



CELAL BAYAR
ÜNİVERSİTESİ

MANİSA CELAL BAYAR ÜNİVERSİTESİ
HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ
Yazılım Mühendisliği

YZM 2118 – Yazılım Mimarisi ve Tasarımı
Ödev Dokümantasyonu

Diyetisyen Uygulaması

Hazırlayanlar

Hüseyin Yasin DÖNMEZ - 202802003

Melsan Zehra YILMAZ - 202802006

Tuba AKBAŞ - 202803008

HAZİRAN –2021

1. GİRİŞ	
1.1	Projenin Linki..... 3
1.2	Projenin Amacı..... 3
1.3	Projenin Kapsamı..... 3
2. PROJE	
2.1	Projenin UML Class Diyagramları..... 4-5
2.2	Projenin Ekran Görüntüleri..... 5-6
2.3	Proje Kodları..... 7-17

1. GİRİŞ

1.1 Projenin Linki

https://www.youtube.com/watch?v=S8xFF_RwL2I

1.2 Projenin Amacı

Bu proje Diyetisyenlerin hastalarını daha yakından takip etmek amacıyla ve işlerini daha hızlı bir şekilde yürütebilmeleri için geliştirilmiştir. Projede kağıt üzerinde yapılan işlemlerin azaltılması ve buna bağlı olarak sürecin hızlandırılması aynı zamanda kağıt israfını engelleyerek doğaya verilen zararın azaltılması sağlanması amaçlanmıştır. Hasta ve diyetisyen arasındaki iletişimin daha iyi, kayıtların da daha hızlı ve daha kolay bir şekilde tutulabilmesi amaçlayan bir projedir.

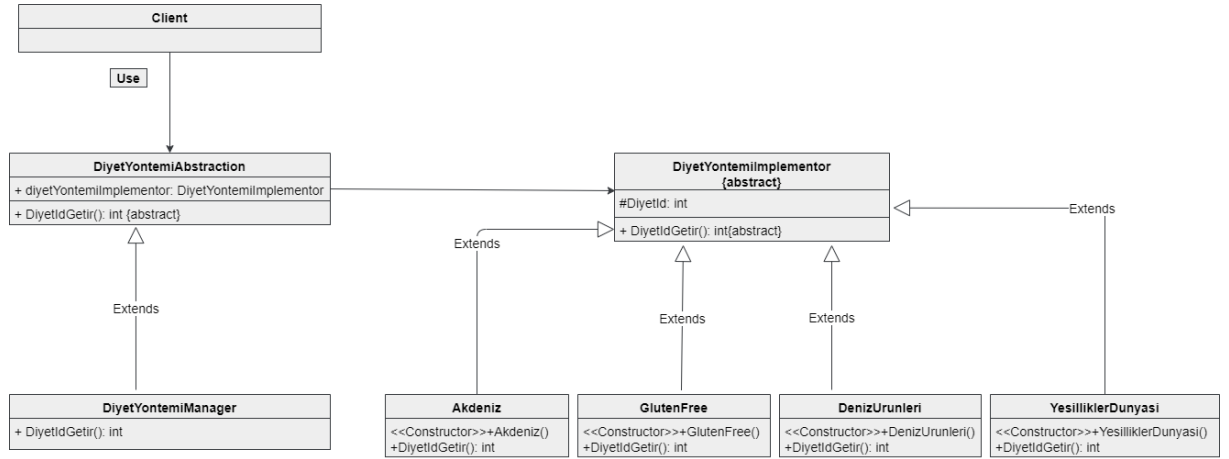
1.3 Kapsam

Bu projede hasta bilgileri ve diyet takvimi ile hastanın diyetisyen bilgileri bir veritabanında tutularak tutarlılık sağlanmaktadır. Projede diyetisyen yeni bir hasta kaydı yapabilmekte aynı zamanda var olan hastaların ise raporlarını istediği format (html-json) ve kombinasyonda (hasta bilgileri-diyet bilgileri veya diyet bilgileri-hasta bilgileri) sistemden oluşturabilmektedir aynı zamanda oluşturulan html raporunu çıktısını alabilir. Sistemde tanımlı adminler tarafından ise sisteme yeni diyetisyenler eklenebilmektedir. Ayrıca raporları bir arada tutarak diyetisyenlerin karşılaşacağı zorlukları minimize etme doğaya ve dünya ekonomisine olan zararı engellemektir. Projemiz sadece Diyetisyenler tarafından kullanılabilir.

