

1	İÇİNDEKİLER	
2	TABLolar LİSTESİ	4
3	ŞEKİLLER LİSTESİ	5
4	HASTANE YÖNETİM SİSTEMİ	8
4.1	GİRİŞ	8
4.2	KULLANILAN TEKNOLOJİLER	8
4.2.1	Microsoft Visual Studio Ultimate 2013	8
4.2.2	Microsoft SQL Server 2012	8
4.2.3	Code First Entity Framework	9
4.2.4	DevExpress 14.2	9
4.2.5	Windows Communication Foundation	9
4.2.6	Windows Servis	9
4.2.7	Team Foundation Server	10
4.3	VERİ TABANI TASARIMI	11
4.3.1	VERİ TABANI TASARIMINDA ENTITY NESNELERİNİN OLUŞTURULMASI	11
4.3.2	ENTITY NESNELERİ İLE VERİ TABANI İLİŞKİSİ KURACAK OLAN CONTEXT SINIFININ OLUŞTURULMASI	12
4.3.3	ENTITY NESNELERİ SONUCU SUNUCU ÜZERİNDE OLUŞAN TABLO YAPILARI	13
4.3.3.1	Şehirler Tablosu	13
4.3.3.2	İlçeler Tablosu	13
4.3.3.3	Kan Grupları Tablosu	13
4.3.3.4	Unvanlar Tablosu	14
4.3.3.5	Departmanlar Tablosu	14

4.3.3.6	Hastalıklar Tablosu	15
4.3.3.7	Doktorlar Tablosu	15
4.3.3.8	Doktor Mailleri ve Doktor Telefonları Tabloları	15
4.3.3.9	Muayeneler Tablosu	16
4.3.3.10	Birimler Tablosu	16
4.3.3.11	Laboratuvar Tetkikler Listesi	16
4.3.3.12	Laboratuvar İstekleri Tablosu	17
4.3.3.13	Laboratuvar İstekleri Nesneleri Tablosu	17
4.3.3.14	İlaçlar Tablosu	17
4.3.3.15	Mesajlar Tablosu	18
4.3.3.16	Hastalar Tablosu	18
4.3.3.17	Personeller Tablosu	19
4.3.3.18	Reçeteler ve Reçete Elemanları Tabloları	19
4.3.3.19	Randevu Saatleri Tablosu	19
4.3.3.20	Randevular Tablosu	20
4.3.4	Veri Tabanı İçerisinde Yer Alan Yabancı Anahtarlar	20
4.4	VERİ ERİŞİM KATMANI (DATA ACCESS LAYER)	22
4.4.1	Abstract Klasörü	22
4.4.2	Content Klasörü	23
4.4.3	Migrations Klasörü	25
4.5	SUNUM ARA YÜZLERİ (INTERFACES)	27
4.6	İŞ KURALLARI KATMANI (BUSINESS LOGIC LAYER)	28
4.7	WINDOWS COMMUNICATION FOUNDATION	31
4.8	WINDOWS SERVICE	33
4.9	MASAÜSTÜ UYGULAMA GELİŞTİRME	34

4.9.1	GİRİŞ EKRANI	34
4.9.2	TÜM EKРАНLARDA ORTAK KOMUTLAR	34
4.9.2.1	BAĞLANTI AYARLARI	34
4.9.2.2	TEMA DEĞİŞTİR	35
4.9.2.3	TÜRKÇE DİL DOSYALARINI İNDİR	35
4.9.3	YÖNETİCİ EKRANI KOMUTLARI	35
4.9.3.1	DEPARTMAN VE UNVAN İŞLEMLERİ	35
4.9.3.2	ŞEHİR VE İLÇE İŞLEMLERİ	37
4.9.3.3	HASTALIKLAR VE İLAÇLAR EKRANI	39
4.9.3.4	DOKTOR İŞLEMLERİ EKRANI	41
4.9.3.5	TATİL İŞLEMLERİ EKRANI	43
4.9.4	PERSONEL EKRANI KOMUTLARI	44
4.9.4.1	HASTA İŞLEMLERİ EKRANI	44
4.9.4.2	RANDEVU İŞLEMLERİ EKRANI	46
4.9.5	DOKTOR EKRANI KOMUTLARI	48
4.9.5.1	MUAYENE İŞLEMLERİ	49
4.10	UYGULAMA DAĞITMADA ONCECLICK AYARLAMA	53
4.11	YAYINLANAN UYGULAMANIN BİLGİSAYARA KURULMASI	59
5	ÖZGEÇMİŞ	Error! Bookmark not defined.

2 TABLOLAR LİSTESİ

Tablo 6.3.1 Entity Sınıflarına Uygulanan Attributlar	11
Tablo 4.3.2 Tablolarda Kullanılan Yabancı Anahtarlar	21

3 ŞEKİLLER LİSTESİ

Şekil 4.3.1 Şehirler Tablosu	13
Şekil 4.3.2 İlçeler Tablosu	13
Şekil 4.3.3 Kan Grupları Tablosu	14
Şekil 4.3.4 Unvanlar Tablosu	14
Şekil 4.3.5 Departmanlar Tablosu	14
Şekil 4.3.6 Hastalıklar Tablosu	15
Şekil 4.3.7 Doktorlar Tablosu	15
Şekil 4.3.8 Doktor Mail ve Telefonları Tabloları	15
Şekil 4.3.9 Muayeneler Tablosu	16
Şekil 4.3.10 Birimler Tablosu	16
Şekil 4.3.11 Laboratuvar Tetkik Listesi	16
Şekil 4.3.12 Laboratuvar İstekleri Tablosu	17
Şekil 4.3.13 Laboratuvar İstekleri Nesneleri Tablosu	17
Şekil 4.3.14 İlaçlar Tablosu	17
Şekil 4.3.15 Mesajlar Tablosu	18
Şekil 4.3.16 Hastalar Tablosu	18
Şekil 4.3.17 Personeller Tablosu	19
Şekil 4.3.18 Reçete ve Reçete Elemanları Tabloları	19
Şekil 4.3.19 Randevu Saatleri Tablosu	19
Şekil 4.3.20 Randevular Tablosu	20
Şekil 4.4.1 DAL Katmanı Proje Yapısı	22
Şekil 4.9.1 Giriş Ekranı	34
Şekil 4.9.2 Kullanıcı Ayarları	34
Şekil 4.9.3 Servis Ayarları	34

Şekil 4.9.4 Tema Ayarı	35
Şekil 4.9.5 Departman ve Ünvan İşlemleri	35
Şekil 4.9.6 Departman Ekranı	36
Şekil 4.9.7 Ünvan Ekranı	36
Şekil 4.9.8 Şehir ve İlçe Ekranı	37
Şekil 4.9.9 Şehir Ekranı	38
Şekil 4.9.10 İlçe Ekranı	38
Şekil 4.9.11 Hastalık ve İlaç Ekranı	39
Şekil 4.9.12 Hastalık Ekranı	40
Şekil 4.9.13 İlaç Ekranı	40
Şekil 4.9.14 Doktor İşlemleri Ekranı	41
Şekil 4.9.15 Doktor Ekranı	42
Şekil 4.9.16 Tatil İşlemleri Ekranı	43
Şekil 4.9.17 Tatil Ekranı	43
Şekil 4.9.18 Hasta İşlemleri Ekranı	44
Şekil 4.9.19 Hasta Ekranı	45
Şekil 4.9.20 Randevu İşlemleri Ekranı	46
Şekil 4.9.21 Yeni Randevu Ekranı	47
Şekil 4.9.22 Randevu Silme Ekranı	48
Şekil 4.9.23 Muayene İşlemleri Ekranı	49
Şekil 4.9.24 Laboratuvar İsteği Ekranı	50
Şekil 4.9.25 Geçmiş Muayene ve Mesajlaşma Bilgileri	51
Şekil 4.9.26 Geçmiş Muayene Ayrıntısı	52
Şekil 4.9.27 Reçete Yazma Ekranı	53
Şekil 4.10.1 Sitenin Açılması	54

Şekil 4.10.2 Publish Menusu	55
Şekil 4.10.3 Application Files	56
Şekil 4.10.4 Prerequisites	57
Şekil 4.10.5 Application Updates	58
Şekil 4.11.1 Kurulum Sayfası	59

4 HASTANE YÖNETİM SİSTEMİ

4.1 GİRİŞ

Projenin geliştirilmesi aşamasında, daha sonra sistemin geliştirilebilir olması ve sistemde ortaya çıkabilecek hataların tespitinin en kolay şekilde olabilmesi için katmanlı mimari kullanıldı. Katmanlı mimaride amaç, her görevin kendine has katmanda yapılmasıdır. Örneğin veri tabanı işlemlerinin veri tabanı erişim katmanında, iş kurallarının denetiminin iş kural katmanında sorgulanması.

4.2 KULLANILAN TEKNOLOJİLER

4.2.1 Microsoft Visual Studio Ultimate 2013

Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework ve Microsoft Silverlight tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve Windows Forms uygulamaları, web siteleri, web uygulamaları ve web servisleri ile birlikte konsol ve grafiksel kullanıcı ara yüzü uygulamaları geliştirmek için kullanılır.

Visual Studio IntelliSense'in yanı sıra "code refactoring" destekleyen bir kod editörü içerir. Entegre hata ayıklayıcı, hem kaynak-seviyesinde hem de makine-seviyesinde çalışır. Diğer yerleşik araçlar, GUI uygulamaları, web tasarımcısı, sınıf tasarımcısı ve veri tabanı şema tasarımcısı yaratabilmek için bir form tasarımcısı içerir. Hemen hemen her işlevsellik düzeyinde dâhil olmak üzere, kaynak kontrol sistemleri için destek (Subversion ve Visual SourceSafe gibi) sunan eklentileri kabul eder.

Visual Studio, değişik programlama dillerini destekler, bu da kod editörü ve hata ayıklayıcısının neredeyse tüm programlama dillerini desteklemesini sağlamaktadır. Dâhili diller C/C++ (Görsel yoluyla C++), VB.NET (Visual Basic .NET üzerinden), C# (Visual C# ile), ve F# (Visual Studio 2010 itibarıyla) içermektedir.

4.2.2 Microsoft SQL Server 2012

SQL Server, Microsoft tarafından geliştirilmiş ilişkisel veri tabanı yönetim sistemidir. İlişkisel veri tabanı sistemi İngilizce olarak Relational Database Management System olarak adlandırılır ve kısaca RDBMS olarak bazı kaynaklarda karşımıza çıkmaktadır. İlişkisel veri tabanı sisteminde veriler tablolar halinde tutulurlar ve bu tablolar kendi aralarında ilişkisel anlamda birbiri ile bağlantılı olabilirler.

4.2.3 Code First Entity Framework

Entity Framework, .Net platformunda ORM (Object Relational Mapping) araçlarından biridir. ORM (Object Relational Mapping) ise veri tabanı ile nesneye yönelik programlama (OOP) arasındaki ilişkiyi kuran teknolojidir. Yani Entity Framework, nesne tabanlı programlamada veri tabanındaki tablolara uygun nesneler oluşturma tekniğidir. Projede kullandığımız Code First Entity mantığında ise önce nesneler kodlanır daha sonra bu nesnelere karşılık gelen veri tabanı, proje çalıştırıldığında kod tarafından verilen bir veri tabanı oluşturma komutu ile sunucu tarafında otomatik olarak oluşturulur.

4.2.4 DevExpress 14.2

.net platformunun bir componentidir. .Net platformu ile uygulama geliştirenler için görsel açıdan çok iyi düzenlemeler sağlar.

4.2.5 Windows Communication Foundation

Windows Communication Foundation farklı makineler üzerindeki uygulamaların birbiriyle iletişim kurmasını sağlayan bir iç sistemdir. .NET Framework 3.0 uygulama geliştirme platformunun bir bileşeni olarak sunulmuştur. WCF olarak kısaltılan bu teknoloji, ilk olarak Indigo kod adıyla dünyaya duyurulmuştur. WCF; Remoting, Web Servisleri, Dağıtık "Transaction"lar gibi dağıtık programlama yollarını tek bir yapıda birleştirmektedir.

WCF Servis odaklı mimari (Service Oriented Architecture) kapsamında dağıtık sistemler tasarlamak ve geliştirmek içindir. İstemciler birden fazla servis hizmetinden faydalanabilirler ve servisler birden fazla istemciler tarafından tüketilebilir. Düzgün arabirimlerle birleşen servisler yaratabilir ve bu servislerin birbirlerini minimum düzeyde etkileyecek şekilde sistemler tasarlanarak birimsellik özelliğinden faydalanılabilir. Servisler WSDL (Web Services Description Language) ara yüzünü kullanarak diğer servislerle mesajlaşırlar ve bu özellik servislerin başka platformlarda yaratılmış servislerle sorunsuz mesajlaşabilmesi sağlar. WCF WS-Addressing, WS-ReliableMessaging ve WS-Security standartları gibi birçok gelişmiş Web hizmetlerini uygular. .NET Framework 4.0 sürümü ile WCF da RSS Hizmetleri sağlar.

4.2.6 Windows Servis

Windows servisleri, işletim sisteminde arka planda çalışan, kullanıcı ile etkileşimde bulunduğu herhangi bir ara yüze sahip olmayan, kaynakların izlenmesi, system olaylarının log olarak tutulması, network erişimlerinin izlenmesi, veri tabanları üzerindeki transaction'ların izlenmesi, sistem performansına atı bilgilerin toplanması, sistem hatalarının (system exceptions) , başarısız program denemelerin (failure) vb. gibi geri plan işlemlerinin takip edilmesinde kullanılan, sisteme kayıt edilmiş (register), çalıştırılabilir nesnelerdir.

