIRA DİYAGRAMI



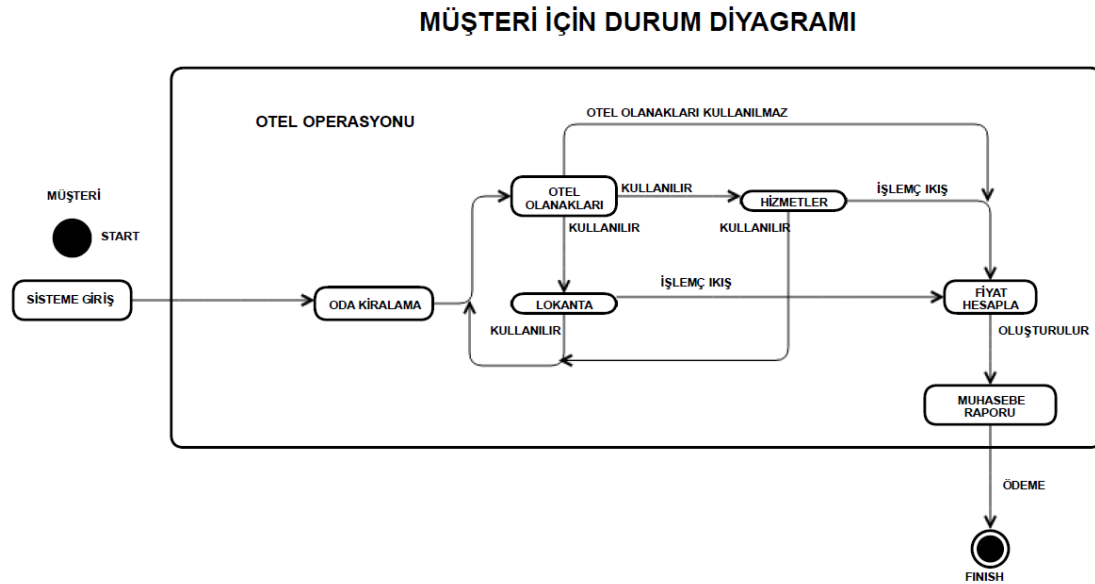
ŞEKİL 5: OTEL OTOMASYONU DİĞER MÜŞTERİLER SIRA DİYAGRAMI



ŞEKİL 6: OTEL OTOMASYONU FİRMA MÜŞTERİLERİ SIRA DİYAGRAMI

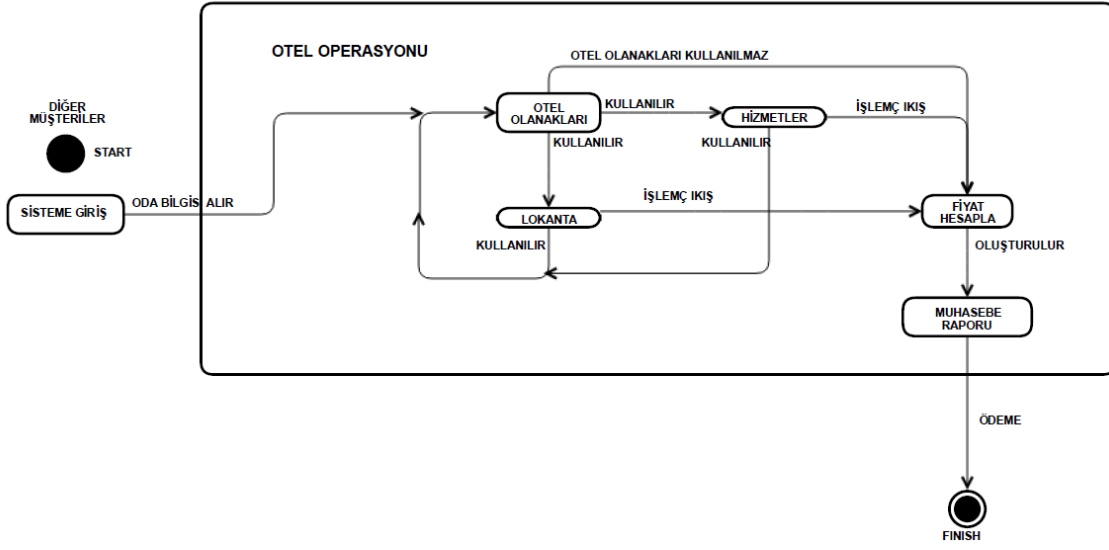
6. DURUM MAKİNASI DİYAGRAMLARI

Durum makinesi diyagramında, belirlediğimiz tüm aktörlerin sistemdeki tüm olası durumların belirlenmesi sağlandı. Kabaca bu aktörlerin aslında sistemde nereye erişebileceklerini göstermektedir. Buna göre;