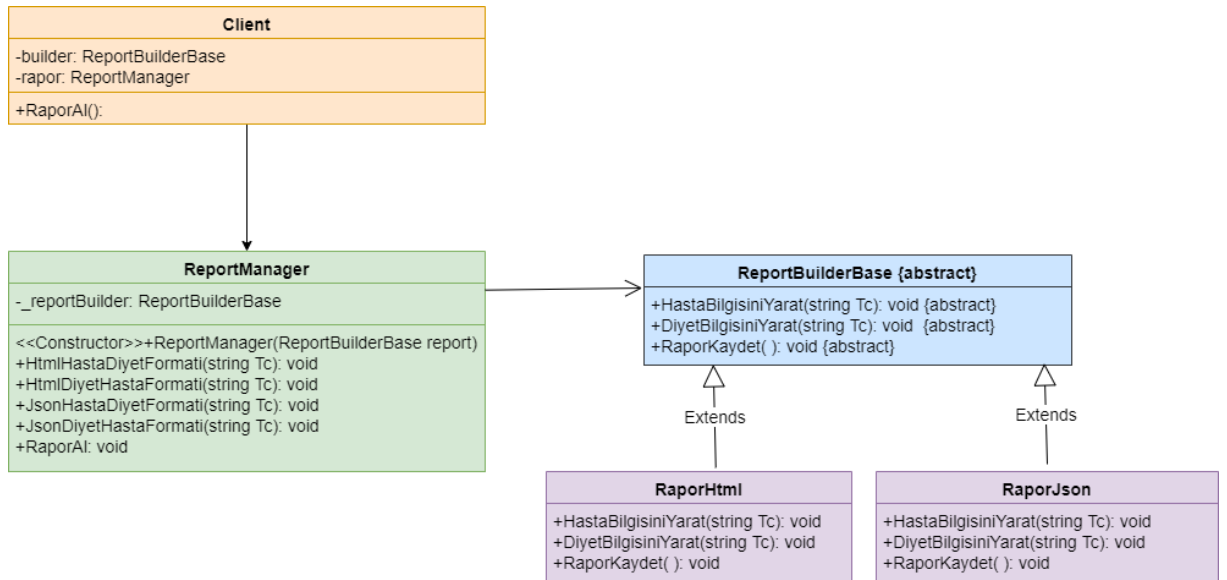
2. PROJE

2.1 Projenin UML Class Diyagramları

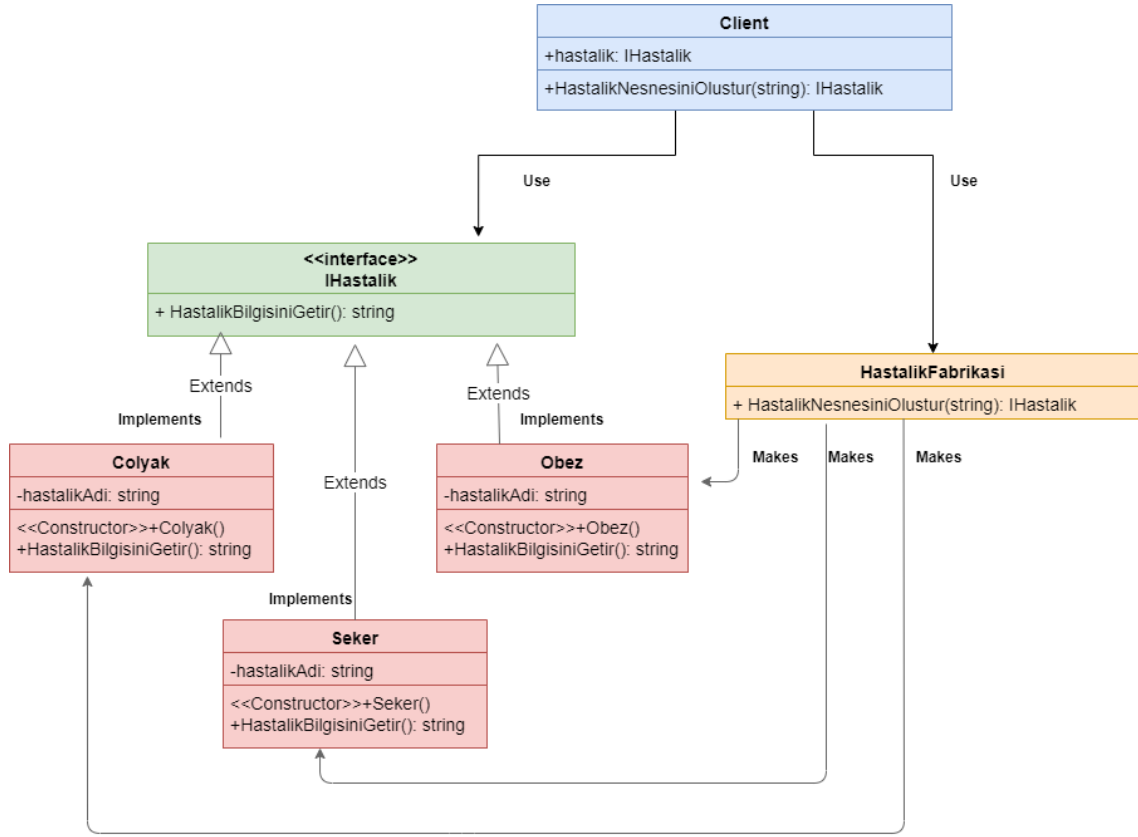
2.1.1 Bridge Tasarım Deseni-Diyet Yöntemi



2.1.2 Builder Tasarım Deseni-Rapor Alma



2.1.3 Fabrika Tasarım Deseni - Hastalıklar



2.2 Projenin Ekran Görüntüleri

Giriş - X

Admin Giriş
Kullanıcı Adı: hydonmez
Şifre: ****
Giriş Yap

Diyetisyen Giriş
Kullanıcı Adı: tubaakbas
Şifre: ****
Giriş Yap

Rapor Alma Formu - X

TC Kimlik No: 21495586145
Rapor Formatı: HTML Hasta Bilgisi Diyet Bilgisi
Rapor Al

Diyetisyen Kayıt Formu - X

Ad: Melsan
Soyad: Yilmaz
Kullanıcı Ad: melsanyilmaz
Şifre: 1234
TC Kimlik No: 11223344556
Kayıt Ol

Hasta Kayıt Formu - X

Ad: Ali
Soyad: Keskinoglu
TC Kimlik No: 21495586145
Hastalık: Colyak Hastaligi
Diyet Yöntemi: Akdeniz
Kayıt Ol

Ana Menu - X

İşlem Seçiniz
Hasta Kaydet
Rapor Al

Hasta Bilgileri - Diyet Bilgileri Html

HASTA BİLGİLERİ

ADI	SOYADI	TC	HASTALIK ADI
Ali	Keskinoglu	21495586145	Colyak Hastaligi

DIYET BİLGİLERİ

DIYET YÖNTEMİ	PAZARTESİ	SALI	ÇARŞAMBA	PERŞEMBE	CUMA	CUMARTESİ	PAZAR
Akdeniz Diyeti	Sabah: 1 Adet Yumurta, 1 Dilim Az Yağlı Beyaz Peynir, Kepekli Ekmek, Domates, Salatalık Öğlen: Ton Balıklı Bol Yeşillikli Salata, 1 Dilim Kepekli Ekmek Akşam: Zeytinyağlı Sebze Yemeği, Yarım Yağlı Yoğurt,Salata	Sabah: 2 Dilim Beyaz Peynir, Zeytin, Kepek Ekmegi, Domates, Salatalık, Yeşillik Öğlen: Fırında Izgara Sebze, Makarna, Ayran Akşam: Zeytinyağlı Bakliyat Yemeği, Yarım Yağlı Yoğurt, Salata	Sabah: Yarım Yağlı Yoğurt, 3 Kaşık Yulaf Ezmesi, Meyve Öğlen: Peynirli Avokadolu Salata, Kepekli Ekmek Akşam: Izgara Balık, Bol Yeşillikli Salata	Sabah: Peynirli Omlet, Zeytin,Kepek Ekmegi, Yeşillik Öğlen: Zeytinyağlı Bakliyat Yemeği, Yarım Yağlı Yoğurt, Salata Akşam: Zeytinyağlı Sebze Yemeği, Bulgur Pilavı, Ayran	Sabah: Lor Peyniri, Ceviz, Kepek Ekmegi, Domates, Salatalık Öğlen: Haşlanmış Mercimekli Salata, Ayran Akşam: Zeytinyağlı Sebze Yemeği, Salata, Kepek Ekmegi	Sabah: Yarım Yağlı Süt, Fındık, 2 Yemek Kaşığı Yulaf Ezmesi, Meyve Öğlen: Sebzeli Makarna, Ayran, Salata Akşam: Izgara Tavuk, Yarım yağlı Yoğurt, Haşlama sebze	Sabah: Menemen, Yarım Yağlı Beyaz Peynir, Kepek Ekmegi, Yeşillik Öğlen: Tam Tahıl Ekmegine Yapılmış Sandviç, Ayran Akşam: Çorba, Zeytinyağlı Bakliyat Yemeği, Salata