Aslında, Windows NT,2000,XP ya da 2003 kullanıcısı iseniz, Windows servisleri ile mutlaka ilgilenmişsinizdir. Sistemlerimizde çalışan pek çok servis vardır. Bu servislerin neler olduğuna, Administrative Tool bölümünde Services kısmından bakabiliriz.

4.2.7 Team Foundation Server

TFS; takım halinde çalışılan projelerde, takım içerisindeki iletişimi ve verimi artıracak, projelerin yönetimini kolaylaştıracak bir Microsoft ürünüdür.

Genel amacı iş birliği sağlamak ve ürün gelişimini kolaylaştırmaktır. TFS ile birlikte ortak sunucu 'da tutulan projeye tüm ortak proje üzerinde çalışanlar ulaşabilir ve işlerini buradan görerek diğer takım arkadaşlarının bilgisine sunabilmektedir. Yani farklı geliştirme ortamları üzerinden ekip halinde proje üzerinde çalışılması ve değişikliklerin herkese yansıtılması sağlanmaktadır.

TFS, projenizi düzen içerisinde, sağlıklı bir ortam sunarak geliştirmenizi sağlar. Program her zaman proje geliştiricilerine en güncel hali sunar ve istenildiği takdirde önceki sürümlere ulaşabilme imkânı sağlar.

Birçok dosyanın bulunduğu proje üzerinde değişiklik yapmak isteyen geliştirici 'Check Out' ismi verilen işlemle istediği değişiklikleri yapmak için çalışmak istediği dosya üzerinde bir bakıma çalışma izni alır. Bunu yapmaya başladığı andan itibaren diğer tüm geliştiriciler arkadaşlarının o dosya üzerinde çalıştığını görecektir. Ekip arkadaşları da bu duruma göre kendi yollarını izleyebilirler. Aynı şekilde dosya üzerinde çalışan kişi tüm işlerini bitirdikten sonra 'Check In' işlemi ile dosyayı tekrar sunucu ortamına geri bıraktığını bildirir ve dosyanın son hali ekibe sunulur. Bu aşamadan sonra ekip arkadaşları geliştiricinin yaptığı değişiklikleri gözlemleyebilirler.

4.3 VERİ TABANI TASARIMI

Veri tabanı tasarımında, kullanılan teknolojiler kısmında da söz edildiği gibi Code First Entity Framework yaklaşımı kullanıldı. Bu yaklaşım sonucu sunucu üzerinde oluşan tablolar ve açıklamaları aşağıda anlatıldığı şekildedir. Tablolar hakkında bilgi almak için kullanılan SQL sorgusu:

```
SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE,
CHARACTER_MAXIMUM_LENGTH, COLUMN_DEFAULT, IS_NULLABLE
FROM HealthSystem.INFORMATION_SCHEMA.COLUMNS WHERE
TABLE_NAME = 'TabloAdı'
```

4.3.1 VERİ TABANI TASARIMINDA ENTITY NESNELERİNİN OLUŞTURULMASI

Projemizde kullandığımız code first Entity framework mantığında veri tabanı sunucu üzerinde oluşturulmadan önce veri tabanında kullanılacak her bir tabloya karşılık gelecek olan sınıf yazılır. Sınıflar yazılmadan önce Entity Framework Kütüphanesi projeye referans verilmelidir. Bu sınıflar yazılırken özelliklerine bazı atributlar tanımlanır. Kullanılan atributlar ve özellikleri aşağıdaki tabloda verilmiştir. Aksi belirtilmediği sürece attribute nesne özelliğine uygulanır.

Tablo 4.3.1 Entity Sınıflarına Uygulanan Atributlar

Attribute	Özelliği
DataContract	System.Runtime.Serialization isim uzayında bulunur. Sınıfın WCF servisi üzerinden sunulacak bir nesne olduğunu belirtir. Sınıfa uygulanır.
KnownType	System.Runtime.Serialization isim uzayında bulunur. Nesnenin WCF üzerinden serileştirilmesini sağlar. Sınıfa uygulanır.
DataMember	Özelliğin WCF üzerinden dışarıya sunulacağını belirtir.
Required	Özelliğin “null” değer taşıyamayacağını belirtir.
MaxLength	String tipi için sahip olacağı max uzunluğu belirtir.

MinLength	String tipi için sahip olacağı min uzunluğu belirtir.
Index	Özelliğin veri tabanında bir indekse karşılık geldiğini belirtir.
DisplayName	Özelliğin ekranda görünen ismini belirtir.
DataType	Data kontrollerinde özelliğin tipini belirtir.

4.3.2 ENTİTY NESNELERİ İLE VERİ TABANI İLİŞKİSİ KURACAK OLAN CONTEXT SINIFININ OLUŞTURULMASI

Entity nesneleri ile veri tabanı ilişkisi kuracak olan sınıfımızdır. DbContext sınıfından türer. Veri tabanında yer alacak her bir tablomuz burada özellik olarak tanımlanır.

Örneğin Doctor tablosunun tanımlanması şu şekilde yapılır:

`public DbSet<Doctor> Doctors {get; set;} bu şekilde tanımlama yapıldıktan sonra veri tabanında Doctors adında bir tablomuz oluşacaktır. DbContext sınıfı, System.Data.Entity isim uzayında bulunmaktadır.`

Bu sınıf oluşturulurken ilerde hata yakalamalarını daha kolay yapmak için veri tabanı üzerinde işlem yapan “`int SaveChanges()`” metodu override edilebilir. Örneğin:

```
public override int SaveChanges()
{
    try { return base.SaveChanges(); }
    catch(Exception ex) { return -1; }
}
```

Bu şekilde override edildiği takdirde hata durumunda işlemi öldürmek yerine -1 sonucunu döndürür. Böylece her `SaveChanges()` metodu try-catch içine almaya gerek kalmayacaktır.

Ayrıca tablolar arası ilişkileri de `OnModelCreating(DbModelBuilder modelBuilder)` metodunu override edilerek belirtilir.

4.3.3 ENTİTY NESNELERİ SONUCU SUNUCU ÜZERİNDE OLUŞAN TABLO YAPILARI

4.3.3.1 Şehirler Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Cities	Id	int	NULL	NULL	NO
2	Cities	Name	nvarchar	30	NULL	NO
3	Cities	IsActive	bit	NULL	NULL	NO

Şekil 4.3.1 Şehirler Tablosu

Türkiye de yer alan 81 ilin listesini tutar.

4.3.3.2 İlçeler Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Countries	Id	int	NULL	NULL	NO
2	Countries	CityId	int	NULL	NULL	NO
3	Countries	Name	nvarchar	30	NULL	NO
4	Countries	IsActive	bit	NULL	NULL	NO

Şekil 4.3.2 İlçeler Tablosu

Türkiye de yer alan 958 ilçenin listesini tutar. Burada ki CityId kolonu ilçenin ait olduğu şehri tutar.

4.3.3.3 Kan Grupları Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	BloodGroups	Id	int	NULL	NULL	NO
2	BloodGroups	Name	nvarchar	7	NULL	NO
3	BloodGroups	IsActive	bit	NULL	NULL	NO

Şekil 4.3.3 Kan Grupları Tablosu

8 adet olan kan grupları listesini tutar

4.3.3.4 Unvanlar Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Degrees	Id	int	NULL	NULL	NO
2	Degrees	Name	nvarchar	30	NULL	NO
3	Degrees	ShortName	nvarchar	10	NULL	NO
4	Degrees	IsActive	bit	NULL	NULL	NO
5	Degrees	IsAdmin	bit	NULL	((0))	NO

Şekil 4.3.4 Unvanlar Tablosu

Sistem içerisinde kullanılan unvanların listesini tutar. Bazı unvan örnekleri: uzman doktor, müdür, kayıt personeli...

4.3.3.5 Departmanlar Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Departments	Id	int	NULL	NULL	NO
2	Departments	Name	nvarchar	25	NULL	NO
3	Departments	IsActive	bit	NULL	NULL	NO
4	Departments	MinAge	int	NULL	NULL	YES
5	Departments	MaxAge	int	NULL	NULL	YES
6	Departments	Gender	bit	NULL	NULL	YES

Şekil 4.3.5 Departmanlar Tablosu

Hastane içerisinde yer alan departmanların listesini tutar. Bazı departman örnekleri: dâhiliye, göz, çocuk hastalıkları... Burada yer alan Gender kolonu, departmana hangi cinsiyetten hastaların, MinAge ve MaxAge kolonları da ay cinsinden hangi yaş aralığında hastaların bu departmana başvurabileceğini belirtir. Her üç kolon için de “null” değer bir sınırlamanın olmadığını belirtir.

4.3.3.6 Hastalıklar Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Diagnosis	Id	int	NULL	NULL	NO
2	Diagnosis	Name	nvarchar	100	NULL	NO
3	Diagnosis	DepartmentId	int	NULL	NULL	NO
4	Diagnosis	IsActive	bit	NULL	NULL	NO

Şekil 4.3.6 Hastalıklar Tablosu

Doktor tarafından hastaya teşhisi konabilecek hastalık isimlerinin listesini tutar.

4.3.3.7 Doktorlar Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Doctors	Id	int	NULL	NULL	NO
2	Doctors	Name	nvarchar	50	NULL	NO
3	Doctors	Surname	nvarchar	30	NULL	NO
4	Doctors	BirthDay	datetime	NULL	NULL	NO
5	Doctors	TcNo	nvarchar	11	NULL	NO
6	Doctors	RegistrationNumber	nvarchar	11	NULL	NO
7	Doctors	DepartmentId	int	NULL	NULL	NO
8	Doctors	DegreeId	int	NULL	NULL	NO
9	Doctors	MotherName	nvarchar	50	NULL	NO
10	Doctors	FatherName	nvarchar	50	NULL	NO
11	Doctors	IsActive	bit	NULL	NULL	NO
12	Doctors	Password	nvarchar	16	(*)	NO

Şekil 4.3.7 Doktorlar Tablosu

Sisteme kayıtlı olan doktorların listesini tutar.

4.3.3.8 Doktor Mailleri ve Doktor Telefonları Tabloları

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	DoctorMails	Id	int	NULL	NULL	NO
2	DoctorMails	DoctorId	int	NULL	NULL	NO
3	DoctorMails	Mail	nvarchar	50	NULL	NO
4	DoctorMails	IsActive	bit	NULL	NULL	NO
5	DoctorPhones	Id	int	NULL	NULL	NO
6	DoctorPhones	DoctorId	int	NULL	NULL	NO
7	DoctorPhones	Phone	nvarchar	11	NULL	NO
8	DoctorPhones	IsActive	bit	NULL	NULL	NO

Şekil 4.3.8 Doktor Mail ve Telefonları Tabloları

Sisteme kayıtlı olan doktorların mail ve telefonları tutan iki ayrı tablodur.

4.3.3.9 Muayeneler Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Examinations	Id	int	NULL	NULL	NO
2	Examinations	PatientId	int	NULL	NULL	NO
3	Examinations	DoctorId	int	NULL	NULL	NO
4	Examinations	Time	datetime	NULL	NULL	NO
5	Examinations	DoctorNote	nvarchar	500	NULL	YES
6	Examinations	CanSendMessage	bit	NULL	NULL	NO
7	Examinations	NextTime	datetime	NULL	NULL	YES
8	Examinations	RendezvousId	int	NULL	NULL	NO
9	Examinations	IsActive	bit	NULL	NULL	NO
10	Examinations	DiagnosisId	int	NULL	NULL	NO

Şekil 4.3.9 Muayeneler Tablosu

Doktorlar tarafından yapılan muayenelerin listesini tutar.

4.3.3.10 Birimler Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	LaboratoryItemUnits	Id	int	NULL	NULL	NO
2	LaboratoryItemUnits	Name	nvarchar	10	NULL	NO
3	LaboratoryItemUnits	IsActive	bit	NULL	NULL	NO

Şekil 4.3.10 Birimler Tablosu

Laboratuvarda kullanılan birimlerin listesini tutar.

4.3.3.11 Laboratuvar Tetkikler Listesi

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	LaboratoryRequestTypes	Id	int	NULL	NULL	NO
2	LaboratoryRequestTypes	Name	nvarchar	20	NULL	NO
3	LaboratoryRequestTypes	Min	float	NULL	NULL	NO
4	LaboratoryRequestTypes	Max	float	NULL	NULL	NO
5	LaboratoryRequestTypes	IsActive	bit	NULL	NULL	NO
6	LaboratoryRequestTypes	LaboratoryItemUnitId	int	NULL	NULL	NO

Şekil 4.3.11 Laboratuvar Tetkik Listesi

Laboratuvar isteğinde istenebilecek tetkik türlerini tutan listedir.

4.3.3.12 Laboratuvar İstekleri Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	LaboratoryRequests	Id	int	NULL	NULL	NO
2	LaboratoryRequests	ExaminationId	int	NULL	NULL	NO
3	LaboratoryRequests	DateTime	datetime	NULL	NULL	NO
4	LaboratoryRequests	Note	nvarchar	-1	NULL	YES
5	LaboratoryRequests	IsActive	bit	NULL	NULL	NO

Şekil 4.3.12 Laboratuvar İstekleri Tablosu

Muayene sırasında doktor tarafından istenen laboratuvar isteklerini tutar.

4.3.3.13 Laboratuvar İstekleri Nesneleri Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	LaboratoryRequestItems	Id	int	NULL	NULL	NO
2	LaboratoryRequestItems	LaboratoryRequestId	int	NULL	NULL	NO
3	LaboratoryRequestItems	LaboratoryRequestTypeId	int	NULL	NULL	NO
4	LaboratoryRequestItems	Result	float	NULL	NULL	YES

Şekil 4.3.13 Laboratuvar İstekleri Nesneleri Tablosu

Doktor tarafından istenen laboratuvar tetkiklerinin istenen türlerini tutar.

