VERİ TABANI PROJESİ

Şeyma Nur MUTLU ve Melike OĞUZ

*Bilgisayar Mühendisliği Kocaeli Üniversitesi* {170201004 & 170201028}@kocaeli.edu.tr

***ÖZET***

***“Veri tabanı projesi" adlı program, bir üretici firmanın en düşük fiyatlı hammadde tedarikçisinden kimyasal bileşen alıp, kimyasal bileşenlerin bir araya getirilmesinden oluşan kimyasal maddelerin en yüksek kâr ile müşterilere satmasını amaçlar. Aynı zamanda yapılan tüm bu işlemleri bir veri tabanında tutmaktadır.***

***GİRİŞ***

Veri tabanı projesinde kullanıcı, üretici firma olarak düşünülerek tasarlanmıştır. Belirli bir takım tedarikçi firmalar bulunmaktadır. Tedarikçi piyasasına yeni tedarikçiler eklemek mümkündür.

Üretici firma üretmek istediği kimyasal maddelerin bileşenlerini tedarikçi firmalardan tedarik edip bu kimyasal bileşenleri işleyip, kimyasal maddeleri satmakta ya da stoğunda biriktirmektedir.

Bu programda amaç, üretici firmanın ihtiyacı olan bileşeni, en az maliyetli firmadan almak ve ürettiği kimyasal maddeyi seçilen kâr oranı ile satmaktır.

Tüm bu alım satım işlemleri, tedarikçi firma bilgileri, müşteri bilgileri, üretici firmanın stok bilgileri, üretilenler ve birçok özellik veri tabanında tutulmaktadır. Bu bilgilere, tamamen kullanıcı dostu olan bir arayüz ile erişmeniz mümkündür.

***2.Temel Bilgiler***

Program, Java programlama dilinde ve **"NetBeans IDE"** de yazılmıştır. Veri tabanı yönetimi için açık kaynak kodlu, internet üzerinden **"MySQL"** veritabanı yönetimi olan **"phpMyAdmin"** kullanılmıştır.

Programın yazılım sürecinde git versiyon kontrol sistemi olarak GitHub kullanılmış olup, kodlarımızın gizliliği için GitHub Education hesabımızın bize sağlamış olduğu private repository özelliğinden faydalanılmıştır.

***3. Tasarım***

Veri tabanı projesi aşağıdaki başlıklar altında geliştirilmiştir.

***3.1 Hiyerarşik Yapı :***

Arayüzde bulunan tüm sekmeler, yapılacak işlemler için gerekli olan tanımlamalar ve tanımlamaların setter getter fonksiyonlarının bulunduğu java classı, bu class için gerekli olan methodların yazılmış olduğu ana class ve arayüzdeki görünümünü tasarlamak için **"JDialog Form"** oluşturulmuştur.

***3.2 Sınıfların Özellikleri :***

**Database (class) :** Veri tabanına erişmek için gerekli olan kullanıcı adı, parola, database ismi, host ve port bilgilerinin bulunduğu sınıftır.

**GirisEkrani (JDialog Form) :** Kullanıcının sisteme erişabilmesi için yetki bilgilerinin alındığı ve alınan bilgilerin kontrol edilip onaylandığı arayüzdür.

**MusteriBilgileri (class) :** Müşterilere ait bilgileri tutan MID, musteriAdi, mahalle, sehir, cadde adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**MusteriBilgileriEkrani (JDialog Form) :** Müşteri bilgilerinin yer aldığı, bu verilerin tablo halinde görüntülendiği ve yeni müşteri ekleme, müşteri bilgilerini güncelleme gibi işlemlerin yer aldığı arayüzdür.

**MusteriTalepleri (class) :** Müşterinin talep ettiği ürün hakkında bilgilerin tutulduğu, talepSirasi, MID, talepEdilenUrun, talepSayisi ve urunID adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**MusteriTalepleriEkrani (JDialog Form) :** Müşteri talep bilgilerinin yer aldığı, bu verilerin tablo halinde görüntülendiği ve yeni talep edilen ürüne ait verilerin eklendiği, kâr oranının seçildiği arayüzdür.