Diyet Bilgileri - Hasta Bilgileri Html

DIYET BİLGİLERİ

DIYET YÖNTEMİ	PAZARTESİ	SALI	ÇARŞAMBA	PERŞEMBE	CUMA	CUMARTESİ	PAZAR
Akdeniz Diyeti	Sabah: 1 Adet Yumurta, 1 Dilim Az Yağlı Beyaz Peynir, Kepekli Ekmek, Domates, Salatalık Öğlen: Ton Balıklı Bol Yeşillikli Salata, 1 Dilim Kepekli Ekmek Akşam: Zeytinyağlı Sebze Yemeği, Yarım Yağlı Yoğurt,Salata	Sabah: 2 Dilim Beyaz Peynir, Zeytin, Kepek Ekmegi, Domates, Salatalık, Yeşillik Öğlen: Fırında Izgara Sebze, Makarna, Ayran Akşam: Zeytinyağlı Bakliyat Yemeği, Yarım Yağlı Yoğurt, Salata	Sabah: Yarım Yağlı Yoğurt, 3 Kaşık Yulaf Ezmesi, Meyve Öğlen: Peynirli Avokadolu Salata, Kepekli Ekmek Akşam: Izgara Balık, Bol Yeşillikli Salata	Sabah: Peynirli Omlet, Zeytin,Kepek Ekmegi, Yeşillik Öğlen: Zeytinyağlı Bakliyat Yemeği, Yarım Yağlı Yoğurt, Salata Akşam: Zeytinyağlı Sebze Yemeği, Bulgur Pilavı, Ayran	Sabah: Lor Peyniri, Ceviz, Kepek Ekmegi, Domates, Salatalık Öğlen: Haşlanmış Mercimekli Salata, Ayran Akşam: Zeytinyağlı Sebze Yemeği, Salata, Kepek Ekmegi	Sabah: Yarım Yağlı Süt, Fındık, 2 Yemek Kaşığı Yulaf Ezmesi, Meyve Öğlen: Sebzeli Makarna, Ayran, Salata Akşam: Izgara Tavuk, Yarım yağlı Yoğurt, Haşlama sebze	Sabah: Menemen, Yarım Yağlı Beyaz Peynir, Kepek Ekmegi, Yeşillik Öğlen: Tam Tahıl Ekmegine Yapılmış Sandviç, Ayran Akşam: Çorba, Zeytinyağlı Bakliyat Yemeği, Salata

HASTA BİLGİLERİ

ADI	SOYADI	TC	HASTALIK ADI
Ali	Keskinoglu	21495586145	Colyak Hastaligi

Hasta Bilgileri – Diyet Bilgileri Json

Rapor.json - Not Defteri

DosyaDüzenBiçimGörünümYardım

```
{
  "Ad": "Ali",
  "Soyad": "Keskinoglu",
  "Tc": "21495586145",
  "Hastalik": "Colyak Hastaligi",
  "DiyetYontemi": 1,
  "HastaId": 3
}

{
  "DiyetId": 1,
  "DiyetAdi": "Akdeniz Diyeti",
  "Pazartesi_Sabah": "1 Adet Yumurta, 1 Dilim Az Yağlı Beyaz Peynir, Kepekli Ekmek, Domates, Salatalık",
  "Pazartesi_Oglen": "Ton Balıklı Bol Yeşillikli Salata, 1 Dilim Kepekli Ekmek",
  "Pazartesi_Aksam": "Zeytinyağlı Sebze Yemeği, Yarım Yağlı Yoğurt,Salata",
  "Sali_Sabah": "2 Dilim Beyaz Peynir, Zeytin, Kepek Ekmegi, Domates, Salatalık, Yeşillik",
  "Sali_Oglen": "Fırında Izgara Sebze, Makarna, Ayran",
  "Sali_Aksam": "Zeytinyağlı Bakliyat Yemeği, Yarım Yağlı Yoğurt, Salata",
  "Carsamba_Sabah": "Yarım Yağlı Yoğurt, 3 Kaşık Yulaf Ezmesi, Meyve",
  "Carsamba_Oglen": "Peynirli Avokadolu Salata, Kepekli Ekmek",
  "Carsamba_Aksam": "Izgara Balık, Bol Yeşillikli Salata",
  "Persembe_Sabah": "Peynirli Omlet, Zeytin,Kepek Ekmegi, Yeşillik",
  "Persembe_Oglen": "Zeytinyağlı Bakliyat Yemeği, Yarım Yağlı Yoğurt, Salata",
  "Persembe_Aksam": "Zeytinyağlı Sebze Yemeği, Bulgur Pilavı, Ayran",
  "Cuma_Sabah": "Lor Peyniri, Ceviz, Kepek Ekmegi, Domates, Salatalık",
  "Cuma_Oglen": "Haşlanmış Mercimekli Salata, Ayran",
  "Cuma_Aksam": "Zeytinyağlı Sebze Yemeği, Salata, Kepek Ekmegi",
  "Cumartesi_Sabah": "Yarım Yağlı Süt, Fındık, 2 Yemek Kaşığı Yulaf Ezmesi, Meyve",
  "Cumartesi_Oglen": "Sebzeli Makarna, Ayran, Salata",
  "Cumartesi_Aksam": "Izgara Tavuk, Yarım yağlı Yoğurt, Haşlama sebze",
  "Pazar_Sabah": "Menemen, Yarım Yağlı Beyaz Peynir, Kepek Ekmegi, Yeşillik",
  "Pazar_Oglen": "Tam Tahıl Ekmegine Yapılmış Sandviç, Ayran",
  "Pazar_Aksam": "Çorba, Zeytinyağlı Bakliyat Yemeği, Salata"
}
```

