

BİL372 DERSİ 1.UYGULAMA ÖDEVİ RAPORU

DR.TANSEL ÖZYER

Ders Akademisyeni | TOBB Ekonomi Ve Teknoloji
Üniversitesi Bilgisayar MÜHENDİSLİĞİ BÖLÜMÜ |
Söğütözü, ANKARA

GRUP 6

Ömer Faruk Polat 181101032
Safa ERGUN - 171101008
Ahmet Burak ÖZGÜNGÖRDÜ
- 141101063

17,11,2020

Sayın Dr. Tansel Özyer,

BİL372 – Veritabanı Sistemleri Dersi 1. Uygulama
Ödevi kapsamında 6. Grup olarak yapmış
olduğumuz IMECE veritabanı

ve veritabanımızın web arayüzünün açıklamalı
raporunu bu dosya içerisinde hazırladık. C# ve
SQLServer programlama dilleriyle hazırladığımız
ödevimizde MVC (Model View Controller) yapısını
kullandık. Grup Üyelerinin her biri bu 3 ana
başlığın gereksinimlerini karşılamak için ayrı ayrı
çalıştı ve Üyeler kodladıkları kısımların raporlarını
ayrı ayrı yazmak suretiyle tek bir ana raporda
birleştirdik.

1- GİRİŞ

Ödev MVC yapısında hazırlanmış olup temel olarak ürün, marka, organizasyon, üretici, tüketici öğelerini içeren bir ekosistem tarafından kullanılması amacıyla oluşturulan bir veritabanını içermektedir. Ödevin 3 ana katmanı Veritabanı Katmanı, İletişim Katmanı ve Web Arayüzü Katmanı olarak tanımlanacak olursa;

- 1- Veritabanı Katmanında veritabanında gerekli olan tablolar ve fonksiyonlar oluşturulmuş olup C# programlama dili desteğinde SQLServer veritabanı dili söz dizimleriyle kodlanmıştır.
- 2- İletişim Katmanında veritabanındaki her bir öge için C# dilinde modeller oluşturulmuş ve nesne yönelimli programlama yöntemleriyle web arayüzünde kullanılabilecek formata çevrilmiştir. Aynı zamanda bu katmanda kullanıcıdan gelen girdiler, yine oluşturulan modeller aracılığıyla veritabanına aktarılmış ve veritabanında gerekli güncellemelerin yapılması sağlanmıştır.
- 3- Web Arayüzü Katmanı kullanıcıyla yüzyüze gelen tek katman olup kolay kullanım amaçlanarak oluşturulmuştur. Ödevde gerekli olan bütün formların girdi alanları, veritabanından gelen bilgilerin yansıtıldığı alanlar ve veritabanının görsel bir benzetimi olan MENÜLER bu katmanda yapılmıştır. Web Arayüzü HTML web dili kullanılarak C# programlama dili yardımıyla oluşturulan ".cshtml" uzantılı dosyalardan oluşmaktadır.

Ödevin Veritabanı katmanını Ömer Faruk Polat,
İletişim Katmanını Safa Ergun,
Web Arayüzü Katmanını Ahmet Burak ÖZGÜNGÖRDÜ kodlamıştır.
Katmanların açıklamaları kodlayan üyeler tarafından ayrı ayrı yazılıp
birleştirilmiştir.

Ödevin GitHub Repository Bağlantı Adresi:

https://github.com/omerfarukpolat/Bil372_Odev1_Grup6.git *

*: Github Repository'sinde uygulama dosyaları,
ödevin hazırlanma sürecini gösteren istatistik dosyaları,
kullanıcı kurulum ve kullanma kılavuzu dosyası
bulunmaktadır.

The diagram is a hand-drawn Entity-Relationship (ER) model on a grid background. It consists of several entities, relationships, and attributes.

- Entities (Rectangles):**
 - PRODUCT**: Attributes include Postcode, M-System, M-Code, M-Name, Is Active, Short Name, Is Abstract, Category, and MANUAL.
 - MANUFACTURERS**: Attributes include ID, Name, Address, and Line.
 - COUNTRY**: Attributes include Code and Name.
 - ORGANISATIONS**: Attributes include ID, Name, Report Org, Is Abstract, Address, and Basic Info.
- Relationships (Diamonds):**
 - SPLIT**: Connects PRODUCT (1) and PRODUCT REPAIRS (M).
 - PRODUCT REPAIRS**: Connects PRODUCT and ORGANISATIONS. Attributes: Repair Name, Repair Date.
 - ROUTER IN**: Connects COUNTRY and ROUTER OUT. Attributes: Router ID, Distinct.
 - ROUTER OUT**: Connects COUNTRY and ROUTER IN. Attributes: Router ID, Distinct.
- Other Elements:**
 - FEATURES**: A rectangle connected to PRODUCT via a diamond relationship labeled "PRODUCT FEATURES". Attributes: Feature ID, Feature Name.
 - ALTERNATIVE ROUTES**: A diamond relationship connected to PRODUCT and ROUTER IN. Attributes: Alternative, Start, End, Systemic.
 - FLAT**: A rectangle connected to ROUTER IN and ROUTER OUT. Attributes: Router ID, Flat ID, Quantity.
 - QUANTITY**: A rectangle connected to ROUTER IN and ROUTER OUT. Attributes: ID, Quantity, Unit.