4.3.3.14 İlaçlar Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Medicaments	Id	int	NULL	NULL	NO
2	Medicaments	Name	nvarchar	200	NULL	NO
3	Medicaments	UsePerDay	float	NULL	NULL	YES
4	Medicaments	Note	nvarchar	500	NULL	YES
5	Medicaments	IsActive	bit	NULL	NULL	NO

Şekil 4.3.14 İlaçlar Tablosu

Sisteme tanıtılmış olan ilaçlar listesini tutar.

4.3.3.15 Mesajlar Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Messages	Id	int	NULL	NULL	NO
2	Messages	DoctorId	int	NULL	NULL	NO
3	Messages	PatientId	int	NULL	NULL	NO
4	Messages	DateTime	datetime	NULL	NULL	NO
5	Messages	Message	nvarchar	500	NULL	YES
6	Messages	FromDoctor	bit	NULL	NULL	NO
7	Messages	Readed	bit	NULL	NULL	NO
8	Messages	IsActive	bit	NULL	NULL	NO

Şekil 4.3.15 Mesajlar Tablosu

Doktor ve hastaları arasında geçebilecek mesajları tutan tablodur.

4.3.3.16 Hastalar Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Patients	Id	int	NULL	NULL	NO
2	Patients	TcNo	nvarchar	11	NULL	NO
3	Patients	Name	nvarchar	50	NULL	NO
4	Patients	Surname	nvarchar	50	NULL	NO
5	Patients	Address	nvarchar	100	NULL	NO
6	Patients	CityId	int	NULL	NULL	NO
7	Patients	CountryId	int	NULL	NULL	NO
8	Patients	Phone	nvarchar	15	NULL	NO
9	Patients	Mail	nvarchar	100	NULL	NO
10	Patients	BirthDay	datetime	NULL	NULL	NO
11	Patients	Gender	bit	NULL	NULL	NO
12	Patients	BloodGroupId	int	NULL	NULL	NO
13	Patients	IsActive	bit	NULL	NULL	NO
14	Patients	Password	nvarchar	16	(*)	NO
15	Patients	FatherName	nvarchar	50	(*)	NO
16	Patients	MotherName	nvarchar	50	(*)	NO

Şekil 4.3.16 Hastalar Tablosu

Sisteme kayıtlı hastaların listesini tutar.

4.3.3.17 Personeller Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Personnels	Id	int	NULL	NULL	NO
2	Personnels	TcNo	nvarchar	11	NULL	NO
3	Personnels	Name	nvarchar	50	NULL	NO
4	Personnels	Surname	nvarchar	50	NULL	NO
5	Personnels	Address	nvarchar	100	NULL	NO
6	Personnels	CityId	int	NULL	NULL	NO
7	Personnels	CountryId	int	NULL	NULL	NO
8	Personnels	Phone	nvarchar	15	NULL	NO
9	Personnels	Mail	nvarchar	100	NULL	NO
10	Personnels	BirthDay	datetime	NULL	NULL	NO
11	Personnels	Gender	bit	NULL	NULL	NO
12	Personnels	IsActive	bit	NULL	NULL	NO
13	Personnels	Password	nvarchar	16	NULL	NO
14	Personnels	DegreeId	int	NULL	((0))	NO

Şekil 4.3.17 Personeller Tablosu

Sisteme kayıtlı doktor ve hastalar dışındaki kişileri tutar.

4.3.3.18 Reçeteler ve Reçete Elemanları Tabloları

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Prescriptions	Id	int	NULL	NULL	NO
2	Prescriptions	ExaminationId	int	NULL	NULL	NO
3	Prescriptions	DoctorNote	nvarchar	500	NULL	YES
4	Prescriptions	IsActive	bit	NULL	NULL	NO
5	PrescriptionItems	Id	int	NULL	NULL	NO
6	PrescriptionItems	PrescriptionId	int	NULL	NULL	NO
7	PrescriptionItems	MedicamentId	int	NULL	NULL	NO
8	PrescriptionItems	UsePerDay	float	NULL	NULL	NO
9	PrescriptionItems	DoctorNote	nvarchar	500	NULL	YES
10	PrescriptionItems	IsActive	bit	NULL	NULL	NO

Şekil 4.3.18 Reçete ve Reçete Elemanları Tabloları

Muayene sonrası doktor tarafından yazılan reçete ve bu reçetelere dâhil olan ilaçları tutan iki ayrı tablodur.

4.3.3.19 Randevu Saatleri Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	RendezvousTimes	Id	int	NULL	NULL	NO
2	RendezvousTimes	StartTime	time	NULL	NULL	NO
3	RendezvousTimes	FinishTime	time	NULL	NULL	NO
4	RendezvousTimes	IsActive	bit	NULL	NULL	NO

Şekil 4.3.19 Randevu Saatleri Tablosu

Gün içerisinde verilebilecek randevu saatlerinin listesini tutar.

4.3.3.20 Randevular Tablosu

	TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	COLUMN_DEFAULT	IS_NULLABLE
1	Rendezvous	Id	int	NULL	NULL	NO
2	Rendezvous	Date	datetime	NULL	NULL	NO
3	Rendezvous	IsActive	bit	NULL	NULL	NO
4	Rendezvous	RendezvousTimeId	int	NULL	NULL	NO
5	Rendezvous	PatientId	int	NULL	NULL	NO
6	Rendezvous	DoctorId	int	NULL	NULL	NO

Şekil 4.3.20 Randevular Tablosu

Web ortamında hasta tarafından alınan veya hastanede personel tarafından verilen randevuları tutar.

4.3.4 Veri Tabanı İçerisinde Yer Alan Yabancı Anahtarlar

Veri tabanında veri bütünlüğünün sağlanması adına çeşitli yabancı anahtarlar kullanılmıştır. Kullanılan yabancı anahtarların listesi aşağıdaki tabloda yer almaktadır. Veri tabanı içerisinde yer alan yabancı anahtarları listelemek için kullanılan SQL sorgusu şu şekildedir:

```
SELECT KF.TABLE_NAME [Referans Alan Tablo]
, KF.COLUMN_NAME [Referans Alan Kolon]
, KP.TABLE_NAME [Referans Tablo]
, KP.COLUMN_NAME [Referans Kolon]
FROM INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS RC
JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE KF ON
RC.CONSTRAINT_NAME = KF.CONSTRAINT_NAME
JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE KP ON
RC.UNIQUE_CONSTRAINT_NAME = KP.CONSTRAINT_NAME
```

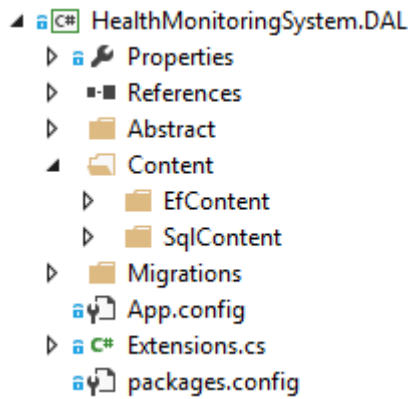
Tablo 4.3.2 Tablolarda Kullanılan Yabancı Anahtarlar

Referans Alan Tablo	Referans Alan Kolon	Referans Tablo	Referans Kolon
Patients	BloodGroupId	BloodGroups	Id
Countries	CityId	Cities	Id
Patients	CityId	Cities	Id
Personnels	CityId	Cities	Id
Patients	CountryId	Countries	Id
Personnels	CountryId	Countries	Id
Doctors	DegreeId	Degrees	Id
Personnels	DegreeId	Degrees	Id
Diagnosis	DepartmentId	Departments	Id
Doctors	DepartmentId	Departments	Id
Examinations	DiagnosisId	Diagnosis	Id
DoctorMails	DoctorId	Doctors	Id
DoctorPhones	DoctorId	Doctors	Id
Examinations	DoctorId	Doctors	Id
Messages	DoctorId	Doctors	Id
Rendezvous	DoctorId	Doctors	Id
LaboratoryRequests	ExaminationId	Examinations	Id
Prescriptions	ExaminationId	Examinations	Id
LaboratoryRequestTypes	LaboratoryItemUnitId	LaboratoryItemUnits	Id
LaboratoryRequestItems	LaboratoryRequestId	LaboratoryRequests	Id
LaboratoryRequestItems	LaboratoryRequestTypeId	LaboratoryRequestTypes	Id
PrescriptionItems	MedicamentId	Medicaments	Id
Consultations	PatientId	Patients	Id
Examinations	PatientId	Patients	Id
Messages	PatientId	Patients	Id
Rendezvous	PatientId	Patients	Id
PrescriptionItems	PrescriptionId	Prescriptions	Id
Examinations	RendezvousId	Rendezvous	Id
Rendezvous	RendezvousTimeId	RendezvousTimes	Id

4.4 VERİ ERİŞİM KATMANI (DATA ACCESS LAYER)

Sistemin veri tabanı işlemlerinin gerçekleştirildiği katmandır. Bu katmanda mümkün olduğunca şartlı işlemlerden arındırılmış şekilde veri tabanı işlemleri yapılır. Şartlı işlemlere bir örnek verilecek olursa: çocuk hastalıkları bölümüne 240 aylıktan büyük hastaların gelememesi veya kadın hastalıkları bölümüne erkek hastaların gelememesi gibi şartlar. Bunlara benzer şartlar BLL katmanında kontrol edilen şartlardır.

DAL katmanının yapısını açıklayacak olursak,



Şekil 4.4.1 DAL Katmanı Proje Yapısı

Proje içerisinde üç adet ana klasör bulunur. Bunlar: Abstract, Content ve Migrationdır.

4.4.1 Abstract Klasörü

İçerisinde interfaceri barındıran klasördür. Her entitye (tablolara karşılık gelen class) ait veri tabanı metotlarını barındıran bir interface oluşturulur. Bu interfacerin oluşturulmasının amacı ilerde Content klasöründe de anlatılacak olan veri tabanı yönetim sistemi üzerinde işlemleri yapacak olan hangi classın çağrılacağını belirlemektir.

Bu klasör içinde dosyaların isimlendirme kuralı şöyledir: I[Entity Adı]DAL

Şimdi de bir dosya üzerinde yapıyı daha yakından inceleyelim:

```
using System.Collections.Generic;
using HealthMonitoringSystem.Entity;

namespace HealthMonitoringSystem.DAL.Abstract
{
```

```

public interface IDepartmentDAL
{
    Department Select(int id);
    Department Select(string name);
    List<Department> Departments(bool? isActive = true, bool includeDoctors = true);
    Extensions.DataBaseResult Insert(Department newDepartment);
    Extensions.DataBaseResult Update(Department newInfoDepartment);
    Extensions.DataBaseResult Delete(int id);
}
}

```

Bu kod yapısında da görüldüğü gibi interface adı IDepartmentDAL'dır. İsimlendirme kurallarından yola çıkarak şu çözümlemeyi rahatlıkla yapabiliriz: bu interface DAL katmanında Department classına aittir. Ve içerisinde Department classına ait DAL katmanını metotlarının tanımlamalarını içerir.

4.4.2 Content Klasörü

Bu klasör içerisinde de kullanılabilecek her bir veri tabanı teknolojisi için bir alt klasör açılır ve bu klasörler içerisinde de isimlendirme kuralı şöyledir:

[Teknoloji Adı Kısaltması][Entity Adı]DAL

Her bir Entity için oluşturulacak olan class kendi entitysinin interfacesinden türetilmek zorundadır.

Bu klasör içerisinde oluşturulan klasörlerin amacı da interfacelerde tanımlamaları yapılan metotların içinin doldurulmasıdır yani işlem kodlarının yazılmasıdır. Şimdi de bu klasör içinde yer alan bir dosya üzerinden yapıyı yakından inceleyelim:

```

namespace HealthMonitoringSystem.DAL.Content.EfContent
{
    public class EfDepartmentDAL :IDepartmentDAL
    {
        public Department Select(int id)
        {
            using (HealthContext ctx = new HealthContext())
            {
                return ctx.Departments.FirstOrDefault(d => d.Id == id);
            }
        }

        public Department Select(string name)
        {
            using (HealthContext ctx = new HealthContext())
            {
                return ctx.Departments.FirstOrDefault(d => d.Name == name);
            }
        }
    }
}

```

```

    }

    public List<Department> Departments(bool? isActive = true, bool includeDoctors
= true)
    {
        using (HealthContext ctx = new HealthContext())
        {
            return ...
        }
    }

    public Extensions.DataBaseResult Insert(Department newDepartment)
    {
        using (HealthContext ctx = new HealthContext())
        {
            if (ctx.Departments.Any(d=>d.Name==newDepartment.Name))
            {
                return Extensions.DataBaseResult.AlreadyFound;
            }

            ctx.Departments.Add(newDepartment);

            return ctx.SaveChanges() > -1 ? Extensions.DataBaseResult.Success :
Extensions.DataBaseResult.Error;
        }
    }

    public Extensions.DataBaseResult Update(Department newInfoDepartment)
    {
        using (HealthContext ctx = new HealthContext())
        {
            if (ctx.Departments.Any(d => d.Name == newInfoDepartment.Name && d.Id
!= newInfoDepartment.Id))
            {
                return Extensions.DataBaseResult.AlreadyFound;
            }

            Department department =
ctx.Departments.FirstOrDefault(d=>d.Id==newInfoDepartment.Id);
            if (department == null)
            {
                return Extensions.DataBaseResult.NotFound;
            }

            ...

            return ctx.SaveChanges() > -1 ? Extensions.DataBaseResult.Success :
Extensions.DataBaseResult.Error;
        }
    }

```



```

    }

    public Extensions.DataBaseResult Delete(int id)
    {
        using (HealthContext ctx = new HealthContext())
        {
            Department department = ctx.Departments.FirstOrDefault(d => d.Id == id);

            if (department == null)
            {
                return Extensions.DataBaseResult.NotFound;
            }

            bool b = ctx.Doctors.Any(d => d.DepartmentId == id) || ctx.Diagnoses.Any(d
=> d.DepartmentId == id);

            if (b)
            { return Extensions.DataBaseResult.Referenced; }

            ctx.Departments.Remove(department);

            return ctx.SaveChanges() > -1 ? Extensions.DataBaseResult.Success :
Extensions.DataBaseResult.Error;
        }
    }
}

```

public class EfDepartmentDAL :IDepartmentDAL satırında da görüldüğü gibi Department classı için oluşturulan bu class kendi interfacesinden türetilmiştir ve interfacenin içerdiği bütün metotları içermek zorundadır. Böylece Department işlemlerinin yapan hangi teknoloji kullanılırsa kullansın ortak metotları bulundurulur. Class ismini de isimlendirme kuralına göre inceleyecek olursak EfDepartmentDAL classının: teknoloji olarak Entity Framework (Ef) kullandığını ve Department işlemlerini yaptığını ve DAL katmanında bulunduğunu anlayabiliriz.