**KimyasalMaddeBilgisi (class) :** Kimyasal maddelere ait bilgileri tutan urunID, kimyasalBilesen, gerekenMiktar adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**KimyasalMaddeBilgisiEkrani (JDialog Form) :** Kimyasal maddelere ait bilgilerin yer aldığı, bu verilerin tablo halinde görüntülendiği ve yeni kimyasal madde bilgilerinin eklendiği arayüzdür.

**MusteriIslemleri (class) :** Müşterilere ait tüm bilgilerin veri tabanından çekilmesi için yazılan, ekleme ve güncelleme işlemlerinin yapıldığı, bu işlemlerin tekrar veri tabanına kaydedildiği methodların yer aldığı sınıftır.

**TedarikciBilgileri (class) :** Tedarikçi firmalara ait birtakım bilgileri tutan FID, firmaAd, ulke ve sehir adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**TedarikciBilgileriEkrani (JDialog Form) :** Tedarikçi firmalara ait bilgilerin yer aldığı, bu verilerin tablo halinde görüntülendiği ve yeni tedarikçi ekleme, tedarikçi bilgilerini güncelleme gibi işlemlerin yer aldığı arayüzdür.

**TedarikEdilenUrunBilgileri (class) :** Tedarikçilerin ürettiği hammadde bilgilerini tutan, FID, uretilenHammadde, hammaddeMiktari, uretimTarihi, rafOmru ve satisFiyati adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**TedarikEdilenUrunBilgileriEkrani (JDialog Form) :** Tedarikçi hammadde bilgilerinin yer aldığı, bu verilerin tablo halinde görüntülendiği ve yeni hammaddelerin eklendiği, var olan hammadde bilgilerinin güncellendiği arayüzdür.

**TedarikciIslemleri (class) :** Tedarikçi firmalara ait tüm verilerin veri tabanındançekilmesi için yazılan, ekleme ve güncelleme işlemlerinin yapıldığı, bu işlemlerin tekrar veri tabanına kaydedildiği methodların yer aldığı sınıftır.

**UreticiBilgileri (class) :** Üretici firmaya ait bilgileri tutan FID, firmaAdi, sehir adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**UreticiBilgileriEkrani (JDialog Form) :** Uretici firmaya ait bilgilerin yer aldığı, bu verilerin tablo halinde görüntülendiği, yeni üretici firmaların eklendiği ya da var olan üretici firmaya ait bilgilerin güncellendiği arayüzdür.

**UreticiAlim (class) :** Üretici firmanın tedarikçi firmalardan aldığı hammadde bilgilerini tutan satinAlmaSirasi, firmaAdi, satinAlinan, alisMaliyeti, alinanMaddeMiktari, stokSayisi adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**UreticiAlimEkrani (JDialog Form) :** Uretici firmanın satın aldığı hammaddelere ait bilgilerin görüntülendiği arayüzdür.

**UreticiSatis (class) :** Üretici firmanın ürettiği kimyasal maddelerin satış bilgilerini tutan urunID, MID, iscilikMaliyeti, toplamMaliyet, satisFiyati adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**UreticiSatisEkrani (JDialog Form) :** Uretici firmanın satış yaptığı kimyasal madde bilgilerinin görüntülendiği, her satışta elde edilen kâr oran bilgisinin verildiği arayüzdür.

**UreticiUretilen (class) :** Üretici firmanın ürettiği kimyasal maddelere ait bilgilerin tutan urunID, kimyasalAdi, uretimTarihi, rafOmru ve stok adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**UreticiUretilenEkrani (JDialog Form) :** Üretici firmanın ürettiği kimyasal madde bilgilerinin tablo halinde görüntülendiği, üretici firmanın üretmek istediği kimyasal maddelerin eklendiği arayüzdür.

**hammaddeSatinAlma (class) :** Üretici firmanın stoğunda bulunması için tedarikçi firmalardan aldığı hammadde bilgilerini tutan hammaddeAdi, istenilenMiktar adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**hammaddeSatinAlmaEkrani (JDialog Form) :** Üretici firmanın tedarik etmek istediği kimyasal bileşeni ve miktarını sipariş ettiği arayüzdür.

**SehirlerArasiMesafe (class) :** Sehirlere ait bilgileri tutan sehirSirasi, sehirler, mesafe adlı değişkenlerin tanımlandığı, bu değişkenlerin setter getter fonksiyonlarının ve constructorın bulunduğu sınıftır.