VERİTABANI MANTIKSAL TASARIMI

PRODUCT:

(M_Syscode, Mcode, M_Name, M_ShortName, M_ParentCode,
M_Abtract, M_Category, Is_Active)

FEATURES:

(FeatureID, FeatureName)

PRODUCT_FEATURES:

(M_Syscode, FeatureID, MinVal, MaxVal)

MANUFACTURERS:

(ManufacturerID, Name, Address, City, Country)

PRODUCT-BRANDS:

(BrandBarcode, M_Syscode, BrandName, ManufacturerID)

ORGANISATIONS:

(OrgID, OrgName, ParentOrg, Abstract, Address, City, District, Type)

BRAND_ORGS:

(LotID, BrandBarcode, Quantity, In, Out, Unit, Baseprice)

FLOW:

(Source LotID, Source OrgID, Target LotID, Target OrgID, BrandBarcode, Quantity,
FlowDate)

ALTERNATIVE BRANDS:

(BrandBarcode, M_Syscode, Alternative BrandBarcode, Alternative Syscode)

COUNTRY:

(CountryCode, CountryName)

COUNTRY_CITY:

(CountryCode, CityID, CityName)

VERİTABANI KATMANI

```
24 başvuru
public class DatabaseController : Controller
{

    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["IMECE"].ConnectionString);

    0 başvuru
    public DatabaseController(char a)
    {
        con.Open();

        var cmd = new SqlCommand();
        cmd.Connection = con;

        cmd.CommandText = "DROP TABLE IF EXISTS ALTERNATIVE_BRANDS";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS FLOW";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS BRAND_ORGS";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS ORGANISATIONS";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS PRODUCT_BRANDS";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS MANUFACTURERS";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS COUNTRY_CITY";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS COUNTRY";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS PRODUCT_FEATURES";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS FEATURES";
        cmd.ExecuteNonQuery();
        cmd.CommandText = "DROP TABLE IF EXISTS PRODUCT";
        cmd.ExecuteNonQuery();
    }
}
```

Öncelikle SQLServer Management Studio aracılığıyla oluşturduğum veritabanını Visual Studio projeme bağladım. Ödevde DatabaseController.cs dosyası üzerinde çalıştım ve veritabanının BÜTÜN fonksiyonları BÜTÜN projede bu sınıf içinden çağırılıp kullanılmaktadır. Construction metodun char ile çağırılması veritabanındaki BÜTÜN tabloları silmekte, integer ile çağırılması BÜTÜN tabloları oluşturmakta ve ülke, şehir girdilerini yapmaktadır.

```

81 cmd.CommandText = @"CREATE TABLE PRODUCT_FEATURES(
82     M_SYSCODE INTEGER,
83     FEATURE_ID INTEGER,
84     MINVAL FLOAT,
85     MAXVAL FLOAT,
86     primary key(MINVAL,MAXVAL),
87     FOREIGN KEY(M_SYSCODE) REFERENCES PRODUCT(M_SYSCODE) ON DELETE CASCADE,
88     FOREIGN KEY(FEATURE_ID) REFERENCES FEATURES(FEATURE_ID)
89 );";
90 cmd.ExecuteNonQuery();
91 Console.WriteLine("Table PRODUCT_FEATURES created");
92
93
94 cmd.CommandText = @"CREATE TABLE COUNTRY(
95     Country_Code CHAR(3) PRIMARY KEY NOT NULL,
96     Country_Name VARCHAR(50)
97 );";
98
99 cmd.ExecuteNonQuery();
100
101 Console.WriteLine("Table COUNTRY created");
102
103
104 cmd.CommandText = @"CREATE TABLE COUNTRY_CITY(
105     Country_Code CHAR(3),
106     CityID INT identity(1,1) NOT NULL PRIMARY KEY,
107     City_Name VARCHAR(100),
108     FOREIGN KEY(Country_Code) REFERENCES COUNTRY(Country_Code) ON DELETE CASCADE
109 );";
110
111 cmd.ExecuteNonQuery();
112
113 cmd.CommandText = "INSERT INTO COUNTRY(Country_Code , Country_Name) " +
114 "VALUES('DEU', 'Almanya')";
115 cmd.ExecuteNonQuery();
116
117 cmd.CommandText = "INSERT INTO COUNTRY(Country_Code , Country_Name) " +
118 "VALUES('USA', 'Amerika')";
119 cmd.ExecuteNonQuery();
120
121 cmd.CommandText = "INSERT INTO COUNTRY(Country_Code , Country_Name) " +