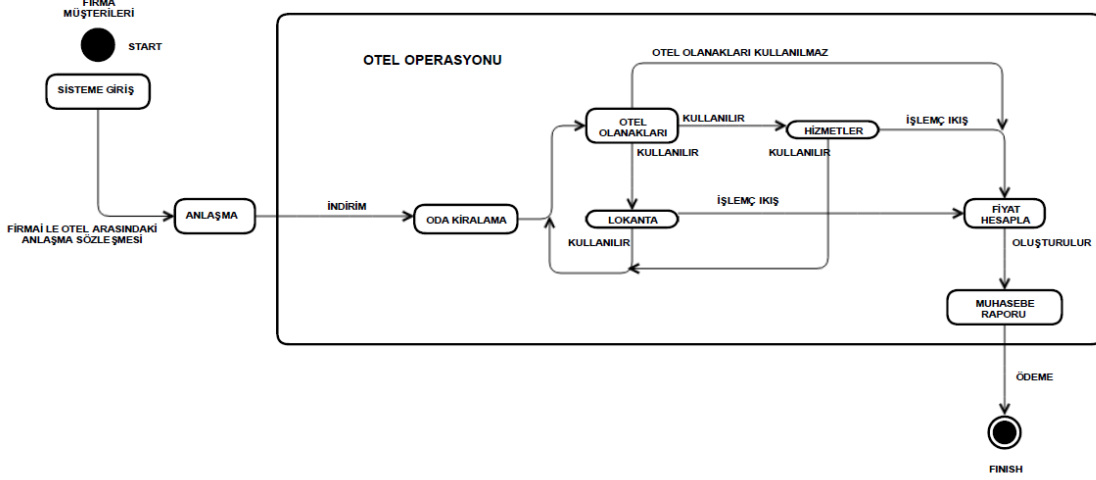
ŞEKİL 7: OTEL OTOMASYONU BİREYSEL MÜŞTERİ STATE MACHINE DİYAGRAMI

DIĞER MÜŞTERİLERİÇİN DURUM DİYAGRAMI



ŞEKİL 8: OTEL OTOMASYONU DIĞER MÜŞTERİ STATE MACHINE DİYAGRAMI

FİRMA MÜŞTERİLERİ İÇİN DURUM DİYAGRAMI



ŞEKİL 9: OTEL OTOMASYONU FİRMA MÜŞTERİSİ STATE MACHINE DİYAGRAMI

7. PLANLAMA VE SÜREÇ MODELİ

Projemizin senaryosunda tüm gereksinimler tam olarak anlatılmış ve açık kapı bırakılmamıştır. Tüm durumların cevapları senaryoda ve çizilen diyagramlarda bulunmaktadır.