Burada dikkat edilmesi gereken nokta: Insert, Update ve Delete metotlarında kullanılan şartlar ve dönüş tipidir. DAL katmanında mümkün olduğunca şartlı işlemlerden kaçınılmasına rağmen veri tabanı işlemlerinde hataya sebep olabilecek yerlerde şartlar kullanılır.

4.4.3 Migrations Klasörü

Projemizde kullandığımız code first Entity framework yapısından gelen bir özel klasördür. Bu klasörün içerisinde yapılan işlemler değişen Entity yapısına göre veri

tabanı yapısını güncellemektir. Veri tabanını otomatik güncellemek için yapılması gereken işlemler şöyledir:

Öncelikle Tools > Nuget Package Manager > Package Manager Console ekranından “Enable-Migrations –EnableAutomaticMigrations” komutu yazılır. Daha sonra entity yapısında bir değişiklik olduğunda aynı ekrandan “update-database –verbose” komutu yazılır. Burada önemli olan nokta bu güncelleme sonrasında veri kaybı söz konusu ise güncelleme işleminin varsayılan değer olarak hata ile sonuçlanacağıdır. Veri kaybı olması durumunda da güncellemeyi zorlamak istiyorsak, Migrations klasöründeki Configuration sınıfının apıcı metoduna şu komut girilir
“AutomaticMigrationDataLossAllowed = true”.

4.5 SUNUM ARA YÜZLERİ (INTERFACES)

BLL katmanını WCF servisi üzerinden dışarıya açmak için kullanılan katmandır. Her BLL sınıfı kendisi için oluşturulmuş interfaceden türetilmek zorundadır. Interface yazılırken iki temel attribute kullanılmaktadır.

Sınıfın bir servis sınıfı olduğunu belirtmek için “ServiceContract” attributu ve metodun da bir servis metodu olduğunu belirtmek için “OperationContract” attributu kullanılır.

Örnek bir dosya:

```
using System.Collections.Generic;
using System.ServiceModel;
using HealthMonitoringSystem.Entity;

namespace HealthMonitoringSystem.Interfaces
{
    [ServiceContract]
    [ServiceKnownType(typeof (City))]
    public interface ICitySol
    {
        [OperationContract]
        City Select(int id);

        [OperationContract]
        List<City> Cities(bool? isActive = true);

        [OperationContract]
        ProcessResult Insert(City newCity);

        [OperationContract]
        ProcessResult Update(City newInfoCity);

        [OperationContract]
        ProcessResult Delete(int id);
    }
}
```

Burada “OperationContract” attributunu taşımayan metod servis üzerinden dışarıya aktarılmayacaktır.

4.6 İŞ KURALLARI KATMANI (BUSINESS LOJIC LAYER)

Gelen giden verilerin kurallara uygunluğunun denetlendiği katmandır. Daha önce DAL katmanında söz ettiğimiz gibi DAL katmanında veri tabanı işlemleri kurallardan arındırılmış şekilde yapıliyordu. DAL katmanında pas geçilen kural denetimleri BLL katmanında uygulanmaktadır. Örneğin kişi kayıtlarında isim alanının boş bırakılamaması veya randevu işlemlerinde erkek hastaların kadın doğum polikliniğine gidememesi gibi. Bu katmanda oluşturulan her bir sınıf kendisi için oluşturulmuş interfaceden türetilmek zorundadır.

İsimlendirme kuralı: [Sınıf Adı]Manager

Örnek bir BLL dosyası:

```
using System;
using System.Collections.Generic;
using HealthMonitoringSystem.DAL.Abstract;
using HealthMonitoringSystem.Entity;
using HealthMonitoringSystem.Interfaces;

namespace HealthMonitoringSystem.BLL
{
    public class CityManager:ICitySol
    {
        private ICityDAL _cityDal;
        public CityManager(ICityDAL cityDal)
        {
            _cityDal = cityDal;
        }

        public City Select(int id)
        {
            return _cityDal.Select(id);
        }

        public List<City> Cities(bool? isActive = true)
        {
            return _cityDal.Citys();
        }

        public ProcessResult Insert(City newCity)
        {
            ProcessResult result = VerifeyModel(newCity);

            if (result.Result != Extensions.BLLResult.Verified)
            {
                return result;
            }

            DAL.Extensions.DataBaseResult insert = _cityDal.Insert(newCity);

            switch (insert)
            {
                case DAL.Extensions.DataBaseResult.AlreadyFound:
                    result.Result = Extensions.BLLResult.AlreadyFound;
            }
        }
    }
}
```

```

result.Errors.Add(Extensions.AlreadyFoundString(newCity.Name));
    break;
    case DAL.Extensions.DataBaseResult.Error:
        result.Result = Extensions.BLLResult.InnerException;
        result.Errors.Add(Extensions.InnerException);
        break;
    case DAL.Extensions.DataBaseResult.Success:
        result.Result = Extensions.BLLResult.Success;
        result.Errors.Add(Extensions.SuccessProcess);
        break;
    case DAL.Extensions.DataBaseResult.ServerDisable:
        result.Result = Extensions.BLLResult.ServerDisable;
        result.Errors.Add(Extensions.ServerDisable);
        break;
}

return result;
}

public ProcessResult Update(City newInfoCity)
{
    ProcessResult result = VerifeyModel(newInfoCity);

    if (result.Result != Extensions.BLLResult.Verified)
    {
        return result;
    }

    DAL.Extensions.DataBaseResult update = _cityDal.Update(newInfoCity);

    switch (update)
    {
        case DAL.Extensions.DataBaseResult.AlreadyFound:
            result.Result = Extensions.BLLResult.AlreadyFound;

result.Errors.Add(Extensions.AlreadyFoundString(newInfoCity.Name));
            break;
        case DAL.Extensions.DataBaseResult.Error:
            result.Result = Extensions.BLLResult.InnerException;
            result.Errors.Add(Extensions.InnerException);
            break;
        case DAL.Extensions.DataBaseResult.Success:
            result.Result = Extensions.BLLResult.Success;
            result.Errors.Add(Extensions.SuccessProcess);
            break;
        case DAL.Extensions.DataBaseResult.ServerDisable:
            result.Result = Extensions.BLLResult.ServerDisable;
            result.Errors.Add(Extensions.ServerDisable);
            break;
    }

    return result;
}

public ProcessResult Delete(int id)
{

```

```

        ProcessResult result = new ProcessResult();
        DAL.Extensions.DataBaseResult delete = _cityDal.Delete(id);
        switch (delete)
        {
            case DAL.Extensions.DataBaseResult.Success:
                result.Result = Extensions.BLLResult.Success;
                result.Errors.Add(Extensions.SuccessProcess);
                break;
            case DAL.Extensions.DataBaseResult.Referanced:
                result.Result = Extensions.BLLResult.Referanced;
                result.Errors.Add("İl başka tablolar üzerinde refere
edildiğinden silinemedi");
                break;
            case DAL.Extensions.DataBaseResult.NotFound:
                result.Result = Extensions.BLLResult.NotFound;
                result.Errors.Add(Extensions.NotFound);
                break;
            case DAL.Extensions.DataBaseResult.Error:
                result.Result = Extensions.BLLResult.InnerException;
                result.Errors.Add(Extensions.InnerException);
                break;
            case DAL.Extensions.DataBaseResult.ServerDisable:
                result.Result = Extensions.BLLResult.ServerDisable;
                result.Errors.Add(Extensions.ServerDisable);
                break;
        }

        return result;
    }

    ProcessResult VerifeyModel(City c)
    {
        ProcessResult result = new ProcessResult();

        if (c.Name.Length == 0)
        {
            result.Errors.Add("Ad kısmı boş bırakılamaz");
        }

        result.Result = result.Errors.Count == 0
            ? Extensions.BLLResult.Verified
            : Extensions.BLLResult.NotVerified;

        return result;
    }
}

```

Burada dikkat edilmesi gereken nokta, manager sınıfının DAL katmanından ilgili sınıfı yapıcı metoduna parametre almasıdır. Aldığı parametreyi gene DAL katmanın interfacesine atar. Burada kalıtım kuralının türeyen nesnenin ana nesneye atanabilme özelliğinden faydalanılır. Böylece parametre olarak DAL katmanının Contract sınıflarından hangisinin nesnesini parametre olarak almışsa onun üzerinden veri tabanı işlemlerini yapar.

4.7 WINDOWS COMMUNICATION FOUNDATION

BLL katmanının web üzerinden dışarıya açılmasını sağlayan katmandır. Bu katman sayesinde sistemin sadece grafik ara yüzünü geliştirmek isteyen şahsa sadece ilgili BLL sınıfının dışarıya açılmasını sağlayan linki vermemiz yeterli olacaktır.

Bu katmanın geliştirilmesi sırasında karşılaşılan sorun şudur: BLL parametre alan bir sınıfsken web üzerinden yapılan istekler parametresiz olarak çağrılır. Bu da hataya sebep olur. Bunun çözümü için kod injection yapılır. Bunun için projeye “Ninject” kütüphanesi referans olarak verilir. Daha sonra WCF servisin Global.asax sınıfı aşağıdaki gibi düzenlenir.

```
public class Global : NinjectHttpApplication
{
    protected override IKernel CreateKernel()
    {
        return new StandardKernel(new ServiceModule());
    }
}
```

Daha sonra projeye ServiceModule.cs adında bir dosya eklenir ve içi şöyle düzenlenir:

```
public class ServiceModule : NinjectModule
{
    /// <summary>
    /// Loads the module into the kernel.
    /// </summary>
    public override void Load()
    {
        this.Bind<IBloodGroupDAL>().To<EfBloodGroupDAL>();
        this.Bind<ICityDAL>().To<EfCityDAL>();
        ///...
        this.Bind<IPrescriptionDAL>().To<EfPrescriptionDAL>();
        this.Bind<IRendezvousDAL>().To<EfRendezvousDAL>();
        this.Bind<IRendezvousTimeDAL>().To<EfRendezvousTimeDAL>();
        this.Bind<IPersonnelDAL>().To<EfPersonnelDAL>();
        this.Bind<IHolidayDAL>().To<EfHolidayDAL>();
    }
}
```

Böylece hangi interface istendiğinde hangi sınıfın bir örneğinin verileceği belirtilir. Ve son olarak da web.config dosyasında servisler şöyle tanımlanır.

```
<serviceHostingEnvironment aspNetCompatibilityEnabled="true"
multipleSiteBindingsEnabled="true">
    <serviceActivations>
        <add service="HealthMonitoringSystem.BLL.BloodGroupManager"
relativeAddress="BloodGroupService.svc"
factory="Ninject.Extensions.Wcf.NinjectServiceHostFactory" />
    </serviceActivations>
</serviceHostingEnvironment>
```

```
        <add service="HealthMonitoringSystem.BLL.CityManager"
relativeAddress="CityService.svc"
factory="Ninject.Extensions.Wcf.NinjectServiceHostFactory" />
<--
        <add service="HealthMonitoringSystem.BLL.DepartmentManager"
relativeAddress="DepartmentService.svc"
factory="Ninject.Extensions.Wcf.NinjectServiceHostFactory" />

    </serviceActivations>

</serviceHostingEnvironment>
```


4.8 WINDOWS SERVICE

Windows servis arka tarafta çalışan küçük programcıklardır. Projede kullanılmasının amacı: arka tarafta otomatik olarak çalışması gereken görevleri yerine getirmek. Bu işlemlere örnek verilecek olursa zamanı geçmiş ama muayeneye gelmeyen randevuların sistemden silinmesi veya gelen randevu bilgilerinin hastaya mail veya sms yoluyla iletilmesi.

4.9 MASAÜSTÜ UYGULAMA GELİŞTİRME

4.9.1 GİRİŞ EKRANI

The screenshot shows a login window titled "Giriş Yap" with a red heart icon and a close button (X). It contains three input fields: "Kimlik Numarası" (Identity Number), "Şifre" (Password), and "Hesap Türü" (Account Type) which is a dropdown menu currently set to "Doktor". Below the fields are two buttons: "İptal" (Cancel) and "Giriş Yap" (Login).

Şekil 4.9.1 Giriş Ekranı

Sisteme girişte kimlik numarası ve şifrenin sorulduğu ekrandır. Hesap türü olarak “Doktor, Personel ve Yönetici” olmak üzere 3 adet seçenek bulunmaktadır.