```

Bütün tablolar ve tabloların değişkenleri ödev metnine bağlı kalınarak oluşturulmuştur. Primary Key'ler ve Foreign Key'ler veritabanı mantıksal tasarımına uygun olarak tanımlanmıştır.

```

317 8 başvuru
318 public List<PRODUCT> getProduct()
319 {
320     List<PRODUCT> listProduct = new List<PRODUCT>();
321     string sql = "SELECT * FROM PRODUCT";
322     using var asd = new SqlCommand(sql, con);
323     using SqlDataReader rdr = asd.ExecuteReader();
324     while (rdr.Read())
325     {
326         PRODUCT p = new PRODUCT();
327         p.M_SYSCODE = rdr.GetInt32(0);
328         p.M_CODE = rdr.GetString(1);
329         p.M_NAME = rdr.GetString(2);
330         p.M_SHORTNAME = rdr.GetString(3);
331         p.M_PARENTCODE = rdr.GetString(4);
332         p.M_ABSTRACT = rdr.GetBoolean(5);
333         p.M_CATEGORY = rdr.GetString(6);
334         p.IS_ACTIVE = rdr.GetBoolean(7);
335         listProduct.Add(p);
336     }
337     return listProduct;
338 }

```

GET fonksiyonlarında kullanıcıya gösterilecek olan veriler Model yapısındaki veritabanı ögesi modelleri aracılığıyla ilgili sınıflara gönderilmiştir.


```

517 public void insertProduct(string code, string name, string shortname, string parentcode, bool isAbstract, string category, bo
1 başvuru
533 public void insertFeatures(string featureName) {...}
1 başvuru
543 public void insertProductFeatures(int syscode, int featureid, float minval, float maxval) {...}
0 başvuru
556 public void insertCountry(string countryCode, string countryName) {...}
0 başvuru
567 public void insertCountryCity(string countryCode, string cityName) {...}
1 başvuru
578 public void insertManufacturers(string manufacturerName, string address, int city, string countryCode) {...}
1 başvuru
591 public void insertProductBrands(int manufacturerid, int syscode, string brandBarcode, string brandName) {...}
4 başvuru
604 public void insertOrganisations(string name, int parentOrg, bool isAbstract, string address, int cityid, string district, int
1 başvuru
620 public void insertBrandOrgs(int orgid, string brandBarcode, float unit, float baseprice, float quantity, float innn, float ou
0 başvuru
636 public void insertAlternativeBrands(string brandBarcode, int syscode, float altBrandBarcode, float altSysCode) {...}
1 başvuru
649 public void insertINFlow(int sourceorgid, string brandBarcode, float quantity, DateTime flowDate)
650 {
651     int lotId;
652     var cmd = new SqlCommand();
653     cmd.Connection = con;
654     string sql = "SELECT LOT_ID FROM BRAND_ORGS WHERE ORG_ID = @sourceorgid AND BRAND_BARCODE = @brandBarcode";
655
656     using var asd = new SqlCommand(sql, con);
657     asd.Parameters.AddWithValue("@sourceorgid", sourceorgid);
658     asd.Parameters.AddWithValue("@quantity", quantity);
659     asd.Parameters.AddWithValue("@brandBarcode", brandBarcode);
660     asd.Parameters.AddWithValue("@flowDate", flowDate);
661     using SqlDataReader rdr = asd.ExecuteReader();
662     while (rdr.Read())
663     {
664         lotId = rdr.GetInt32(0);
665         asd.Parameters.AddWithValue("@lotId", lotId);
666     }
667     rdr.Close();
668     string s = "INSERT INTO FLOW(Source_LOT_ID, Source_ORG_ID, Target_LOT_ID, Target_ORG_ID, BRAND_BARCODE, QUANTITY, FlowDate)
669         VALUES (@lotId, @sourceorgid, @lotId, @sourceorgid, @brandBarcode, @quantity, @flowDate)";
670
671     asd.CommandText = s;
672     asd.ExecuteNonQuery();