Diyet Bilgileri - Hasta Bilgileri Json

Rapor.json - Not Defteri

DosyaDüzenBiçimGörünümYardım

```
{
  "DiyetId": 1,
  "DiyetAdi": "Akdeniz Diyeti",
  "Pazartesi_Sabah": "1 Adet Yumurta, 1 Dilim Az Yağlı Beyaz Peynir, Kepekli Ekmek, Domates, Salatalık",
  "Pazartesi_Oglen": "Ton Balıklı Bol Yeşillikli Salata, 1 Dilim Kepekli Ekmek",
  "Pazartesi_Aksam": "Zeytinyağlı Sebze Yemeği, Yarım Yağlı Yoğurt,Salata",
  "Sali_Sabah": "2 Dilim Beyaz Peynir, Zeytin, Kepek Ekmegi, Domates, Salatalık, Yeşillik",
  "Sali_Oglen": "Fırında Izgara Sebze, Makarna, Ayran",
  "Sali_Aksam": "Zeytinyağlı Bakliyat Yemeği, Yarım Yağlı Yoğurt, Salata",
  "Carsamba_Sabah": "Yarım Yağlı Yoğurt, 3 Kaşık Yulaf Ezmesi, Meyve",
  "Carsamba_Oglen": "Peynirli Avokadolu Salata, Kepekli Ekmek",
  "Carsamba_Aksam": "Izgara Balık, Bol Yeşillikli Salata",
  "Persembe_Sabah": "Peynirli Omlet, Zeytin,Kepek Ekmegi, Yeşillik",
  "Persembe_Oglen": "Zeytinyağlı Bakliyat Yemeği, Yarım Yağlı Yoğurt, Salata",
  "Persembe_Aksam": "Zeytinyağlı Sebze Yemeği, Bulgur Pilavı, Ayran",
  "Cuma_Sabah": "Lor Peyniri, Ceviz, Kepek Ekmegi, Domates, Salatalık",
  "Cuma_Oglen": "Haşlanmış Mercimekli Salata, Ayran",
  "Cuma_Aksam": "Zeytinyağlı Sebze Yemeği, Salata, Kepek Ekmegi",
  "Cumartesi_Sabah": "Yarım Yağlı Süt, Fındık, 2 Yemek Kaşığı Yulaf Ezmesi, Meyve",
  "Cumartesi_Oglen": "Sebzeli Makarna, Ayran, Salata",
  "Cumartesi_Aksam": "Izgara Tavuk, Yarım yağlı Yoğurt, Haşlama sebze",
  "Pazar_Sabah": "Menemen, Yarım Yağlı Beyaz Peynir, Kepek Ekmegi, Yeşillik",
  "Pazar_Oglen": "Tam Tahıl Ekmegine Yapılmış Sandviç, Ayran",
  "Pazar_Aksam": "Çorba, Zeytinyağlı Bakliyat Yemeği, Salata"
}

{
  "Ad": "Ali",
  "Soyad": "Keskinoglu",
  "Tc": "21495586145",
  "Hastalik": "Colyak Hastaligi",
  "DiyetYontemi": 1,
  "HastaId": 3
}
```

2.3 Proje Kodları

2.3.1 Bridge Tasarım Deseni için Kodlar

Client Sınıfı

```
public partial class HastaKayitFormu : Form
{
    //bridge tasarım deseni için DiyetYontemiManagerdan sınıfından bir nesne yaratılır
    DiyetYontemiAbstraction diyet = new DiyetYontemiManager();

    private void btnKayitOl_Click(object sender, EventArgs e)
    {
        HastaKayitManager hastaKayitManager = new HastaKayitManager();//hasta islemlerinin yapıldığı sınıftan bir nesne yaratılır

        if (!BosGecildiMi()) //Hicbir alan bos gecilmediyse hastayi ekler.
        {
            //cmbden secilen hastaliga gore hastalik nesnesi olusturulur ve hastaliga atanir
            hastalik = hastalikFabrikasi.HastalikNesnesiOlustur(cmbHastalik.Text);
            HastaTbl hasta = new HastaTbl //hasta sınıfından bir hasta nesnesi yaratılır
            {
                HastaAd = txtAd.Text,
                HastaSoyad = txtSoyad.Text,
                HastaTc = txtTc.Text,

                //57 satırda olusan hastaligin hastalikbilgisi getir metodu cagırılarak hastaligin ismi alınır
                Hastalik = hastalik.HastalikBilgisiniGetir(),
                HastaninDiyetisyeni = DiyetisyenGirisIslemleriManager.girisId,
                DiyetYontemId = diyet.DiyetIdGetir() //secilen diyeteye gore o diyetin id'si gelecek
            };

            //hasta kayit edilme islemi başarılı ise true geri donderilir ve kontrol metodu cagırılır
            Kontrol(hastaKayitManager.HastaEkle(hasta));
        }
    }

    private void cmbDiyetYontemi_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (cmbDiyetYontemi.SelectedItem == null)
        {
            return;
        }

        if (cmbDiyetYontemi.SelectedItem.ToString() == "Akdeniz")
        {
            diyet._diyetYontemiImplementor = new Akdeniz(); //diyetimplementor Akdeniz sınıfından bir nesne olarak tanımlanır
        }
        else if (cmbDiyetYontemi.SelectedItem.ToString() == "Gluten Free")
        {
            diyet._diyetYontemiImplementor = new GlutenFree();
        }

        else if (cmbDiyetYontemi.SelectedItem.ToString() == "Yesillikler Dunyasi")
        {
            diyet._diyetYontemiImplementor = new YesilliklerDunyasi();
        }
        else if (cmbDiyetYontemi.SelectedItem.ToString() == "Deniz Urunleri")
        {
            diyet._diyetYontemiImplementor = new DenizUrunleri();
        }
    }
}
```

DiyetYontemiAbstraction Sınıfı

```
abstract class DiyetYontemiAbstraction
{
    /*DiyetYontemiImplementor soyut sınıfının bir referansı yazıldı ve bu özellik sayesinde diğer
    sınıflarla iletişim kuruldu.(Bridge Tasarım Deseni)*/
    5 başvuru
    public DiyetYontemiImplementor _diyetYontemiImplementor { get; set; }
    //Alt sınıflarda kullanılarak ve veritabanındaki DiyetListesiTbl tablosundaki Id'sini getirmek için DiyetIdGetir() metodu yazıldı.
    2 başvuru
    public abstract int DiyetIdGetir();
}
```

DiyetYontemiManager Sınıfı

```
class DiyetYontemiManager : DiyetYontemiAbstraction
{
    2 başvuru
    public override int DiyetIdGetir()
    {
        /*DiyetYontemiAbstraction sınıfında bulunan DiyetIdGetir() metodu burada ezildi. */

        /*DiyetImplementor 'a Client sınıfta yaratılan nesne olduğu için hangi diyet sınıfından nesne türetilmişse o diyet
        sınıfının metodunu çağırır ve sonucu geri döndürür*/
        return _diyetYontemiImplementor.DiyetIdGetir();
    }
}
```