4.9.2 TÜM EKRANLARDA ORTAK KOMUTLAR

The screenshot shows a settings menu titled "Kullanıcı Ayarları" with a blue person icon. It lists four options: "Bağlantı Ayarları" (Connection Settings) with a gear icon, "Tema Değiştir" (Change Theme) with a paintbrush icon, and "Türkçe Dil Dosyaları İndir" (Download Turkish Language Files) with a green ABC icon. There is also an upward arrow icon next to the title.

Şekil 4.9.2 Kullanıcı Ayarları

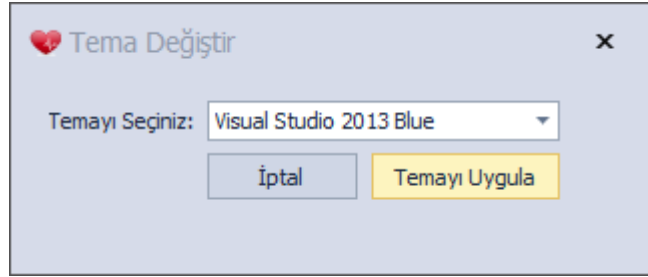
4.9.2.1 BAĞLANTI AYARLARI

The screenshot shows a dialog box titled "Servis İçin Root Adres Ayarlama" with a red heart icon and a close button (X). It contains a "Root Adres" field with the value "127.0.0.1" and a port field with the value "81". Below the fields is a note: "Başına http:// yazmadan giriniz." (Enter without http:// at the beginning). There are two buttons: "İptal" (Cancel) and "Tamam" (OK).

Şekil 4.9.3 Servis Ayarları

BLL katmanının yayınlandığı WCF web servise ait yolun ayarlandığı ekrandır. Sisteme doğru yol girilmediği sürece sistem çalışmayacaktır.

4.9.2.2 TEMA DEĞİŞTİR



Şekil 4.9.4 Tema Ayarı

Kullanıcın dilerse programın temasını değiştirebileceği ekrandır.

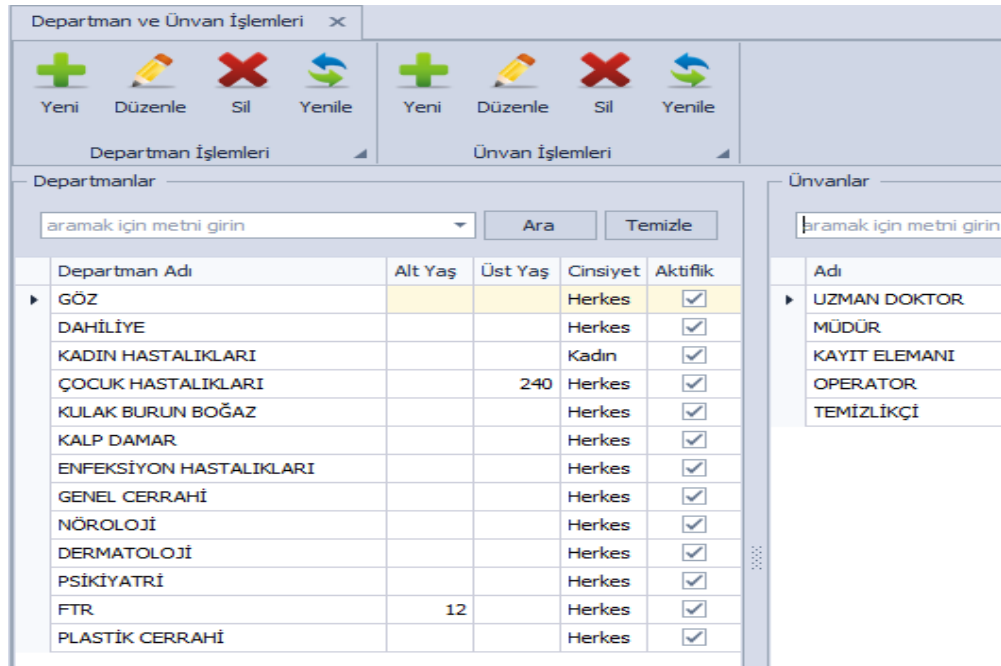
4.9.2.3 TÜRKÇE DİL DOSYALARINI İNDİR

Programda kullanılan DevExpress bileşenlerine ait ileri özelliklerin Türkçe dil dosyalarının ana sistemden çekilmesini sağlayan komuttur.

4.9.3 YÖNETİCİ EKRANI KOMUTLARI

Sadece sistem yönetici tarafından denetlenen özelliklerdir.

4.9.3.1 DEPARTMAN VE UNVAN İŞLEMLERİ



Şekil 4.9.5 Departman ve Ünvan İşlemleri

Sisteme kayıtlı departman ve unvanların görüntülendiği ve düzenlendiği ekrandır.

4.9.3.1.1 DEPARTMAN EKRANI

Şekil 4.9.6 Departman Ekranı

Sisteme yeni bir departman eklemek veya var olan bir departmanı düzenlemek için kullanılan ekrandır. Bu ekranda yer alan en küçük ve en yüksek yaşlar, ilgili departmana gidebilecek hastaların yaşlarının ay cinsinden sınırlarıdır. Eğer böyle bir şart yoksa ilgili alan boş bırakılır. Departman düzenleneceği zaman da ana ekranda ilgili departman seçildikten sonra “Düzenle” butonuna tıklanır ve seçilen departman bu ekrana parametre olarak gönderilir böylece yeni departman ekleme ve var olan departmanı güncelleme için aynı ekran kullanılmış olur.

4.9.3.1.2 UNVAN EKRANI

Şekil 4.9.7 Ünvan Ekranı

Sisteme yeni bir unvan girilmesi veya var olan bir unvanı düzenlemek için kullanılan ekrandır. Bu ekranda yer alan “Yönetici” seçeneği eklenecek unvanın sistem üzerinde yetkili olup olmadığını belirten seçenektir. Unvan düzenleme işlemi de daha önce anlatılan departman düzenleme işlemi ile aynı mantıkta çalışmaktadır.

4.9.3.2 ŞEHİR VE İLÇE İŞLEMLERİ

Şehir ve İlçe İşlemleri

Şehirler

aramak için metni girin

Şehir Adı	Aktiflik
BİNGÖL	<input checked="" type="checkbox"/>
BİTLİS	<input checked="" type="checkbox"/>
BOLU	<input checked="" type="checkbox"/>
BURDUR	<input checked="" type="checkbox"/>
BURSA	<input checked="" type="checkbox"/>
ÇANAKKALE	<input checked="" type="checkbox"/>
ÇANKIRI	<input checked="" type="checkbox"/>
ÇORUM	<input checked="" type="checkbox"/>
DENİZLİ	<input checked="" type="checkbox"/>
DİYARBAKIR	<input checked="" type="checkbox"/>
EDİRNE	<input checked="" type="checkbox"/>
ELAZIĞ	<input checked="" type="checkbox"/>
ERZİNCAN	<input checked="" type="checkbox"/>
ERZURUM	<input checked="" type="checkbox"/>
ESKİŞEHİR	<input checked="" type="checkbox"/>
GAZİANTEP	<input checked="" type="checkbox"/>
GİRESUN	<input checked="" type="checkbox"/>
GÜMÜŞHANE	<input checked="" type="checkbox"/>
HAKKARİ	<input checked="" type="checkbox"/>
HATAY	<input checked="" type="checkbox"/>

İlçeler

aramak için metni girin

İlçe Adı	Şehir
SEYHAN	ADANA
YÜREĞİR	ADANA
SARIÇAM	ADANA
ÇUKUROVA	ADANA
ALADAĞ(KARSANTI)	ADANA
CEYHAN	ADANA
FEKE	ADANA
İMAMOĞLU	ADANA
KARAIŞALI	ADANA
KARATAŞ	ADANA
KOZAN	ADANA
POZANTI	ADANA
SAİMBEYLİ	ADANA
TUFANBEYLİ	ADANA
YUMURTALIK	ADANA

Şekil 4.9.8 Şehir ve İlçe Ekranı

Sisteme kayıtlı şehir ve ilçelerin listesinin gösterildiği ekrandır. Şehir listesinde herhangi bir şehir tıklandığı zaman ilçe listesinde tıklanan şehrin ilçe listesi gösterilecektir.

4.9.3.2.1 ŞEHİR EKRANI

Şekil 4.9.9 Şehir Ekranı

Sisteme yeni şehir ekleneceği veya var olan bir şehrin güncelleneceği zaman kullanılan ekrandır.

4.9.3.2.2 İLÇE EKRANI

Şekil 4.9.10 İlçe Ekranı

Sisteme yeni ilçe eklenmesinde veya var olan bir ilçenin düzenlenmesinde kullanılan ekrandır. Bu ekranda yer alan il seçeneği, eklenecek olan ilçenin bağlı olduğu ildir.

4.9.3.3 HASTALIKLAR VE İLAÇLAR EKRANI

Teşhisler ve İlaçlar x

Yeni Düzenle Sil Yenile Yeni Düzenle Sil Yenile

Tanı İşlemleri İlaç İşlemleri

Teşhisler

aramak için metni girin Ara Temizle

Ad	Bölüm	Aktiflik
► MİYOP	GÖZ	<input checked="" type="checkbox"/>
AĞRI	DAHİLİYE	<input checked="" type="checkbox"/>
AKDENİZ ANEMİSİ	DAHİLİYE	<input checked="" type="checkbox"/>
ALERJİ	DAHİLİYE	<input checked="" type="checkbox"/>
ANEMİ	DAHİLİYE	<input checked="" type="checkbox"/>
BAŞ DÖNMESİ	DAHİLİYE	<input checked="" type="checkbox"/>
DİABET	DAHİLİYE	<input checked="" type="checkbox"/>
GASTRİT	DAHİLİYE	<input checked="" type="checkbox"/>
GUATR	DAHİLİYE	<input checked="" type="checkbox"/>
HEMOFİLİ	DAHİLİYE	<input checked="" type="checkbox"/>
HEPATİT	DAHİLİYE	<input checked="" type="checkbox"/>
İSHAL	DAHİLİYE	<input checked="" type="checkbox"/>
KABIZLIK	DAHİLİYE	<input checked="" type="checkbox"/>
KALINBAĞIRSAK	DAHİLİYE	<input checked="" type="checkbox"/>
KANSER	DAHİLİYE	<input checked="" type="checkbox"/>
KARACİĞER KANSERİ	DAHİLİYE	<input checked="" type="checkbox"/>
KİST	DAHİLİYE	<input checked="" type="checkbox"/>
LÖSEMİ	DAHİLİYE	<input checked="" type="checkbox"/>

İlaçlar

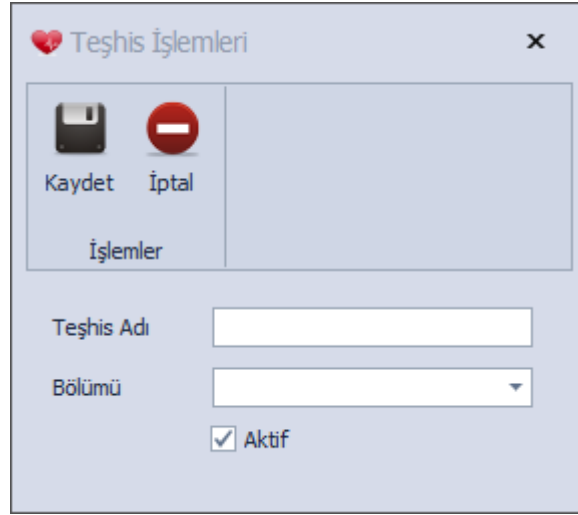
aramak için metni girin

İlaç Adı	Kullanım(Gün.Adet)	Not
► HUMAN ALBUMI...	1,5	
HUMAN ALBUMI...	1,5	
OCTAGAM 10 G...	1,3	
OCTAGAM 5 G ...	3,1	
OCTAGAM IMM...	3,3	
PLASBUMIN %2...	3,2	
PLASBUMIN %2...	2,1	
PLASBUMIN %2...	2,5	
PLASBUMIN %2...	1,4	
GAMUNEX %10...	1,3	
GAMUNEX %10...	3,1	
GAMUNEX %10...	2,2	
CEALB ALBUMI...	3,2	
CEALB ALBUMI...	2,4	
TETAQUINTETA...	3,3	
FLEBOGAMMA ...	3,3	
FLEBOGAMMA ...	2,4	
FLEBOGAMMA ...	1,5	

Şekil 4.9.11 Hastalık ve İlaç Ekranı

Sisteme kayıtlı hastalık ve ilaçların listelendiği ekrandır.

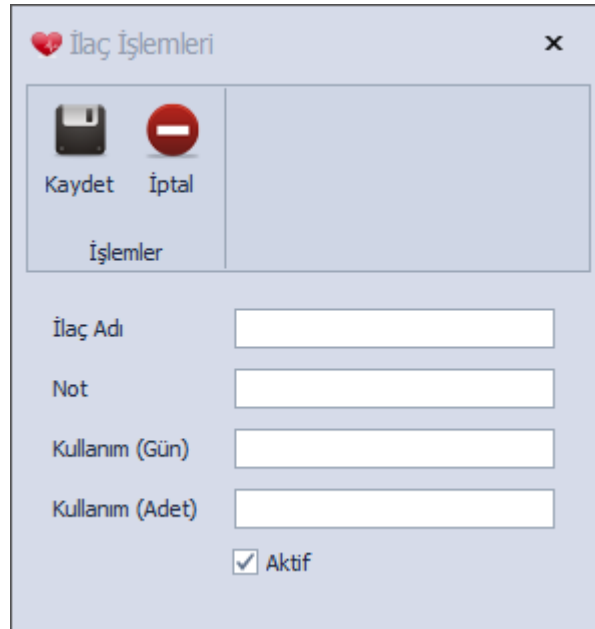
4.9.3.3.1 HASTALIK EKRANI



Şekil 4.9.12 Hastalık Ekranı

Sisteme yeni hastalık eklenmesini veya var olan bir hastalığın düzenlenmesi amacıyla kullanılan ekrandır. Ekranda yer alan “Bölümü” seçeneği hastalığın hangi departmanda yer aldığını belirtir.