```

Her tablo için gerekli CRUD fonksiyonları oluşturulmuştur. Fonksiyonların parametreleri SQL parametresinde DÖNÜŞTÜRÜLÜP veritabanında ilgili yerlere yerleştirilmiştir.

```

2.basvuru
public void deleteFromOrganisations(int orgid, int deleteStyle)
{
    var cmd = new SqlCommand();
    cmd.Connection = con;
    cmd.Parameters.AddWithValue("@orgid", orgid);
    if (deleteStyle == 1)
    {
        string s2 = "UPDATE ORGANISATIONS SET PARENT_ORG = (SELECT PARENT_ORG FROM ORGANISATIONS WHERE ORG_ID = @orgid) " +
            "WHERE PARENT_ORG = (SELECT ORG_ID FROM ORGANISATIONS WHERE ORG_ID = @orgid)";
        cmd.CommandText = s2;
        cmd.ExecuteNonQuery();
        string s = "DELETE FROM ORGANISATIONS WHERE ORG_ID=@orgid";
        cmd.CommandText = s;
        cmd.ExecuteNonQuery();
    }
    else
    {
        string sql = "SELECT * FROM ORGANISATIONS WHERE PARENT_ORG = @orgid";
        using var asd = new SqlCommand(sql, con);
        asd.Parameters.AddWithValue("@orgid", orgid);
        using SqlDataReader rdr = asd.ExecuteReader();
        while (rdr.Read())
        {
            int childOrgid = rdr.GetInt32(0);
            string s3 = "DELETE FROM ORGANISATIONS WHERE ORG_ID = " + childOrgid;
            asd.Parameters.AddWithValue("@childOrgid", childOrgid);
            asd.CommandText = s3;
            rdr.Close();
            asd.ExecuteNonQuery();
            deleteFromOrganisations(childOrgid, 0);
        }
        rdr.Close();
        string s = "DELETE FROM ORGANISATIONS WHERE ORG_ID=@orgid";
        cmd.CommandText = s;
        cmd.ExecuteNonQuery();
    }
}

```

Ödev formlarında belirtilen organizasyon silme işleminde üst organizasyon silindiğinde ona bağlanan BÜTÜN organizasyonların silinmesi için özyinelemeli bir fonksiyon kullanılmıştır. Aynı özellik ÜRÜN silme işleminde de mevcuttur.