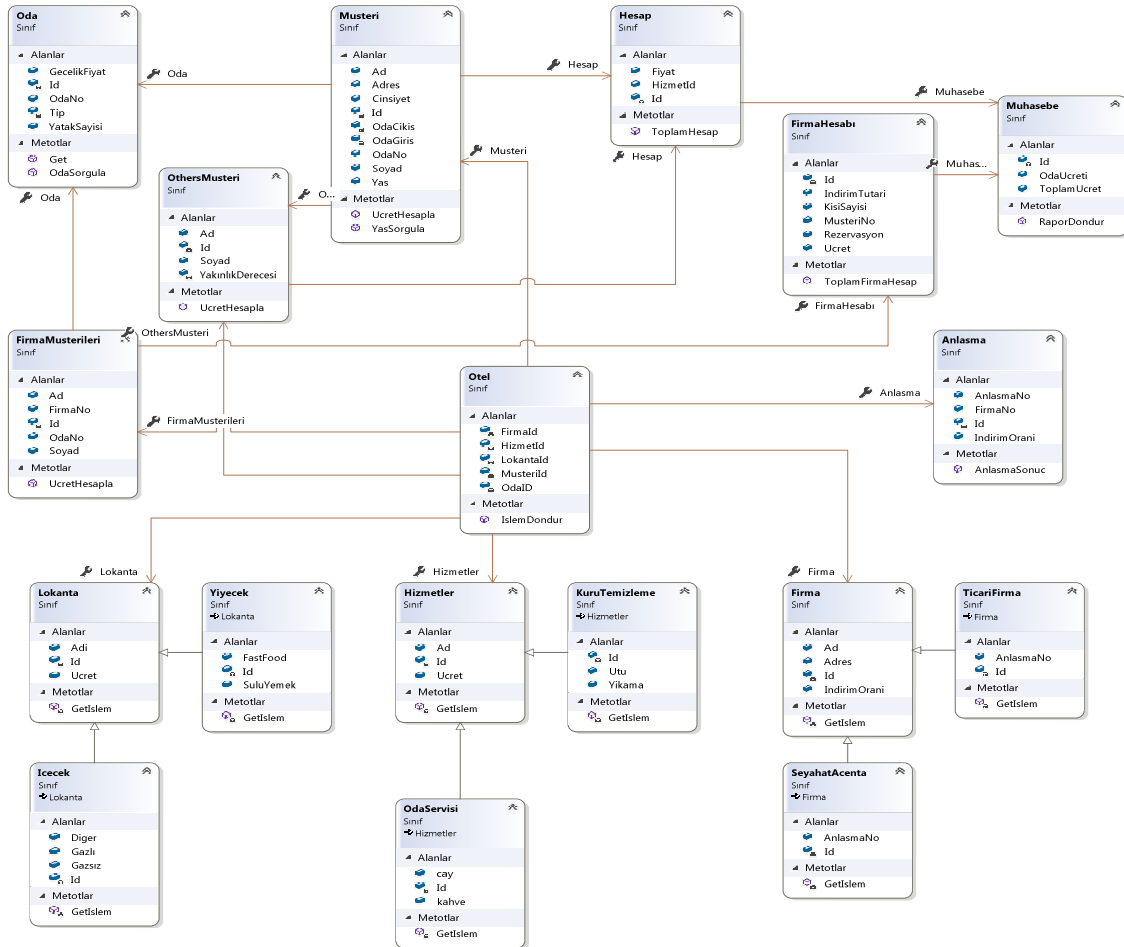
Bu durumda yazılım geliştirme sürecinde **ŞELELE MODELİ** kullanımını uygun bulduk .

Şelele modelinin yanında çevik süreçlerinde, proje geliştirme aşamasında kullanımı da uygundur.

Projemizde, **Planning** kısmında projemizde bulunan durumlar incelenir ve bundan sonra yazılım sürecinin planlaması yapılır. Daha sonra programın **Modelling** kısmına geçilir ve proje oluşturulmaya başlanır. Bundan sonra **Code** ve **Test** durumuna geçilir, son durumda ise **Destek** ve **Geri Dönüşler** sonucunda proje geliştirilmeye devam edilir.

8. TASARIM

Tasarım kısmında, sistemin tüm nesneleri belirlenip, bu nesnelere ait özellik ve metodlar belirlendi. Belirlenenlere göre Class diyagramı çizildi.



ŞEKİL 10: OTEL OTOMASYONU CLASS DİYAGRAMI

9. SİSTEMİN KODU

Sistemin kodu C# ile nesneye yönelik yazıldı. Ancak sunumda da belirttiğimiz gibi metodların içini tam anlamıyla oluşturamadık. Bunun sebebi projemizin masaüstü bir uygulama olacağını öngörmemize rağmen bir arayüz tasarımının yapılmamış olmasından kaynaklanmaktadır. Bu tasarımı yapacak süremiz olmadığından sadece oluşturduğumuz classlar ve metodlarıyla değerlendirilmesini temenni ediyoruz.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