4.9.3.3.2 İLAÇ EKRANI









Şekil 4.9.13 İlaç Ekranı

Sisteme yeni ilaç ekleme veya var olan bir ilacı düzenlemek amacıyla kullanılan ekrandır. Ekranda yer alan “Not” seçeneği ilaç hakkında varsa açıklama, “Kullanım (gün)” ve “Kullanım(Adet)” seçenekleri ise ilacın olağan şartlarda tavsiye edilen kullanım sıklığını ifade eder.

4.9.3.4 DOKTOR İŞLEMLERİ EKRANI

Doktor İşlemleri x



Doktor İşlemleri

Genel İş...

Ara Temizle

Gruplamak için bir sütun başlığını buraya sürükleyin

	Ünvan	Departman	Ad	Soyad	Baba Adı	Anne Adı
►	UZM. DR.	GÖZ	EMRULLAH	YURT	EMİN	SALİHA
	UZM. DR.	PLASTİK CERRAHİ	MEHMET	SÖĞÜT	YUNUS	HAFİZE
	UZM. DR.	PLASTİK CERRAHİ	AMİNE	ORHAN	BEŞİR	MÜMİNE
	UZM. DR.	GÖZ	MUSTAFA	SARUHAN	ZEKİ	EMİNE
	UZM. DR.	DAHİLİYE	HUSSEİN	BAHAVE	ADAM	ZAİNAB
	UZM. DR.	FTR	AYFER	AYDIN	NURİ	HALİME
	UZM. DR.	FTR	MUHAMMED	YILDIZ	ABDULHAKİM	AMİNE
	UZM. DR.	DAHİLİYE	TUBA	VATAN	SALİH	EMİNE
	UZM. DR.	PSİKİYATRİ	SAİD NUR	YAĞMAHAN	ZEKİ	NEZAHAT
	UZM. DR.	KADIN HASTALIKLARI	VERA	YURT	MEHMET EMİN	GÜLİSTAN
	UZM. DR.	PSİKİYATRİ	KEMAL	SAYAR	CEMİL	AYŞE
	UZM. DR.	DERMATOLOJİ	SELMAN	ÖZCAN	HAMZA	GAMZE
	UZM. DR.	KADIN HASTALIKLARI	ANNA MARIA	WLAT	HUSSEİN	VERA
	UZM. DR.	DERMATOLOJİ	RİDVAN	EKİCİ	UFUK	KADRİYE
	UZM. DR.	NÖROLOJİ	HADİ	KIZMAZ	DİLAVER	BÜŞRA
	UZM. DR.	ÇOCUK HASTALIKLARI	ANIL	AKBIYIK	ALİ	ZEHRA
	UZM. DR.	ÇOCUK HASTALIKLARI	ZEHRA	YAĞMAHAN	MEHMET	TUBA
	UZM. DR.	NÖROLOJİ	SELAMİ	BAĞLAMIS	ŞEHMUS	LALE

Şekil 4.9.14 Doktor İşlemleri Ekranı

Sisteme kayıtlı doktorların listelendiği ekrandır. “Doktor Listesini Yazdır” butonu ile doktorların listesi yazdırılabilir.

4.9.3.4.1 DOKTOR EKRANI

Doktor Ekleme

Kaydet İptal

İşlemler

Tc Numarası

Adı

Soyadı

Anne Adı

Baba Adı

Doğum Tarihi

Departman Adı Departman Seçiniz

Ünvanı Ünvan Seçiniz

Aktiflik ☒ Aktif

Telefon Numaraları

Telefon

Mailler

Mail

Telefon Numarası Ekle E-mail Adresi Ekle

Şekil 4.9.15 Doktor Ekranı

Sisteme yeni doktor ekleme, var olan bir doktoru düzenleme veya ana ekranda seçilen doktorun detaylarının görüntülendiği ekrandır.

4.9.3.5 TATİL İŞLEMLERİ EKRANI

Tatil İşlemleri x

Yeni Düzenle Sil Yenile

Tatil İşlemleri

aramak için metni girin Ara Temizle

Tatil Adı	Gün	Ay	Yıl
► YILBAŞI		1 Ocak	2015
ULUSAL EGEMENLİK VE ÇOCUK BAYR...		23 Nisan	Her Yıl
EMEK VE DAYANIŞMA GÜNÜ		1 Mayıs	Her Yıl
ATATÜRK'Ü ANMA GENÇLİK VE SPOR...		19 Mayıs	Her Yıl
ZAFER BAYRAMI		30 Ağustos	Her Yıl
CUMHURİYET BAYRAMI		29 Ekim	Her Yıl
RAMAZAN BAYRAMI 1. Gün		17 Temmuz	2015
RAMAZAN BAYRAMI 2. Gün		18 Temmuz	2015
RAMAZAN BAYRAMI 3. Gün		19 Temmuz	2015
KURBAN BAYRAMI 1. Gün		24 Eylül	2015
KURBAN BAYRAMI 2. Gün		25 Eylül	2015
KURBAN BAYRAMI 3. Gün		26 Eylül	2015
KURBAN BAYRAMI 4. Gün		27 Eylül	2015

Şekil 4.9.16 Tatil İşlemleri Ekranı

Sisteme kayıtlı tatillerin listelendiği ekrandır.

4.9.3.5.1 TATİL EKRANI

♥ Tatil İşlemleri x

Kaydet İptal

İşlemler

Tatil Adı

Tatilin İlk Günü 22 Ocak 2015

☒ Her Yıl Aynı Günde Oluyor

Tatil Uzunluğu

Şekil 4.9.17 Tatil Ekranı






Sisteme yeni tatil ekleme veya var olan bir tatili düzenlemek için kullanılan ekrandır. Tatil ekleme işlemi şu şekilde olmaktadır:

Tatilin ilk günü seçilip tatil uzunluğu belirtildikten sonra sistem tatil uzunluğu kadar günü başlangıç gününden itibaren tatil olarak sisteme kaydeder. “Her Yıl Aynı Günde Oluyor” seçeneği de girilen tatilin her yıl periyodik olarak aynı günde olan tatildir. Örneğin resmi bayram günleri buna örnektir.

4.9.4 PERSONEL EKRANI KOMUTLARI

Sadece normal personel tarafından görülebilen ekrandır.

4.9.4.1 HASTA İŞLEMLERİ EKRANI

Kayıtlı Hastalar x							
<div>      </div>							
Hasta İşlemleri							
<input type="text" value="İbrahim için metni girin"/> <input type="button" value="Ara"/> <input type="button" value="Temizle"/>							
Tc No	Adı Soyadı	Doğum Günü	Kan Grubu	Şehir	İlçe	Cinsiyet	
38623173763	SERKAN ÜNLÜ	20 Ekim 1974	A RH +	ADIYAMAN	ÇELİKHAN	Erkek	
86480656234	MELİSA ALP	02 Eylül 1961	B RH +	AMASYA	MERZİFON	Kadın	
84271116602	SERLİ ABRAHAMO...	17 Nisan 1988	A RH -	EDİRNE	KEŞAN	Erkek	
52795883947	MERYEM ARSLAN	15 Ekim 1982	AB RH -	BİNGÖL	KARLIOVA	Kadın	
61396189701	GAMZE YILMAZ	06 Eylül 2008	AB RH -	DENİZLİ	GÜNEY	Kadın	
12345678904	ÖMER FARUK YIL...	10 Eylül 1974	AB RH +	AĞRI	DOĞUBEYAZIT	Erkek	
48611594121	DİLAN ÇELİK	18 Ağustos 2011	AB RH +	DİYARBAKIR	DİCLE	Kadın	
21684553654	DİCLE YILMAZ	25 Haziran 1978	O RH +	ELAZIĞ	BASKİL	Kadın	
54388594822	HATİCE ÖZGÜVEN	02 Nisan 1968	B RH -	AYDIN	AYDIN	Kadın	
85584851507	ELİF ALMAS	08 Temmuz 1963	O RH +	ŞANLIURFA	BOZOVA	Kadın	
16346145208	HEKİM HEKİMOĞLU	24 Nisan 2011	AB RH -	BİNGÖL	KARLIOVA	Erkek	
85192892867	FURKAN BİRİCİK	26 Nisan 1962	O RH -	ARTVİN	ARHAVİ	Erkek	
57737061394	MEHMET AKINCILAR	23 Mayıs 2009	AB RH +	ÇORUM	ORTAKÖY	Erkek	
82528301274	FEHMİ KUTLAR	07 Haziran 2000	O RH +	BURDUR	GÖLHİSAR	Erkek	
92176692075	JALE HÜR	17 Ağustos 1972	A RH -	BİTLİS	GÜROYMAK	Kadın	
86917671884	DERYA DOST	26 Haziran 1987	AB RH +	OSMANİYE	HASANBEYLİ	Kadın	
84369363479	MERCAN GÜROYMAZ	03 Ağustos 1993	AB RH +	AYDIN	İNCİRLİOVA	Kadın	
87729643232	HAKAN HOROZ	15 Ağustos 1961	B RH +	HATAY	HASSA	Erkek	
43244589614	VEYSEL KARAKAŞ	24 Mayıs 1992	B RH -	BİNGÖL	KARLIOVA	Erkek	

Şekil 4.9.18 Hasta İşlemleri Ekranı

Sisteme kayıtlı hastaların listesini gösteren ekrandır.

4.9.4.1.1 HASTA EKRANI

Hasta Kayıt

Kaydet İptal

İşlemler

Kişisel Bilgiler

TC Kimlik No

Adı

Soyadı

Baba Adı

Anne Adı

Doğum Tarihi

Cinsiyeti

Kan Grubu

☒ Aktif

İletişim Bilgileri

Adresi

Şehri

İlçesi

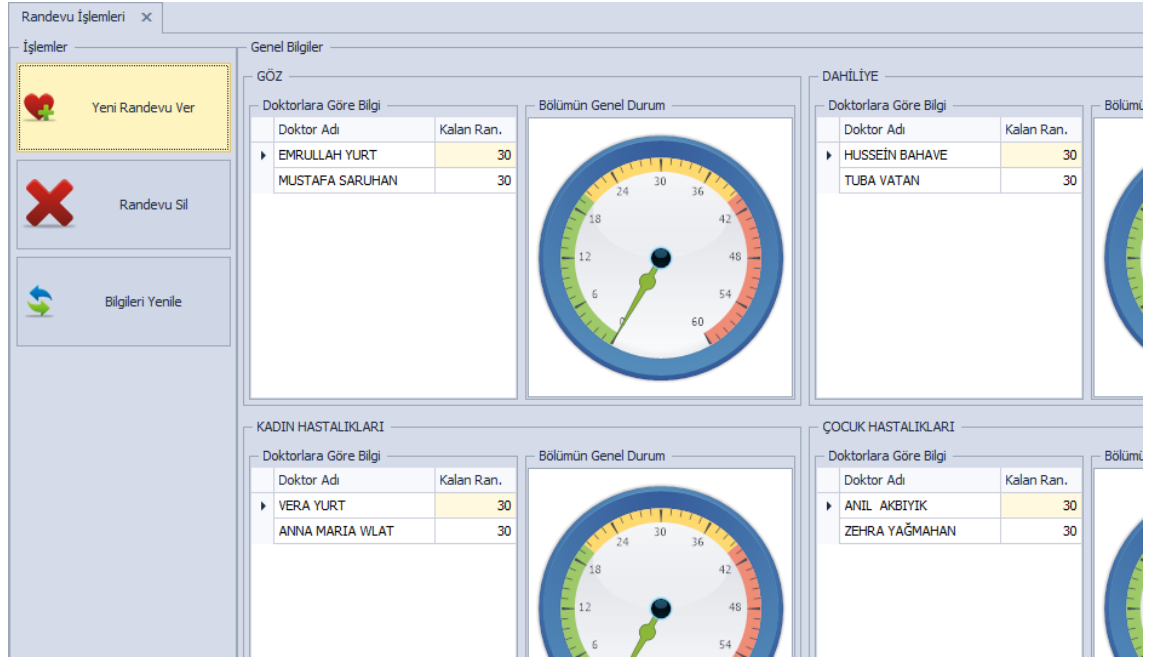
Telefon Numarası

Mail Adresi

Şekil 4.9.19 Hasta Ekranı

Sisteme yeni hasta kaydı almak veya var olan bir hastanın bilgilerini düzenlemek için kullanılan ekrandır.

4.9.4.2 RANDEVU İŞLEMLERİ EKRANI



Şekil 4.9.20 Randevu İşlemleri Ekranı

Sisteme randevu girilmesini veya var olan bir randevuyu silmek için kullanılan ekrandır. Ekranda yer alan “Genel Bilgiler” panelinde hangi doktora kaç hasta kaydının daha yapılabileceği gösterilmektedir ve ekrandaki sayaç da hangi bölümde bölümün doluluk oranı gösterilmektedir.

4.9.4.2.1 YENİ RANDEVU EKRANI

Yeni Randevu

Kaydet İptal Bölüm ve Doktor Listesi Yenile

İşlemler

Hasta TC No 12345678904

Yeni Hasta Kaydı Hasta Bilgilerini Getir

Hasta Adı ÖMER FARUK YILDIZ

Bölüm Seçiniz DAHİLİYE

Doktor Seçiniz TUBA VATAN

Uygun Saatler

Başlama Saati
09:00:00
09:30:00
09:45:00
10:00:00
10:15:00
10:30:00
10:45:00

Şekil 4.9.21 Yeni Randevu Ekranı

Sisteme yeni randevu eklemek için kullanılan ekrandır. Randevu verme işlemi şöyle yapılmaktadır, öncelikle randevu verilecek hastanın kimlik numarası girilerek “Hasta Bilgilerini Getir” butonu ile ekrana hasta bilgileri getirilir. Daha sonra bölüm ve doktor bilgileri girilir. Hasta ilk defa geliyorsa “Yeni Hasta Kaydı” butonu ile yeni hasta kaydı ekranı açılarak yeni hasta kaydı yapılır.