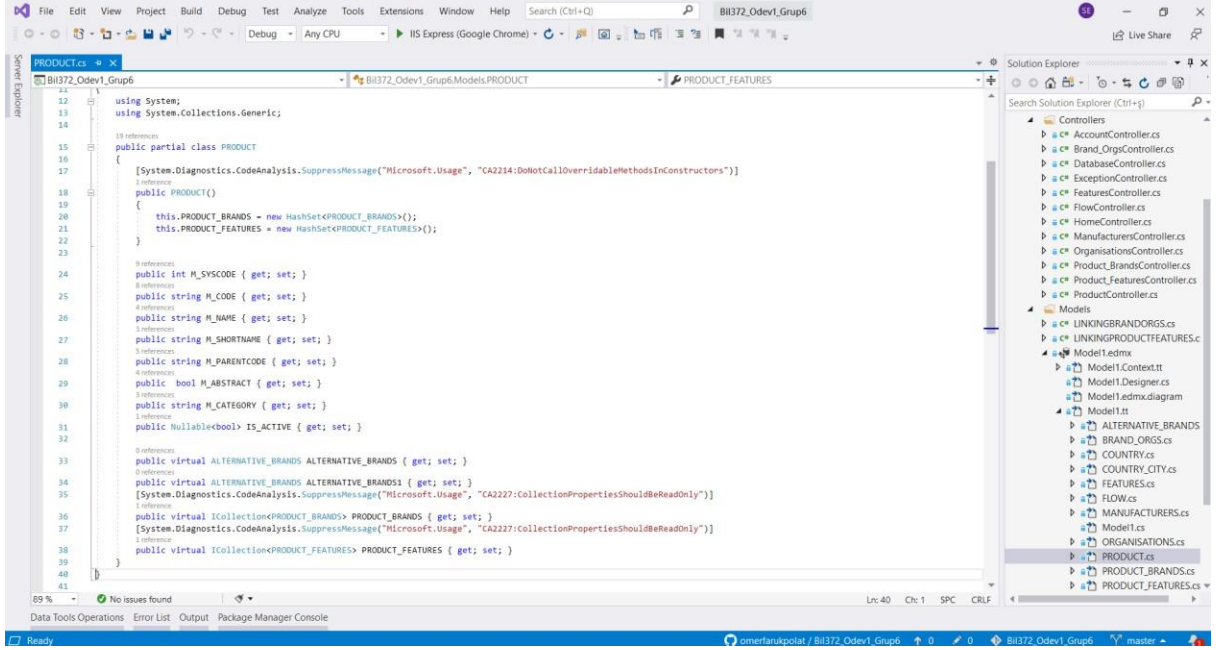

```

1001 public List<LINKINGBRANDORGS> searchWithOrgName(string orgname)
1027 public List<LINKINGBRANDORGS> searchWithBrandName(string brandname)
1053 public List<LINKINGPRODUCTFEATURES> searchWithProductName(string productname)
1054 {
1055     List<LINKINGPRODUCTFEATURES> productNames = new List<LINKINGPRODUCTFEATURES>();
1056     string sql = "SELECT M_CODE ,M_NAME, M_CATEGORY ,FEATURE_NAME , MINVAL , MAXVAL FROM " +
1057         "PRODUCT,FEATURES,PRODUCT_FEATURES WHERE " +
1058         "PRODUCT.M_NAME LIKE " + "'" + productname + "'" + " AND " +
1059         "PRODUCT.M_SYSCODE = PRODUCT_FEATURES.M_SYSCODE AND " +
1060         "FEATURES.FEATURE_ID = PRODUCT_FEATURES.FEATURE_ID";
1061     using var asd = new SqlCommand(sql, con);
1062     // asd.Parameters.AddWithValue("@brandname", ("'" + brandname + "'"));
1063     using SqlDataReader rdr = asd.ExecuteReader();
1064     while (rdr.Read())
1065     {
1066         LINKINGPRODUCTFEATURES b = new LINKINGPRODUCTFEATURES();
1067         b.M_CODE = rdr.GetString(0);
1068         b.M_NAME = rdr.GetString(1);
1069         b.M_CATEGORY = rdr.GetString(2);
1070         b.FEATURE_NAME = rdr.GetString(3);
1071         b.MINVAL = (float)rdr.GetDouble(4);
1072         b.MAXVAL = (float)rdr.GetDouble(5);
1073         productNames.Add(b);
1074     }
1075     return productNames;
1076 }
1077 public List<LINKINGPRODUCTFEATURES> searchWithFeature(string featurename)
1101 }
1102 }
1103 }

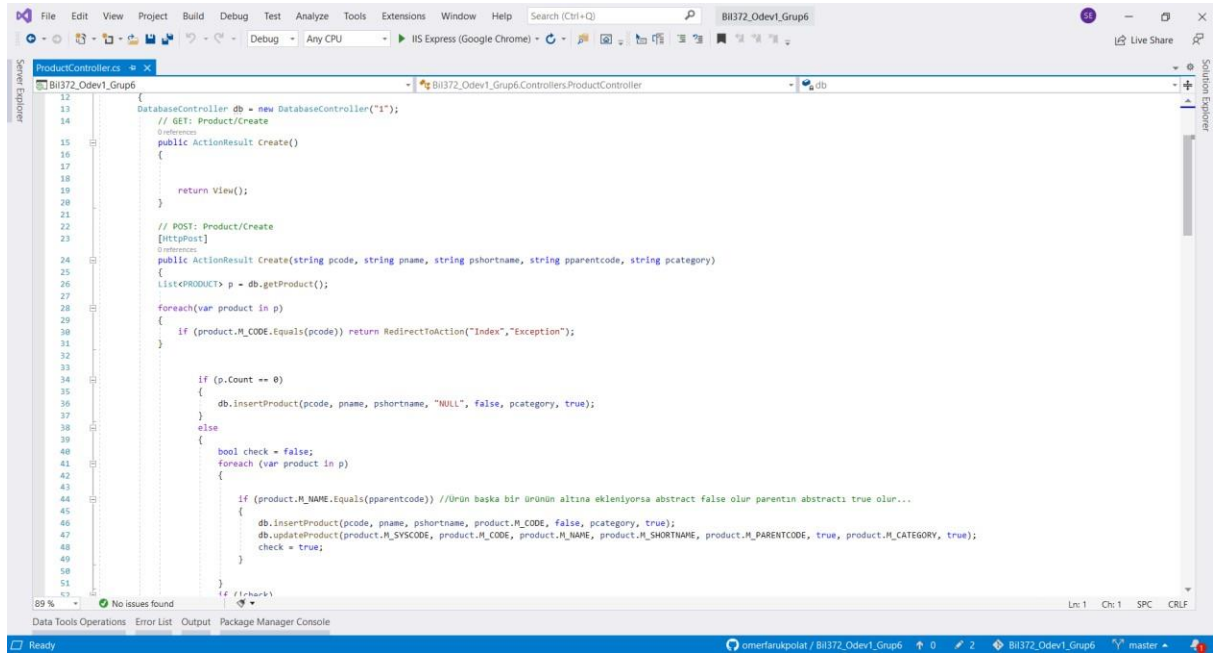
```

Kullanıcı dostu bir program yapmayı amaçladığımız için arama işleminde marka adı girildiğinde o markaya sahip olan BÜTÜN organizasyon isimleri ve markanın ilgili organizasyondaki özellikleri listelenmiştir. Aynı zamanda organizasyon adıyla arama yapıldığında o organizasyonun sahip olduğu markalar ve özellikleri listelenmiştir. Arama birebir eşleşme şeklinde yapılmamış, kullanıcının anahtar kelime girmesiyle birlikte o kelimenin geçtiği BÜTÜN ilgili ögeler sıralanmıştır.