9.1 Oda.cs

```
namespace OtelOtomasyonu
{
    public class Oda
    {
        public int OdaNo;
        public int YatakSayisi;
        private int Id;
        public double GecelikFiyat;
        private readonly String Tip;

        public void Get()
        {
            throw new System.NotImplementedException();
        }

        public void OdaSorgula()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.2 Anlasma.cs

```
namespace OtelOtomasyonu
{
    public class Anlasma
    {
        private int Id;
        public int FirmaNo;
        public int AnlasmaNo;
        public double IndirimOrani;

        public void AnlasmaSonuc()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.3 Otel.cs

```
namespace OtelOtomasyonu
{
    public class Otel
    {
        private int OdaID;
        private int HizmetId;
        private int LokantaId;
        private int MusteriId;
        private int FirmaId;

        public Lokanta Lokanta
        {
            get => default(Lokanta);
            set
            {
            }
        }

        public Hizmetler Hizmetler
        {
            get => default(Hizmetler);
            set
            {
            }
        }

        public Firma Firma
        {
            get => default(Firma);
            set
            {
            }
        }

        public Anlasma Anlasma
        {
            get => default(Anlasma);
            set
            {
            }
        }

        public Musteri Musteri
        {
            get => default(Musteri);
            set
            {
            }
        }

        public FirmaMusterileri FirmaMusterileri
        {
            get => default(FirmaMusterileri);
            set
            {
            }
        }

        public OthersMusteri OthersMusteri
        {
            get => default(OthersMusteri);
            set
            {
            }
        }

        public void IslemDondur()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.4 FirmaHesabi.cs

```
namespace OtelOtomasyonu
{
    public class FirmaHesabi
    {
        private int Id;
        public int MusteriNo;
        public int KisiSayisi;
        public DateTime Rezervasyon;
        public double Ucret;
        public double IndirimTutari;

        public Muhasebe Muhasebe
        {
            get => default(Muhasebe);
            set
            {
            }
        }

        public double ToplamFirmaHesap()
        {
            return Ucret * IndirimTutari;
        }
    }
}
```

9.5 FirmaMusterileri.cs

```
namespace OtelOtomasyonu
{
    public class FirmaMusterileri
    {
        private int Id;
        public int FirmaNo;
        public string Ad;
        public string Soyad;
        public int OdaNo;

        public Oda Oda
        {
            get => default(Oda);
            set
            {
            }
        }

        public FirmaHesabi FirmaHesabi
        {
            get => default(FirmaHesabi);
            set
            {
            }
        }

        public void UcretHesapla()
        {
        }
    }
}
```

9.6 Hesap.cs

```
namespace OtelOtomasyonu
{
    public class Hesap
    {
        private int Id;
        public double Fiyat;
        public int HizmetId;

        public Muhasebe Muhasebe
        {
            get => default(Muhasebe);
            set
            {
            }
        }

        public void ToplamHesap()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.7 Muhasebe.cs

```
namespace OtelOtomasyonu
{
    public class Muhasebe
    {
        private int Id;
        public int OdaUcreti;
        public double ToplamUcret;

        public void RaporDondur()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.8 Musteri.cs

```
namespace OtelOtomasyonu
{
    public class Muhasebe
    {
        private int Id;
        public int OdaUcreti;
        public double ToplamUcret;

        public void RaporDondur()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.9 OthersMusteri.cs

```
namespace OtelOtomasyonu
{
    public class OthersMusteri
    {
        private int Id;
        public string Ad;
        public string Soyad;
        private string YakınlıkDerecesi;

        public Hesap Hesap
        {
            get => default(Hesap);
            set
            {
            }
        }

        public void UcretHesapla()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.10 Lokanta.cs

```
namespace OtelOtomasyonu
{
    public class Lokanta
    {
        private int Id;
        public string Adi;
        public double Ucret;

        private void GetIslem()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.11 Yiyecek.cs

```
namespace OtelOtomasyonu
{
    public class Yiyecek : Lokanta
    {
        private int Id;
        public String SuluYemek;
        public string FastFood;

        private int GetIslem()
        {
            return Id;
        }
    }
}
```