4.9.4.2.2 RANDEVU SİLME EKRANI

Şekil 4.9.22 Randevu Silme Ekranı

Sisteme girilen bir randevuyu silmek için kullanılan ekrandır. Öncelikle silinecek randevuya ait numara girilip “Randevu Bul” ekranı tıklanarak silinecek randevu bilgileri kontrol edilerek silinir. Randevuyu silmek için randevunun muayeneye girmemiş olması gerek.

4.9.5 DOKTOR EKRANI KOMUTLARI

Sadece doktor ekranında görülebilen komutlardır.

4.9.5.1 MUAYENE İŞLEMLERİ

The screenshot displays a software interface for patient examination processes. The interface is divided into several sections:

- Sıradaki Hastalar (Next Patients):** A list of patients, with "ÖMER FARUK YILDIZ" selected.
- Şuanki Muayene Bilgisi (Açık Muayene) (Current Examination Information (Open Examination)):**
 - Hastanın Adı Soyadı: ÖMER FARUK YILDIZ
 - Teghis: A dropdown menu.
 - Not: A text area for notes.
 - Sonraki Kontrol Zamanı: A date and time selector, currently set to "gün sonra" (tomorrow).
 - ☐ Mesaj Gönderemez
 - Buttons: "Laboratuvar İsteği Oluştur" (Create Lab Request), "Muayene Bitirme Seçenekleri" (Examination Completion Options), "Geçmiş Muayene ve Mesajlaşma Bilgileri" (Past Examination and Messaging Information), and "Laboratuvar Sonuçlarını Yazdır" (Print Lab Results).
- Laboratuvar Sonuçları (Laboratory Results):** A table with columns: Teşik, Min Değer, Max Değer, and Sonuç.
- Mesajlar (Messages):** A table with columns: Tarih, Gönderen, and Mesaj.
- Geçmişteki Muayeneler (Past Examinations):** A table with columns: Tarih, Kontrol Tarihi, and Teghis.
- Laboratuvar İstekleri (Laboratory Requests):** A section for managing lab requests.

Şekil 4.9.23 Muayene İşlemleri Ekranı

Doktorun hastalarının listesini gördüğü ekrandır. Hasta listesini iki farklı şekilde görebilir. Varsayılan değer olarak muayene olmuş hastaların isimleri listede gösterilmektedir. İsterse hasta listesini sağ tıklayıp “Muayene Olmuşları Göster” seçeneğini seçerek o günkü bütün hastaları görebilir.

4.9.5.1.1 LABORATUVAR İSTEĞİ OLUŞTURMA EKRANI

<input type="checkbox"/>	İstek Adı
<input type="checkbox"/>	GLUKOZ
<input type="checkbox"/>	TRİGLİSERİT
<input type="checkbox"/>	KOLESTEROL
<input type="checkbox"/>	HDL-KOLESTEROL
<input type="checkbox"/>	LDL-KOLESTEROL
<input type="checkbox"/>	BUN
<input type="checkbox"/>	KREATİNİN
<input type="checkbox"/>	ÜRİK ASİT
<input type="checkbox"/>	KALSİYUM
<input type="checkbox"/>	FOSFOR
<input type="checkbox"/>	TOTAL BİLİRUBİN
<input type="checkbox"/>	DİREKT BİLİRUBİN
<input type="checkbox"/>	AMİLAZ

Şekil 4.9.24 Laboratuvar İsteği Ekranı

Muayene edilen hasta için gerek görülürse laboratuvar isteği oluşturmak için kullanılacak ekrandır.

4.9.5.1.2 GEÇMİŞ MUAYENE VE MESAJLAŞMA BİLGİLERİ

Geçmişteki Muayeneler		
Tarih	Kontrol Tarihi	Teşhis
▶ 13 Kasım 2014	03 Aralık 2014	ALERJİ
09 Aralık 2014	24 Aralık 2014	ÜLSER
15 Ocak 2015		ALERJİ

Şekil 4.9.25 Geçmiş Muayene ve Mesajlaşma Bilgileri

Muayene edilen hastanın istenirse geçmişte olduğu muayeneler ve doktorla yaptığı mesajların listesini görebilir. Bu panelde listelenen muayeneye çift tıklanarak ayrıntılarına ulaşılabilir.

Geçmiş Muayene

X

Muayene Bilgileri

Tarih9 Aralık 2014 Salı

Kontrol Tarihi24 Aralık 2014 Çarşamba

TeşhisÜLSER

Doktor NotoAsitli yiyecekler yemeyecsin

Laboratuvar İstekleri

9.12.2014 08:19

Reçete Bilgileri

İlaç Adı	Kullanma Sıklığı
OCTAGAM 5 G 100 ML INFUZYO...	Günde 2 defa
CEALB ALBUMIN %20 50 ML 10 GR	Günde 3 defa
HUMAN ALBUMIN %20 50 ML IN...	Günde 1 defa
CALCIDINE 75 GR GRANUL	Günde 5 defa
AYRA 8 MG 28 TABLET	Günde 2 defa

Tetkik	Min	Max	Sonuç
KOLESTEROL	70	200	
FOSFOR	2,5	4,7	
LİPAZ	13	60	
AST	0	32	
ALT	0	33	
ALKALEN FOSFATAZ	35	105	
DEMİR BAĞLAMA	225	480	

Şekil 4.9.26 Geçmiş Muayene Ayrıntısı

Çift tıklanan geçmiş muayenede yazılan reçeteye ve istenen laboratuvar isteklerine ulaşılabilir.

4.9.5.1.3 REÇETELİ MUAYENE EKRANI

Reçete Yaz: ÖMER FARUK YILDIZ

Kaydet Reçeteye Not Ekle İptal İşlemler

Tüm İlaçlar

aramak için metni girin Ara Temizle

	Adı	Kullanım D...
<input checked="" type="checkbox"/>	HUMAN ALBUMIN %20 BERK 50 ML	1,5
<input type="checkbox"/>	OCTAGAM 10 G 200 ML INFUZYON...	1,3
<input type="checkbox"/>	OCTAGAM 5 G 100 ML INFUZYONL...	3,1
<input checked="" type="checkbox"/>	OCTAGAM IMMUNGLOBULI 2.5 GR ...	3,3
<input type="checkbox"/>	PLASBUMIN %20 50 ML 1 FLAKON	3,2
<input checked="" type="checkbox"/>	PLASBUMIN %20 100 ML 1 FLAKON	2,1
<input type="checkbox"/>	PLASBUMIN %25 50 ML 1 FLAKON	2,5
<input checked="" type="checkbox"/>	PLASBUMIN %25 100 ML 1 FLAKON	1,4
<input type="checkbox"/>	GAMUNEX %10 100 ML IV ENJEKSI...	1,3
<input checked="" type="checkbox"/>	GAMUNEX %10 200 ML IV ENJEKSI...	3,1
<input type="checkbox"/>	GAMUNEX %10 50 ML IV ENJEKSIY...	2,2
<input type="checkbox"/>	CEALB ALBUMIN %20 100 ML 1 AD...	3,2
<input type="checkbox"/>	CEALB ALBUMIN %20 50 ML 10 GR	2,4

5868

Reçeteye Eklenmiş İlaçlar

İlaç Adı	Kullanım Sıklığı (gün,adet)
HUMAN ALBUMIN %20 BERK 50 ML	1,2
OCTAGAM IMMUNGLOBULI 2.5 GR 1 FLA...	1,3
PLASBUMIN %20 100 ML 1 FLAKON	2,1
PLASBUMIN %25 100 ML 1 FLAKON	1,4
GAMUNEX %10 200 ML IV ENJEKSIYONL...	1,2

Şekil 4.9.27 Reçete Yazma Ekranı

Hasta muayene olduktan sonra doktor iki şekilde muayeneyi kapatabilir. Bu seçeneklere “Muayene Kapatma Seçenekleri” butonu ile ulaşabilir. Muayeneyi reçeteli olarak kapatmayı seçerse reçete yazma ekranı açılır. Bu ekranın sol tarafından reçeteye seçeceği ilaçları seçtikten sonra sağ tarafta ilaçların kullanma sıklığını da belirtip kaydeder. Burada dilerse reçeteye not da ekleyebilir.

4.10 UYGULAMA DAĞITMADA ONCECLICK AYARLAMA

Programın son kullanıcıya ulaştırılması için kurulum paketi oluşturma yöntemi kullanılabileceği gibi Microsoft firması tarafından geliştirilen ve tek tıkla programının kurulmasını sağlayan onceclick yöntemi de kullanılabilir. Onceclick ayarlandıktan sonra programın kurulumu şu şekilde yapılmaktadır. Öncelikle ilgili linkten kurulum dosyası indirilir. Daha sonra dosya çalıştırıldığında önce programın sistem gereksinimleri test edilir. Gereksinimler sağlanıyorsa program dosyaları indirilip gerekli ayarlamalar otomatik olarak yapılır.

Onceclick ayarlamak için de öncelikle Visual Studio’yu yönetici olarak çalıştırmak gerekmektedir.

Bu işlem için öncelikle IIS üzerinde yayın yapılacağı bir web sayfanın kurulması gerekmektedir. Web sayfasının kurulumu da şu şekilde yapılmaktadır:

Dosyaların kaydedileceği bir klasör oluşturulur.

Web sayfası IIS üzerinde kurulur. Bunun için IIS yönetici ekranında server siteler başlığı üzerinde sağ tıklanıp “Yeni Site” komutu tıklanır. Aşağıdaki pencere açılacaktır.

Şekil 4.10.1 Sitenin Açılması

Bu ekranda girilecek bilgiler şöyledir:

Site Name: sitenin adı

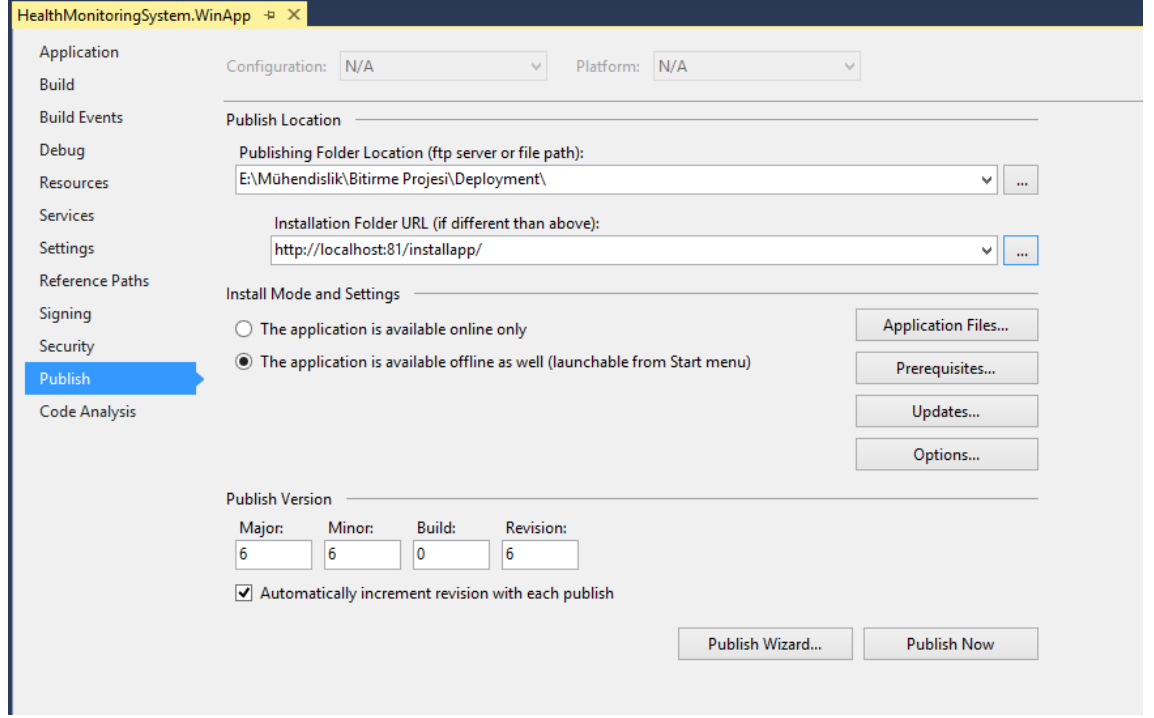
Physical path: kurulum dosyalarının bulunacağı klasörün adresi

Port: sitenin yayınlanacağı port numarası

Bu bilgiler girildikten sonra OK butonuna tıklanır ve site yayına girer.

Bundan sonra yapılacak işlem programın Visual Studio ekranından siteye yüklenmesidir.

Projeye sağ tıklanıp özellikler tıklanır ve projenin özellikleri menüsünden “Publish” menüsü açılır.