İLETİŞİM KATMANI



Oluşturulan veritabanındaki her tablo için bir model oluşturuldu ve bu modelleri kullanarak veritabanında oluşturulan verileri bu modellere yerleştirildi. Yukarıda örnek bir PRODUCT modeli gösterilmiştir...



```
12 {
13     DatabaseController db = new DatabaseController("1");
14     // GET: Product/Create
15     public ActionResult Create()
16     {
17     }
18     return View();
19 }
20
21 // POST: Product/Create
22 [HttpPost]
23 public ActionResult Create(string pcode, string pname, string pshortname, string pparentcode, string category)
24 {
25     List<PRODUCT> p = db.getProduct();
26     foreach (var product in p)
27     {
28         if (product.M_CODE.Equals(pcode)) return RedirectToAction("Index", "Exception");
29     }
30
31     if (p.Count == 0)
32     {
33         db.insertProduct(pcode, pname, pshortname, "NULL", false, category, true);
34     }
35     else
36     {
37         bool check = false;
38         foreach (var product in p)
39         {
40             if (product.M_NAME.Equals(pparentcode)) //Ürün başka bir ürünün altına ekleniyorsa abstract false olum parentin abstractı true olun...
41             {
42                 db.insertProduct(pcode, pname, pshortname, product.M_CODE, false, category, true);
43                 db.updateProduct(product.M_SYSCODE, product.M_CODE, product.M_NAME, product.M_SHORTNAME, product.M_PARENTCODE, true, product.M_CATEGORY, true);
44                 check = true;
45             }
46         }
47     }
48     if (!check)
49     {
50     }
51 }
```

Her controller'da bir tane veritabanı objesi oluşturarak veritabanı metotlarına ulaşıldı. Bu metotları kullanarak veritabanındaki verilere ulaşıldı ve bu veriler view'daki sayfalara gönderildi.

Aynı zamanda Viewdaki sayfalardan gelen veriler veritabanı metotlarını kullanarak veritabanına kaydedilmiş oldu.

Ekleme silme güncelleme işlemlerinde yapılması gereken kontroller burada yapıldı.

WEB ARAYÜZÜ KATMANI

Login

Username

Password

Giriş

Kayıt Ol

Sistemin ana açılış ekranı bu şekilde. Login yapmadan sayfalara erişim imkanı olmuyor. Veritabanı ilk site açıldığında yükleniyor.

Username = Organizasyon Adı

Password = Organizasyon Id

İlk durumda herhangi bir organizasyon olmadığı için Kayıt Ol üzerinden organizasyon oluşturmak gerekiyor.

Register

Organizasyon Adı

Organizasyon Adresi

Organizasyon Şehri

Organizasyon İlçesi

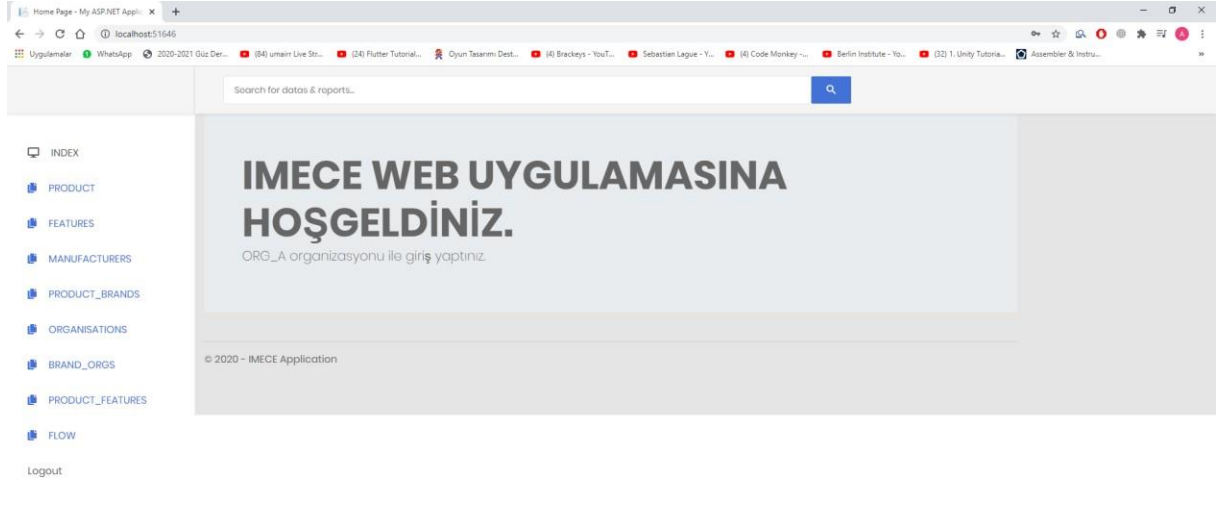
Organizasyon Tipi

Tedarikçi / Tüketici / Her ikisinde (0/1/2)