DiyetYontemiImplementor Sınıfı

```
public abstract class DiyetYontemiImplementor
{
    8 başvuru
    protected int DiyetId { get; set; } //Alt sınıflarda her diyet yontemi DiyetId özelliği ile ezilerek atama yapılması için kullanılır.
    5 başvuru
    public abstract int DiyetIdGetir();/*DiyetYontemiManager sınıfında bulunan DiyetIdGetir() metodunun içerisinde çağırıldı.
    Ayrıca bu metodumuz alt sınıflarında ezilerek Id'lerine ulaşıldı.*/
}
```

Akdeniz Sınıfı

```
public class Akdeniz : DiyetYontemiImplementor
{
    1 başvuru
    public Akdeniz()
    {
        //Sınıfın kurucu metodunda DiyetId'ye değer atıyoruz. DiyetListesiTbl'de Akdeniz Diyeti 1 numaralı Id'de tutuluyor
        DiyetId = 1;
    }

    //Üst sınıftaki DiyetIdGetir() metodunu ezdik ve Akdeniz diyetinin DiyetId'si return edildi
    5 başvuru
    public override int DiyetIdGetir()
    {
        return DiyetId;
    }
}
```


GlutenFree Sınıfı

```
public class GlutenFree : DiyetYontemiImplementor
{
    1 başvuru
    public GlutenFree()
    {
        //Sınıfın kurucu metodunda DiyetId'ye değer atıyoruz. DiyetListesiTbl'de Gluten Free Diyeti 2 numaralı Id'de tutuluyor
        DiyetId = 2;
    }

    //Üst sınıftaki DiyetIdGetir() metodunu ezdik ve Gluten Free Diyetinin DiyetId'si return edildi.
    5 başvuru
    public override int DiyetIdGetir()
    {
        return DiyetId;
    }
}
```

DenizUrunleri Sınıfı

```
public class DenizUrunleri: DiyetYontemiImplementor
{
    1 başvuru
    public DenizUrunleri()
    {
        //Sınıfın kurucu metodunda DiyetId'ye değer atıyoruz. DiyetListesiTbl'de Deniz Ürünleri Diyeti 3 numaralı Id'de tutuluyor
        DiyetId = 3;
    }

    //Üst sınıftaki DiyetIdGetir() metodunu ezdik ve Deniz Urunleri Diyetinin DiyetId'si return edildi
    5 başvuru
    public override int DiyetIdGetir()
    {
        return DiyetId;
    }
}
```

YesilliklerDunyasi Sınıfı

```
public class YesilliklerDunyasi : DiyetYontemiImplementor
{
    1 başvuru
    public YesilliklerDunyasi()
    {
        //Sınıfın kurucu metodunda DiyetId'ye değer atıyoruz. DiyetListesiTbl'de Yesillikler Dünyası Diyeti 4 numaralı Id'de tutuluyor
        DiyetId = 4;
    }

    //Üst sınıftaki DiyetIdGetir() metodunu ezdik ve Yesillikler Dünyası Diyetinin DiyetId'si return edildi
    5 başvuru
    public override int DiyetIdGetir()
    {
        return DiyetId;
    }
}
```

2.3.2 Fabrika Tasarım Deseni

Client Sınıfı

```
public partial class HastaKayitFormu : Form
{
    //bridge tasarım deseni için DiyetYontemiManagerdan sınıfından bir nesne yaratılır
    DiyetYontemiAbstraction diyet = new DiyetYontemiManager();

    //hastalık bilgisini almak ve fabrika desenini kullanmak için HastalıkFabrikasi'ndan bir nesne yaratılır
    HastalıkFabrikasi hastalikFabrikasi = new HastalikFabrikasi();

    IHastalik hastalik;//Hastalık fabrikası deseninde hastalıklara ulaşabilmek için IHastalik interfaceden bir özellik yaratılır

    private void btnKayitOl_Click(object sender, EventArgs e)
    {
        HastaKayitManager hastaKayitManager = new HastaKayitManager();//hasta işlemlerinin yapıldığı sınıftan bir nesne yaratılır

        if (!BosGecildiMi()) //Hicbir alan bos gecilmediyse hastayi ekler.
        {
            //cmbden secilen hastaliga gore hastalik nesnesi olusturulur ve hastaliga atanir
            hastalik = hastalikFabrikasi.HastalikNesnesiOlustur(cmbHastalik.Text);
            HastaTbl hasta = new HastaTbl //hasta sınıfından bir hasta nesnesi yaratılır
            {
                HastaAd = txtAd.Text,
                HastaSoyad = txtSoyad.Text,
                HastaTc = txtTc.Text,

                //57 satırda olusan hastaligin hastalikbilgisi getir metodu cagrilarak hastaligin ismi alinir
                Hastalik = hastalik.HastalikBilgisiniGetir(),
                HastaninDiyetisyeni = DiyetisyenGirisIslemleriManager.girisId,
                DiyetYontemId = diyet.DiyetIdGetir() //secilen diyete gore o diyetin id'si gelecek
            };

            //hasta kayit edilme islemi başarılı ise true geri donderilir ve kontrol metodu cagrilir
            Kontrol(hastaKayitManager.HastaEkle(hasta));
        }
    }
}
```

HastalikFabrikasi Sınıfı

```
public class HastalikFabrikasi //Fabrika Tasarım Deseni
{
    1 başvuru
    public IHastalik HastalikNesnesiOlustur(string hastalikAdi) //Gelen hastalık adına göre ilgili hastalığın nesnesini oluşturun
    {
        //İlgili nesneyi return eder.
        if(hastalikAdi == "Colyak Hastaligi")
        {
            return new Colyak();
        }
        else if (hastalikAdi == "Seker Hastaligi")
        {
            return new Seker();
        }
        else
        {
            return new Obez();
        }
    }
}
```

IHastalik Arayüzü

```
public interface IHastalik //Fabrika Tasarım Deseni
{
    4 başvuru
    string HastalikBilgisiniGetir();//Hastalık adına ulaşabilmek için interfacede bir metod tanımlandık.
}
```

Obez Sınıfı

```
class Obez : IHastalik //Fabrika Tasarım Deseni
{
    private string hastalikAdi;
    1 başvuru
    public Obez()
    {
        hastalikAdi = "Obez Hastaligi"; //Kurucu metodunda hastalikAdina atama yaptık
    }
    4 başvuru
    public string HastalikBilgisiniGetir()
    {
        return hastalikAdi; //Üst sınıfta bulunan metodu ezerek ilgili sınıftaki hastalık adı geri döndürülür.
    }
}
```

Colyak Sınıfı

```
public class Colyak: IHastalik //Fabrika Tasarım Deseni
{
    2 başvuru
    private string hastalikAdi { get; set; }
    1 başvuru
    public Colyak()
    {
        hastalikAdi = "Colyak Hastaligi"; //Kurucu metodunda hastalikAdina atama yaptık
    }
    4 başvuru
    public string HastalikBilgisiniGetir()
    {
        return hastalikAdi; //Üst sınıfta bulunan metodu ezerek ilgili sınıftaki hastalık adı geri döndürülür.
    }
}
```