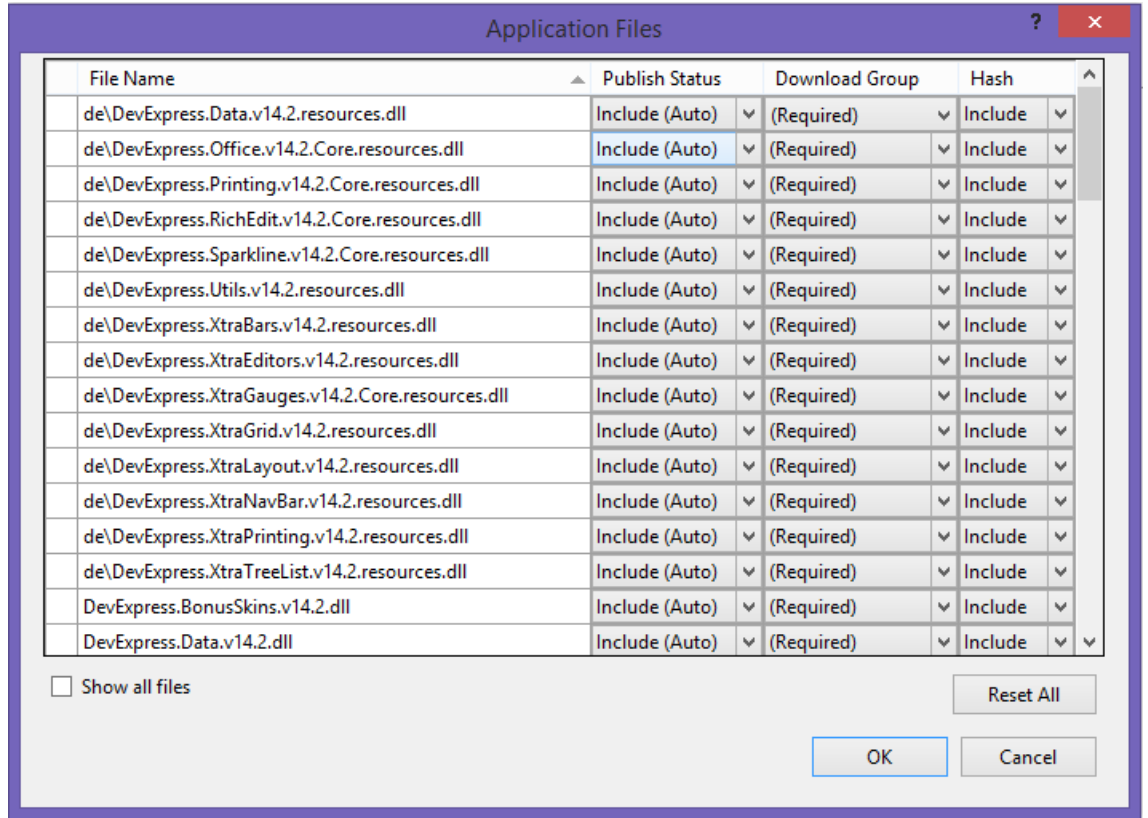
Şekil 4.10.2 Publish Menusu

Bu sayfada yer alan alanların açıklaması şöyledir:

Publishing Folder Location: Kurulum dosyalarının kaydedileceği klasör yolu girilir.

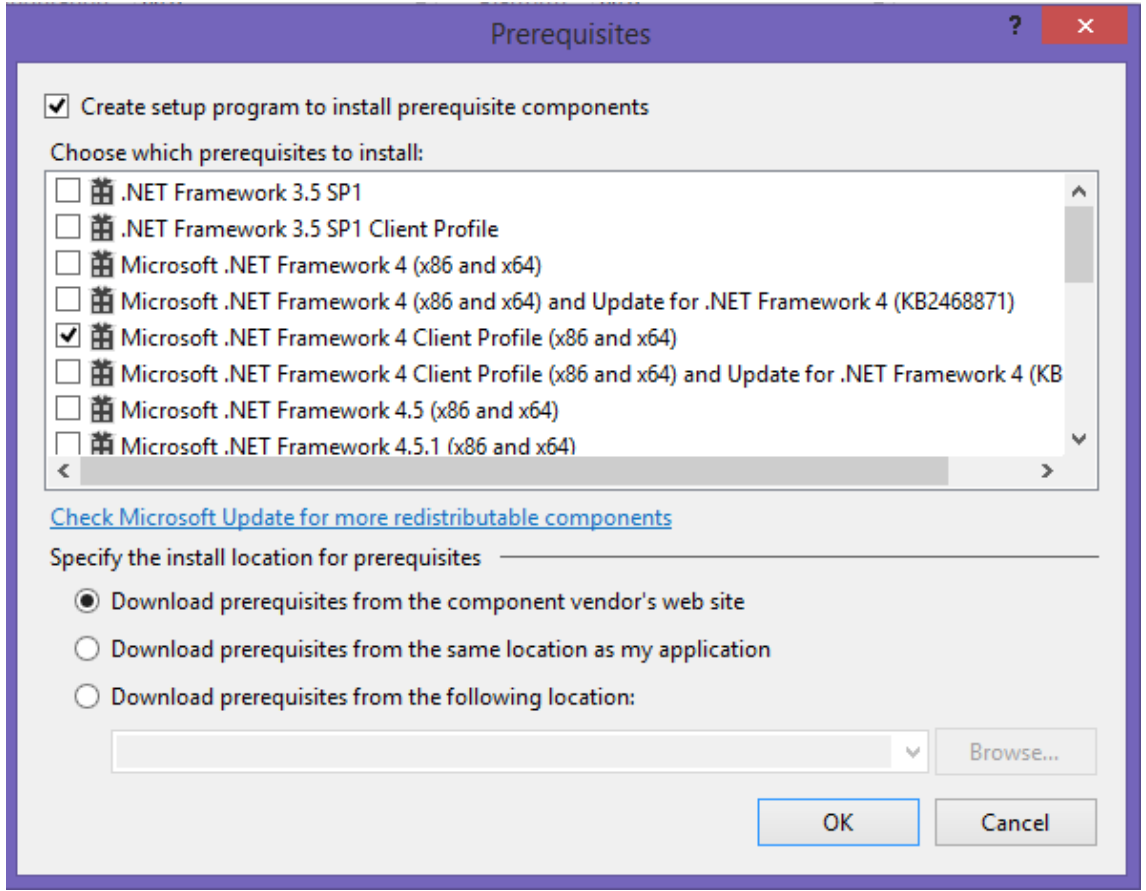
Installation Folder URL: Kurulum dosyaları için oluşturulan web sayfasının adresi buraya girilir.

Publish Version: Programın o an yayındaki versiyon numarasını belirtir. Altta yer alan tik işareti işaretliyse her yeni publishte revision numarası otomatik bir arttırılır.



Şekil 4.10.3 Application Files

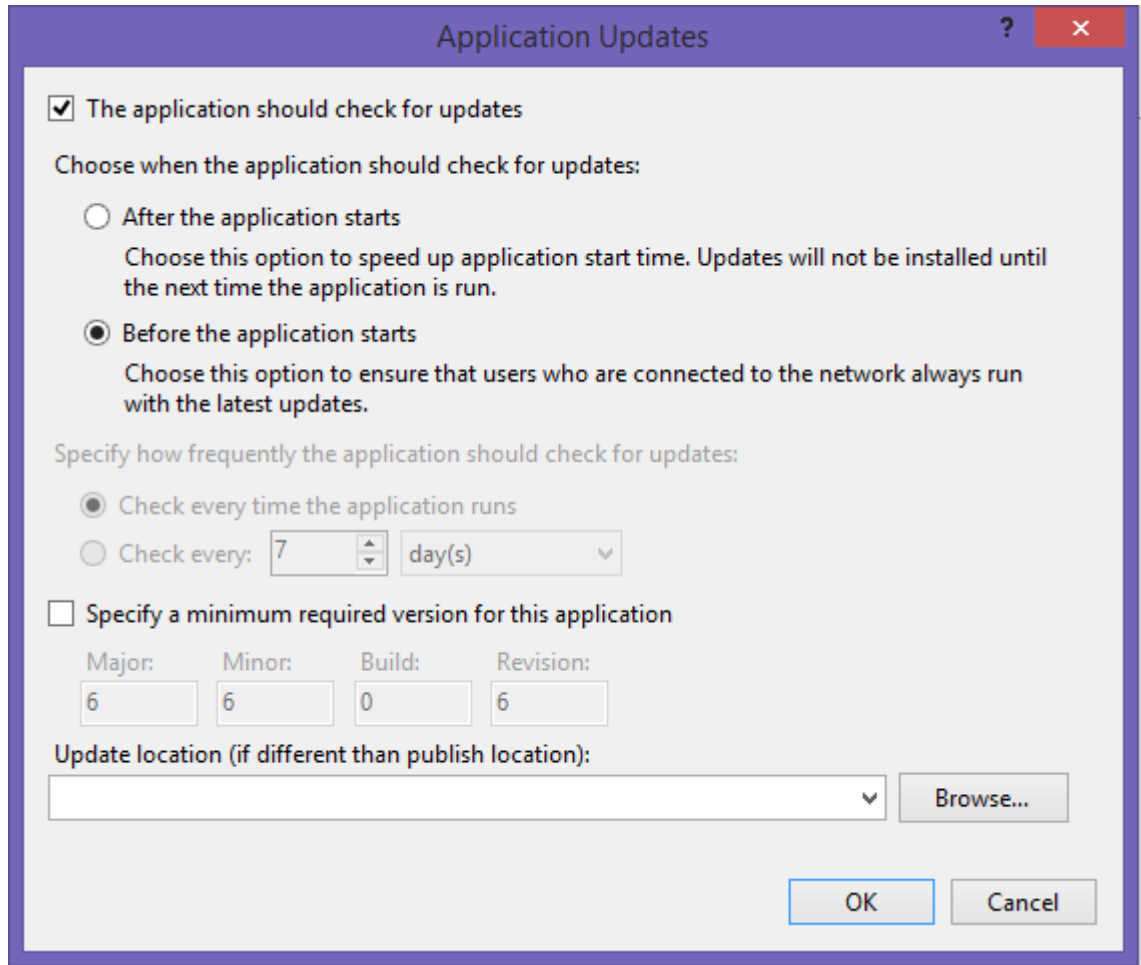
Application Files butonu ile açılan ekranda ise kurulumla dahil edilecek dosyalar seçilir.



Şekil 4.10.4 Prerequisites

Prerequisites butonu ile açılan ekranda da programın çalışması için gereken minimum sistem bileşenleri seçilir. Bu ekranda ayrıca bileşenlerin sistemde bulunamaması durumunda yapılacaklar da seçilebilmektedir. Altta bunun için üç seçenek sunulmaktadır. Sırası ile sunulan seçenekler:

- Bileşeni üreticinin sitesinden indir
- Bileşeni kurulum dosyası ile aynı konumdan indir
- Bileşeni belirtilen yerden indir. Bu seçenek seçilirse aşağıdaki satıra yolu girmek gerekmektedir.

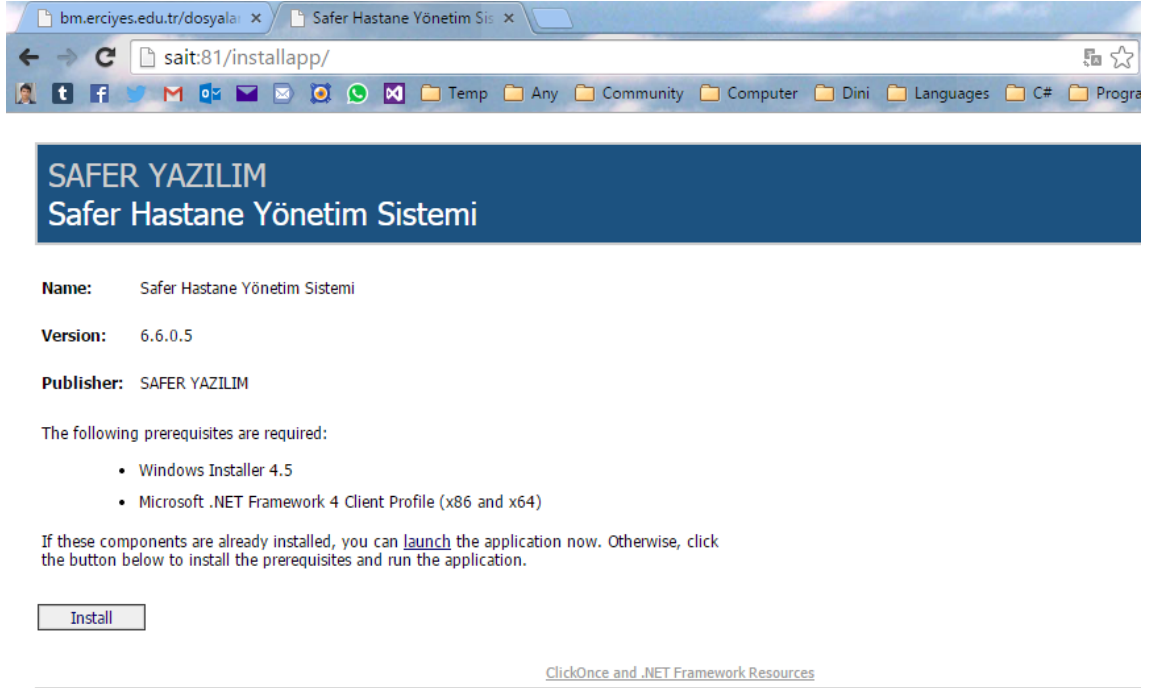


Şekil 4.10.5 Application Updates

Updates butonu ile açılan ekranda da uygulamanın güncelleme kontrolleri ayarlanmaktadır.

Bu işlemlerden sonra “Publish Now” butonu tıklanarak uygulama sunucuya yüklenerek yayınlanır.

4.11 YAYINLANAN UYGULAMANIN BİLGİSAYARA KURULMASI



Şekil 4.11.1 Kurulum Sayfası

Uygulama sunucuda yayınlandıktan sonra uygulamaya ait link üzerinden kurulum sayfası açılır ve “Install” butonuna tıklanarak uygulama yüklenir.