9.12 Icecek.cs

```
namespace OtelOtomasyonu
{
    public class Icecek : Lokanta
    {
        private int Id;
        public string Gazlı;
        public string Gazsız;
        public string Diger;

        private void GetIslem()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.13 Hizmetler.cs

```
namespace OtelOtomasyonu
{
    public class Hizmetler
    {
        private int Id;
        public string Ad;
        public string Ucret;

        private void GetIslem()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.14 OdaServisi.cs

```
namespace OtelOtomasyonu
{
    public class OdaServisi : Hizmetler
    {
        private int Id;
        public int cay;
        public int kahve;

        private void GetIslem()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.15 KuruTemizleme.cs

```
namespace OtelOtomasyonu
{
    public class KuruTemizleme : Hizmetler
    {
        private int Id;
        public int Utu;
        public int Yikama;

        private int GetIslem()
        {
            return Id;
        }
    }
}
```

9.16 Firma.cs

```
namespace OtelOtomasyonu
{
    public class Firma
    {
        private int Id;
        public string Ad;
        public string Adres;
        public double IndirimOrani;

        private int GetIslem()
        {
            return Id;
        }
    }
}
```

9.17 SeyahatAcenta.cs

```
namespace OtelOtomasyonu
{
    public class SeyahatAcenta : Firma
    {
        private int Id;
        public int AnlasmaNo;

        private void GetIslem()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

9.18 TicariFirma.cs

```
namespace OtelOtomasyonu
{
    public class TicariFirma : Firma
    {
        private int Id;
        public int AnlasmaNo;

        private void GetIslem()
        {
            throw new System.NotImplementedException();
        }
    }
}
```


10. MALİYET

	a	b	c	d
Organic	2,4	1,05	2,5	0,38
Semi-detached	3	1,12	2,5	0,35
Embedded	3,6	1,2	2,5	0,32

$$MM = a (KLOC)^b$$

$$TDEV = c (MM)^d$$

ŞEKİL 11: COCOMO a,b,c,d Değerleri

Otel otomasyonu sistemimizde yaklaşık 10.000 satır kod olduğunu varsayıyoruz çünkü; büyük projelere göre daha basit bir yapıya sahiptir ve gereksinimlerde bunu göstermektedir.

Tüm bir otomasyon programı için KLOC=10 olarak alıyoruz. Ve basit bir proje olması sonucunda **Organic** olarak değerlendiriyoruz;