Seker Sınıfı

```
class Seker : IHastalik //Fabrika Tasarım Deseni
{
    private string hastalikAdi;
    1 başvuru
    public Seker()
    {
        hastalikAdi = "Seker Hastaligi"; //Kurucu metodunda hastalikAdina atama yaptık
    }
    4 başvuru
    public string HastalikBilgisiniGetir()
    {
        return hastalikAdi; //Üst sınıfta bulunan metodu ezerek ilgili sınıftaki hastalık adı geri döndürülür.
    }
}
```

2.3.3 Builder Tasarım Deseni

Client Sınıfı

```
public partial class RaporFormu : Form
{
    private ReportBuilderBase builder; //ReportBuilder sınıfından bir özellik tanımlandı
    private ReportManager rapor; //ReportManager sınıfından bir özellik tanımlandı

    private void btnRaporAl_Click(object sender, EventArgs e)
    {
        if (KontrolEt())//Alanlar boş değilse
        {
            if (TcKontrol(txtTc.Text)) //Girilen Tc'ye sahip bir hasta varsa raporu alınır
            {
                rapor.RaporAl();//yaratılan nesnenin raporAl fonksiyonuna erişilir
            }
            else
            {
                //Girilen Tc'ye sahip bir hasta yoksa uyarı mesajı yazdırır
                MessageBox.Show("Girilen TC ile Eşleşen Bir Kayıt Bulunamadı");
            }
        }
    }

    private void cmbRaporTipi_SelectedIndexChanged(object sender, EventArgs e)
    {
        //İstenilen rapor formatına göre nesne yaratılır ve ilgili nesnesinin istenilen kombinasyonu fonksiyonu çağırılır.
        if (cmbRaporTipi.SelectedItem == null)
        {
            return; //nullsa geriye donderilir
        }

        if (cmbRaporTipi.SelectedItem.ToString() == "HTML Hasta Bilgisi Diyet Bilgisi")
        {
            builder = new RaporHtml(); //html formatında isteniliyorsa ilk önce html nesnesi yaratılır ve buildera atanır
            rapor = new ReportManager(builder);//rapor manager nesnesinin kurucu metoduna builder nesnesi gönderilir

            rapor.HtmlHastaDiyetFormati(txtTc.Text);//rapor nesnesinin htmlhastadiyet sıralaması şeklinde rapor veren formatı çağrılır
        }
        else if (cmbRaporTipi.SelectedItem.ToString() == "HTML Diyet Bilgisi Hasta Bilgisi")
        {
            builder = new RaporHtml();
            rapor = new ReportManager(builder);

            rapor.HtmlDiyetHastaFormati(txtTc.Text);
        }

        else if (cmbRaporTipi.SelectedItem.ToString() == "JSON Diyet Bilgisi Hasta Bilgisi")
        {
            builder = new RaporJson();
            rapor = new ReportManager(builder);
            rapor.JsonDiyetHastaFormati(txtTc.Text);
        }
        else if (cmbRaporTipi.SelectedItem.ToString() == "JSON Hasta Bilgisi Diyet Bilgisi")
        {
            builder = new RaporJson();
            rapor = new ReportManager(builder);
            rapor.JsonHastaDiyetFormati(txtTc.Text);
        }
    }
}
```

ReportManager Sınıfı

```
public class ReportManager //Builder Tasarım Deseni - Client sadece bu sınıfı tanır.
{
    //Abstract sınıf olan ReportBuilderBase sınıfından bir özellik oluşturuyoruz.
    private ReportBuilderBase _reportBuilder;

    4 başvuru
    public ReportManager(ReportBuilderBase report) //Kurucu metodu ReportBuilderBase tipinde bir parametre alır
    {
        _reportBuilder = report;//gelen parametre _reportBuilder'a atanır.
    }

    1 başvuru
    public void HtmlHastaDiyetFormati(string Tc)//İlk önce Hasta Bilgileri daha sonra Diyet Bilgileri html formatında getirilir.
    {
        _reportBuilder.HastaBilgisiniYarat(Tc);//_reportBuilder'dan dönen diyet bilgisi rapora eklenir.
        _reportBuilder.DiyetBilgisiniYarat(Tc);
    }

    1 başvuru
    public void HtmlDiyetHastaFormati(string Tc)//İlk önce Diyet Bilgileri daha sonra Hasta Bilgileri html formatında getirilir.
    {
        _reportBuilder.DiyetBilgisiniYarat(Tc);
        _reportBuilder.HastaBilgisiniYarat(Tc);
    }

    public void JsonHastaDiyetFormati(string Tc)//İlk önce Hasta Bilgileri daha sonra Diyet Bilgileri json formatında getirilir.
    {
        _reportBuilder.HastaBilgisiniYarat(Tc);
        _reportBuilder.DiyetBilgisiniYarat(Tc);
    }

    1 başvuru
    public void JsonDiyetHastaFormati(string Tc)//İlk önce Diyet Bilgileri daha sonra Diyet Hasta json formatında getirilir.
    {
        _reportBuilder.DiyetBilgisiniYarat(Tc);
        _reportBuilder.HastaBilgisiniYarat(Tc);
    }

    1 başvuru
    public void RaporAl()
    {
        //Soyut sınıfta bulunan (ReportBuilderBase) RaporKaydet() metoduna erişilir ve uygun formatta Rapor alınır
        _reportBuilder.RaporKaydet();
    }
}
```

ReportBuilderBase Soyut Sınıfı

```
public abstract class ReportBuilderBase //Builder Tasarım Deseni
{
    //veritabanına erişebilmek için veritabanından bir nesne yaratıldı
    protected DiyetisyenVeriTabaniEntities veriTabani = new DiyetisyenVeriTabaniEntities();
    //StringBuilder kütüphanesinden bir nesne yaratıldı oluşan rapor bilgilerini bir arada tutmak için yaratıldı
    protected StringBuilder sb = new StringBuilder();

    /*ReportManager sınıfında bu üç metoda erişilir
    Alt sınıflarda ezilerek diyet bilgisi ve hasta bilgisi yaratılır tc parametre olarak alınır*/
    6 başvuru
    public abstract void HastaBilgisiniYarat(string Tc);
    6 başvuru
    public abstract void DiyetBilgisiniYarat(string Tc);
    3 başvuru
    public abstract void RaporKaydet(); //Raporu kaydetmek için gerekli işlemleri yapar
}
```