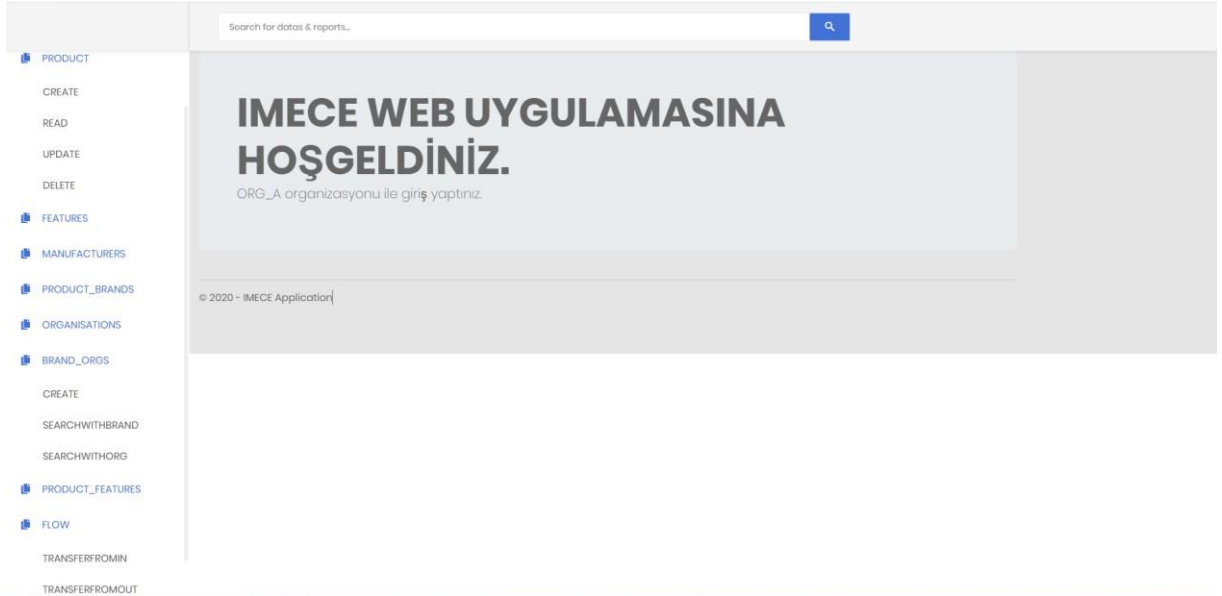
Oluştur

Burada girilen Organizasyon Adı Kullanıcı adı. Kayıt Ol üzerinden oluşturulan organizasyonların parentı yok.

Projede giriş ve kayıtlar için bir tablo tutulmayacağı için ek bilgiler alınmamıştır. İstendiği şekilde Organizasyon adı ve şifre ile giriş yapılmaktadır. Oluşturulan idnin organizasyon sahibinin mail ve telefonuna gönderildiği varsayılabilir.



Giriş yaptıktan sonra bu sayfa karşımıza çıkıyor. Sol taraftaki layoutu kullanarak istediğimiz işlemleri yapabiliyoruz. Örnek için...



Soldaki sayfalara geçiş gösterimi.

INDEX

PRODUCT

FEATURES

MANUFACTURERS

PRODUCT_BRANDS

ORGANISATIONS

BRAND_ORGS

PRODUCT_FEATURES

FLOW

Logout

Ürün Oluştur

Ürün Kodu

Ürün Adı

Ürün Kısa Adı

Ürün Üst Kodu

Kategori

Oluştur

© 2020 - IMECE Application

PRODUCT->CREATE

Bir oluşturma örneği..

Search for datas & reports...

© 2020 - IMECE Application

Sistem Kod	Ürün Kod	Ad	Kısa Ad	Üst Kod	Kategori
1	1	PRODUCT_1	P1	0	Eşya
2	2	PRODUCT_2	P2	0	Giyim
3	3	PRODUCT_3	P3	2	Kravat

INDEX

PRODUCT

FEATURES

MANUFACTURERS

PRODUCT_BRANDS

ORGANISATIONS

BRAND_ORGS

PRODUCT_FEATURES

FLOW

Logout

PRODUCT->READ

Bir okuma örneği...

Search for datas & reports...

Q

INDEX

PRODUCT

FEATURES

MANUFACTURERS

PRODUCT_BRANDS

ORGANISATIONS

BRAND_ORGS

PRODUCT_FEATURES

FLOW

Logout

Organizasyon Güncelle

Organizasyon Kodu

2

Organizasyon Adı

ORG_A

Üst Kodu

0

Organizasyon Adresi

ORG_A

Organizasyon Şehri

9

Organizasyon İlçesi

Çankaya

Organizasyon Tipi

2

Güncelle

ORGANISATIONS -> Update

Bir güncelleme örneği. Güncelleme sayfaları şu şekilde çalışıyor. O sayfadaki tablolarda bulunan primary key(burada Organizasyon Kodu(Org_id)) değeri girdiğiniz zaman onun bütün değerleri sayfaya dolduruluyor.

Burada Organizasyon Kodu = 2 girildiği anda diğer veriler otomatik dolduruldu. Daha sonra istediğimiz değeri değiştirip (Org Id hariç) güncelleyebiliyoruz.

Search for datas & reports...