$$\text{Effort:MM} = 2,4(10)^{1,05} = \mathbf{26 \text{ man-months}}$$

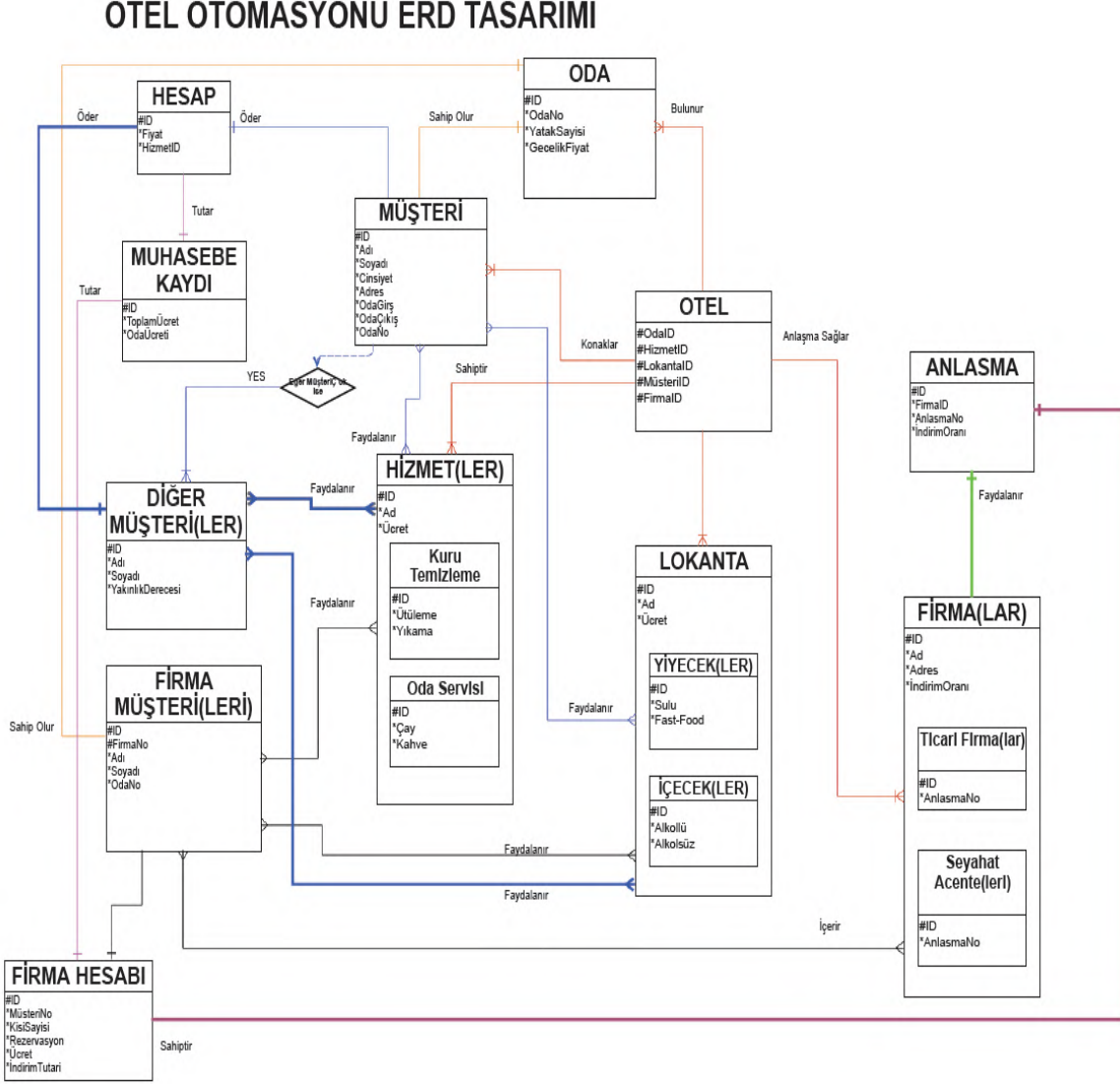
$$\text{Schedule:TDEV} = 2,5(26)^{0,38} = \mathbf{8 \text{ months}}$$

$$\text{Productivity:} 10000 \text{ DSI}/26 = \mathbf{384 \text{ DSI/MM}}$$

$$\text{Avarage Staffing: } 26 \text{ MM}/8 \text{ months} = \mathbf{3,25 \text{ (FSP= FULL-TIME-Equivalent Software Personnel)}}$$

11. VERİTABANI (BONUS)

Oluşturduğumuz yazılım sürecine ek olarak bu sistemin veri tabanı kısmında ER – Diyagramı da aşağıdaki gibidir.



ŞEKİL 12: OTEL OTOMASYONU ER Diyagramı

12. SONUÇ

Sonuç olarak, bu sistemin yazılımı yapılmadan önce izlenecek adımları daha iyi öğrenildi. Derste teorik olarak öğrendiğimiz konuları pratiğe dökerek aslında bir projede herşeyin koddan ibaret olmadığı sonucuna vardık. Tabi ki kod olmadan bir proje olmaz. Fakat o kodu en optimum düzeyde yazmak için çerçeve etkinlikleri, süreç modelleme, planlama, maliyet kesinlikle yapılmalıdır. Aksi takdirde ortaya çok fazla işlevi olmayan bir kod çıkar. Kod çalışsa bile daha sonradan bir yerinde sıkıntı çıksa bulunması zor olur. Uzun lafın kısıası arkamızdan yazdığımız kod ile ilgili kötü yorum yapılmasını engellemek için bu süreçlerin yapılması gerektiğini savunuyoruz. Saygılarımızla...

Ahmet Talha ÇAKIROĞLU

Mert Faik DELİ

KUBİLAY ÖZ

13.KAYNAKÇA

<http://oguzhan.menemencioglu.com/BLM426>

<https://www.gliffy.com/>

<https://fikirjeneratoru.com/>

<https://www.codex.com.tr>