RaporHtml Sınıfı

```
class RaporHtml : ReportBuilderBase //Builder Tasarım Deseni
{
    //Diyet ile ilgili tüm bilgiler oluşturulur ve parametre olarak hasta TC'si alınır
    6 başvuru
    public override void DiyetBilgisiniYarat(string Tc) //Diyet bilgileri yaratılır.
    {
        sb.Append("<h3>DİYET BİLGİLERİ </h3>"); //Tablonun başlığı yazıldı
        //Tablo görünümü ayarlandı.
        sb.Append("<table class=\"display\" id=\"dt\" style=\"align: left; width: 99%;\" ");
        sb.Append("<rules=\"all\" border=\"1px\" ");
        sb.Append("<cellspacing=\"0px\" cellpadding=\"4px\">");

        sb.Append("<thead>");
        sb.Append("<tr></tr>");
        sb.Append("<tr style=\"background-color: Silver; color: #212121;\" ");
        sb.Append("<font-weight: bold\">");

        sb.Append("<tr> <th>DİYET YÖNTEMİ</th><th>PAZARTESİ</th><th>SALI</th><th>ÇARŞAMBA</th><th>PERŞEMBE\" +
            "</th><th>CUMA</th><th>CUMARTESİ</th><th>PAZAR</th></tr>"); //Tablonun sütun isimleri yazıldı

        //Eşleşen kayıt getirildi ve DiyetId için join yapıldı.
        var rows2 = from diyetbilgi in veriTabani.DiyetListesiTbl
                    join hastabilgi in veriTabani.HastaTbl on diyetbilgi.DiyetId equals hastabilgi.DiyetYontemId

        //Eşleşen kayıt ile ilgili bilgiler tabloya getirildi
        let row = "</td><td>" + diyetbilgi.DiyetAdi + "</td><td>"

        + "<b>Sabah: </b>" + diyetbilgi.Pazartesi_Sabah + "</br>" + "<b>Öğlen: </b>" + diyetbilgi.Pazartesi_Oglen
        + "</br>" + "<b>Akşam: </b>" + diyetbilgi.Pazartesi_Aksam + "</td><td>"

        + "<b>Sabah: </b>" + diyetbilgi.Sali_Sabah + "</br>" + "<b>Öğlen: </b>" + diyetbilgi.Sali_Oglen + "</br>" + "<b>Akşam: </b>"
        + diyetbilgi.Sali_Aksam + "</td><td>"

        + "<b>Sabah: </b>" + diyetbilgi.Carsamba_Sabah + "</br>" + "<b>Öğlen: </b>" + diyetbilgi.Carsamba_Oglen
        + "</br>" + "<b>Akşam: </b>" + diyetbilgi.Carsamba_Aksam + "</td><td>"

        + "<b>Sabah: </b>" + diyetbilgi.Persembe_Sabah + "</br>" + "<b>Öğlen: </b>" + diyetbilgi.Persembe_Oglen
        + "</br>" + "<b>Akşam: </b>" + diyetbilgi.Persembe_Aksam + "</td><td>"

        + "<b>Sabah: </b>" + diyetbilgi.Cuma_Sabah + "</br>" + "<b>Öğlen: </b>" + diyetbilgi.Cuma_Oglen
        + "</br>" + "<b>Akşam: </b>" + diyetbilgi.Cuma_Aksam + "</td><td>"

        + "<b>Sabah: </b>" + diyetbilgi.Cumartesi_Sabah + "</br>" + "<b>Öğlen: </b>" + diyetbilgi.Cumartesi_Oglen
        + "</br>" + "<b>Akşam: </b>" + diyetbilgi.Cumartesi_Aksam + "</td><td>"

        + "<b>Sabah: </b>" + diyetbilgi.Pazar_Sabah + "</br>" + "<b>Öğlen: </b>" + diyetbilgi.Pazar_Oglen
        + "</br>" + "<b>Akşam: </b>" + diyetbilgi.Pazar_Aksam
        where hastabilgi.HastaTc == Tc select row;

        //Kayıtlar tabloya aktarıldı
        rows2.ToList().ForEach(row => sb.Append("<tr>" + row + "</tr>"));
        sb.Append("</table>"); //Tablo kapatıldı
    }

    public override void HastaBilgisiniYarat(string Tc) //Hasta bilgileri yaratıldı
    {
        sb.Append("<h3>HASTA BİLGİLERİ </h3>"); //Tablonun başlığı yazıldı
        //Tablo görünümü ayarlandı.
        sb.Append("<table class=\"display\" id=\"dt\" style=\"align: left; width: 99%;\" ");
        sb.Append("<rules=\"all\" border=\"1px\" ");
        sb.Append("<cellspacing=\"0px\" cellpadding=\"4px\">");

        sb.Append("<thead>");
        sb.Append("<tr></tr>");
        sb.Append("<tr style=\"background-color: Silver; color: #212121;\" ");
        sb.Append("<font-weight: bold\">");

        sb.Append("<tr><th> ADI </th><th> SOYADI </th><th>TC</th> <th> HASTALIK ADI </th>"); //Sutun isimleri yazıldı
        //Eşleşen kayıt için ilgili bilgiler tabloya getirildi
        var rows = from gecici in veriTabani.HastaTbl
                    let row = "<td>" + gecici.HastaAd + "</td><td>" + gecici.HastaSoyad + "</td><td>" + gecici.HastaTc
                    + "</td><td>" + gecici.Hastalik
                    where gecici.HastaTc == Tc
                    select row;

        //getirilen liste ilgili alanlarda tabloya yazıldı.
        rows.ToList().ForEach(row => sb.Append("<tr>" + row + "</tr>"));
        sb.Append("</table>"); //Tablo kapatıldı
    }
}
```

```

public override void RaporKaydet()
{
    //Dosyaya tüm bilgileri yazılır ve dosya Rapor.html formatında oluşturulur.
    System.IO.File.WriteAllText(@"Rapor.html", sb.ToString()); //Dosya çalışma konuma kayıt edilir.
    //Uyarı mesajı yazdırılır.
    MessageBox.Show("Rapor Oluşturuldu Dosyanızı Kontrol Ediniz!");
}