Q

INDEX

PRODUCT

FEATURES

MANUFACTURERS

PRODUCT_BRANDS







ORGANISATIONS

BRAND_ORGS

PRODUCT_FEATURES

FLOW

Logout

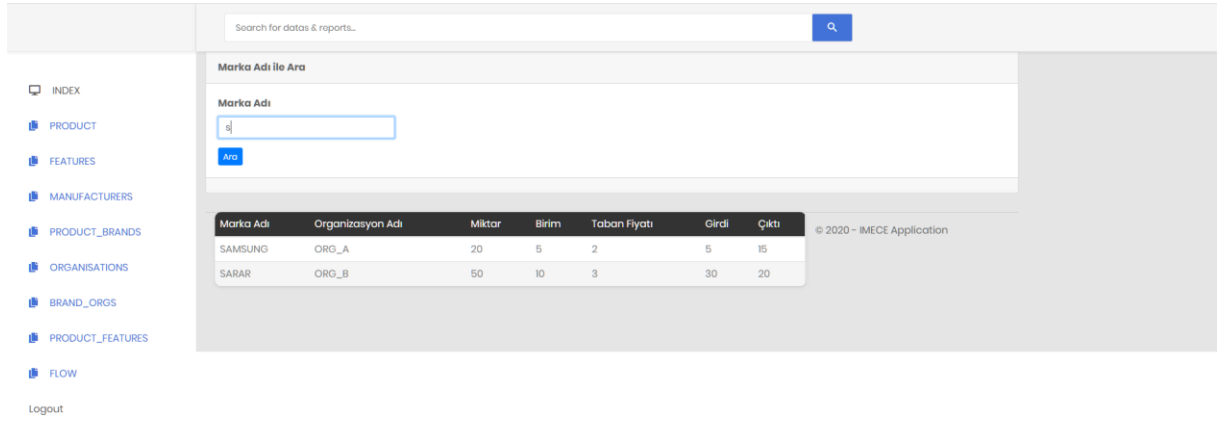
AD	ÜST ORGANİZASYON KOD	ADRES	ŞEHİR	İLÇE	TÜRÜ	
ORG_A	0	TÜRKİYE	9	Çankaya	2	 
ORG_B	0	TÜRKİYE	9	Çankaya	2	 
ORG_C	1	TÜRKİYE	9	Yenimahalle	1	 

© 2020 - İMECE Application

ORGANISATIONS -> Delete

Bir silme örneği. Product ve Organizasyon için 2 silme türü var. Diğer sayfalar için tek silme türü. Sağdaki çöp kutusu CASCADE DELETE soldaki kalem işareti ÜST BİRİME BAĞLAMA şeklinde siler.

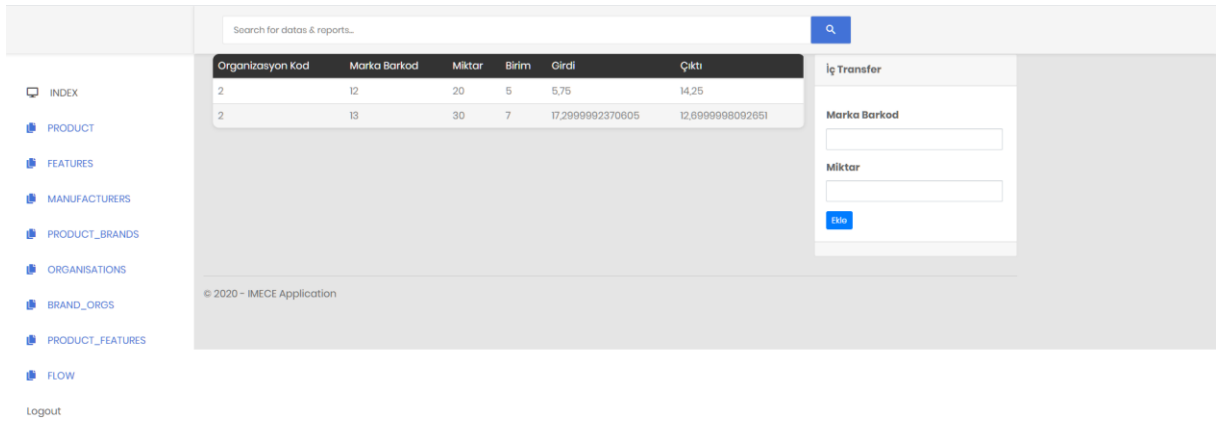
Simgelere basıldığında silme işlemi gerçekleşir.



BRAND_ORGS-> SEARCHWITHBRANDS

Bir arama örneği.

s ile aradığımızda SAMSUNG VE SARAR markalarını görebiliyoruz.



FLOW -> TRANSFERFROMIN

Bir transfer örneği. Değerleri girererek transferler yapabiliyoruz.

Not: Float değer alan miktar , girdi , çıktı , birim gibi değerleri girerken nokta(.) kullanarak işlem yapıyoruz.

Örnek : 5.25 şeklinde sayfalarda float değerlere atama

yapılmalıdır. Logout ile çıkış yapılabilir.

Giriş yapılmadan herhangi bir sayfa açılmaz.(logine yönlendirir)