```

RaporJson Sınıfı

```

class DiyetBilgisi //Diyet bilgilerini json'a yazmak için oluşturuldu
{
    1 başvuru
    public int DiyetId { get; set; }
    1 başvuru
    public string DiyetAdi { get; set; }
    1 başvuru
    public string Pazartesi_Sabah { get; set; }
    1 başvuru
    public string Pazartesi_Oglen { get; set; }
    1 başvuru
    public string Pazartesi_Aksam { get; set; }
    1 başvuru
    public string Salı_Sabah { get; set; }
    1 başvuru
    public string Salı_Oglen { get; set; }
    1 başvuru
    public string Salı_Aksam { get; set; }
    1 başvuru
    public string Carsamba_Sabah { get; set; }
    1 başvuru
    public string Carsamba_Oglen { get; set; }
    1 başvuru
    public string Carsamba_Aksam { get; set; }
    1 başvuru
    public string Persembe_Sabah { get; set; }
    1 başvuru
    public string Persembe_Oglen { get; set; }
    1 başvuru
    public string Persembe_Aksam { get; set; }

    public string Cuma_Sabah { get; set; }
    1 başvuru
    public string Cuma_Oglen { get; set; }
    1 başvuru
    public string Cuma_Aksam { get; set; }
    1 başvuru
    public string Cumartesi_Sabah { get; set; }
    1 başvuru
    public string Cumartesi_Oglen { get; set; }
    1 başvuru
    public string Cumartesi_Aksam { get; set; }
    1 başvuru
    public string Pazar_Sabah { get; set; }
    1 başvuru
    public string Pazar_Oglen { get; set; }
    1 başvuru
    public string Pazar_Aksam { get; set; }
}

```

```

class HastaBilgisi //Hasta bilgilerini json'a yazmak için oluşturuldu
{
    1 başvuru
    public string Ad { get; set; }
    1 başvuru
    public string Soyad { get; set; }
    1 başvuru
    public string Tc { get; set; }
    1 başvuru
    public string Hastalik { get; set; }
    1 başvuru
    public int DiyetYontemi { get; set; }
    1 başvuru
    public int HastaId { get; set; }
}

class RaporJson : ReportBuilderBase //Builder Tasarım Deseni
{
    //Üst sınıfta bulunan metot ezilerek Diyet Bilgisi yaratılarak json için hazırlandı
    6 başvuru
    public override void DiyetBilgisiniYarat(string Tc)
    {
        int diyetId; //istenilen diyet yönteminin Id'sinin tutulması için tanımlandı

        //Metotdan gelen hasta tc'si ile veritabanında bulunan hasta getirilir
        var hasta = from hastabilgisi in veriTabani.HastaTbl where hastabilgisi.HastaTc == Tc select hastabilgisi;
        DiyetBilgisi diyetBilgisi = new DiyetBilgisi(); //DiyetBilgisi sınıfından bir nesne üretildi json için

        foreach (var item in hasta)//eşleşen tc bilgisine ait hastanın diyet bilgilerine erişilir.
        {
            diyetId = item.DiyetYontemId;

            //İlgili diyet bilgileri sonuc'a aktarılır.
            var sonuc = from gecici in veriTabani.DiyetListesiTbl where gecici.DiyetId == diyetId select gecici;

            foreach (var item2 in sonuc)//sonuc listesi üzerinden dönülerek diyet bilgileri diyetBilgisi nesnesine aktarılır.
            {
                diyetBilgisi.DiyetAdi = item2.DiyetAdi;
                diyetBilgisi.DiyetId = item2.DiyetId;

                diyetBilgisi.Pazartesi_Sabah = item2.Pazartesi_Sabah;
                diyetBilgisi.Pazartesi_Oglen = item2.Pazartesi_Oglen;
                diyetBilgisi.Pazartesi_Aksam = item2.Pazartesi_Aksam;

                diyetBilgisi.Salı_Sabah = item2.Salı_Sabah;
                diyetBilgisi.Salı_Oglen = item2.Salı_Oglen;
                diyetBilgisi.Salı_Aksam = item2.Salı_Aksam;

                diyetBilgisi.Carsamba_Sabah = item2.Carsamba_Sabah;
                diyetBilgisi.Carsamba_Oglen = item2.Carsamba_Oglen;
                diyetBilgisi.Carsamba_Aksam = item2.Carsamba_Aksam;

                diyetBilgisi.Persembe_Sabah = item2.Persembe_Sabah;
                diyetBilgisi.Persembe_Oglen = item2.Persembe_Oglen;
                diyetBilgisi.Persembe_Aksam = item2.Persembe_Aksam;

                diyetBilgisi.Cuma_Sabah = item2.Cuma_Sabah;
                diyetBilgisi.Cuma_Oglen = item2.Cuma_Oglen;
                diyetBilgisi.Cuma_Aksam = item2.Cuma_Aksam;

                diyetBilgisi.Cumartesi_Sabah = item2.Cumartesi_Sabah;
                diyetBilgisi.Cumartesi_Oglen = item2.Cumartesi_Oglen;
                diyetBilgisi.Cumartesi_Aksam = item2.Cumartesi_Aksam;

                diyetBilgisi.Pazar_Sabah = item2.Pazar_Sabah;
                diyetBilgisi.Pazar_Oglen = item2.Pazar_Oglen;
                diyetBilgisi.Pazar_Aksam = item2.Pazar_Aksam;
            }
        }
        sb.Append(JsonConvert.SerializeObject(diyetBilgisi)); //diyetBilgisi'ni json yazma formatına çevirir
    }
}

```



```

public override void HastaBilgisiniYarat(string Tc)//Hastalık ile ilgili bilgiler yaratılarak json formatına getirilir
{
    HastaBilgisi hastaBilgisi = new HastaBilgisi();/*HastaTbl'de bulunan özellikler HastaBilgisi sınıfına yazıldı sınıfın
                                                    nesnesi oluşturuldu ve HastaTbl'de bulunan özellikler json için bu nesneye atandı */

    //Metotdan gelen tc ile eşleşen hasta tc'si getirildi
    var sorgu = from gecici in veriTabani.HastaTbl where gecici.HastaTc == Tc select gecici;

    foreach (var item in sorgu) //Hasta bilgileri alındı
    {
        hastaBilgisi.HastaId=item.HastaId;
        hastaBilgisi.Hastalik = item.Hastalik;
        hastaBilgisi.DiyetYontemi = item.DiyetYontemId;
        hastaBilgisi.Ad = item.HastaAd;
        hastaBilgisi.Soyad = item.HastaSoyad;
        hastaBilgisi.Tc = item.HastaTc;
    }

    sb.Append ( JsonConvert.SerializeObject(hastaBilgisi));//hastaBilgisi'ni json yazma formatına çevirir
}

3 başvuru
public override void RaporKaydet()//Rapor kaydedilir.
{
    //Bilgiler Rapor.json dosyasına yazılır
    File.WriteAllText(@"Rapor.json", sb.ToString());//Dosya çalışma konuma kayıt edilir.
    //Uyarı mesajı yazdırılır.
    MessageBox.Show("Rapor Oluşturuldu Dosyanızı Kontrol Ediniz!");
}

```