**SehirlerArasiMesafeEkrani (JDialog Form) :** Şehirlere ait bilgilerin tablo halinde görüntülendiği, yeni şehirlerin eklendiği, mevcut şehir bilgilerinin güncellendiği arayüzdür.

**UreticiIslemleri (class) :** Uretici firmaya ait tüm bilgilerin veri tabanından çekilmesi için yazılan, ekleme ve güncelleme işlemlerinin yapıldığı, bu işlemlerin tekrar veri tabanına kaydedildiği methodların yer aldığı sınıftır.

**YönlendirmeEkrani (JDialog Form) :** Kullanıcının istediği verilere ulaşıp, bir takım işlemler yapabilmesi için tasarlanmış olan programın giriş ekran arayüzüdür.

***4.Karşılaşılan Sorunlar ve Çözümleri***

**4.1 Xampp Platformu Üzerinde MySQL Serverın Başlatılamaması :**

Xampp platformunda MySQL için starta bastığımda devamlı başlatmayı durdurup hata veriyordu. Ben de MySQL için **config butonu** üzerindeki **my.ini** dosyasını açıp **##UTF 8 SETTINGS** satırından önceki satıra **innorb\_flush\_method = normal** satırını ekledim.

**4.2 Xammp Kontrol Panelinden MySQL'in Aktifleştirilememesi :**

Makine üzerinde MySQL yüklü ise genellikle makine açılırken aktifleşir. Bu platform üzerinden tekrar aktifleştirme yapmak istediğimizde doğal olarak hata verir. Bu durumda **services.msc** programı üzerinde hali hazırda çalışan MySQL programını **durdurup** işlemi tekrar yaptığımızda sorunun ortadan kalktığını görürüz.

***5-Normalizasyon***

Veri tabanı projesinde normalizasyon işlemleri aşağıdaki başlıklar altında uygunlanmıştır.

***5.1- Normalizasyon nedir ?***

Veri tabanının mantıksal tasarımının tamamlanması ve veri tabanında bulunması planlanan tabloların oluşmasından sonra gerçekleştirilen bir dizi yöntemlerdir.

Normalizasyonun amacı, veri tabanı tasarımlarındaki olası hataları en aza indirerek, veri tabanında oluşabilecek anormal durumları minimize etmektir.

***5.2- First Normal Form ( 1 NF )***

First normal form, tüm niteliklerin çekirdek nitelikte olmasını şart koşar. Bir niteliğin tekrarlı ve birden fazla değer almasının engellenmesi, niteliğin o alanda bulunan en küçük veri parçaları ile tanımlanmasına **bir niteliğin çekirdek nitelikte olması** denilmektedir.

***5.3- Second Normal Form ( 2 NF )***

Bir tablonun 2 NF kuralına uyabilmesi için, öncelikle o tablonun 1 NF kuralına uyması gerekmektedir. Bu kuralı sağladıktan sonra bütün asal olmayan niteliklerin anahtar olarak belirlenen nitelik ya da niteliklere **tam bağımlı olması** gerekir.

Bir tablonun 2 NF'e getirilmesi için **tabloda parçalama yapmak** gerekir. Parçalama işlemi genellikle **problem yaratan işlevsel bağımlılıklar** gözönüne alınarak gerçekleştirilir.

***5.4- Third Normal Form ( 3 NF )***

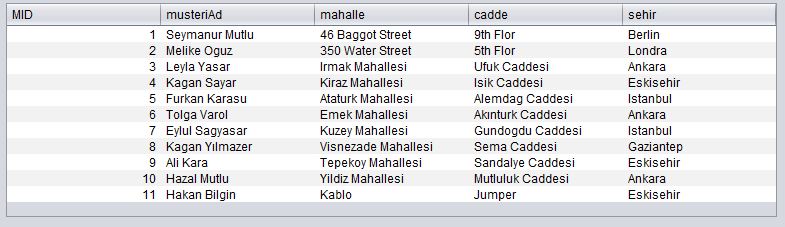
Bir tablonun 3 NF kuralına uyması için öncelikle 2 NF kuralına uyması gerekmektedir. Bu koşul sağlandıktan sonra, bütün asal olmayan niteliklerin anahtar olarak belirlenen nitelik ya da niteliklere **geçişken olarak değil, doğrudan bağlı olması** gerekir.

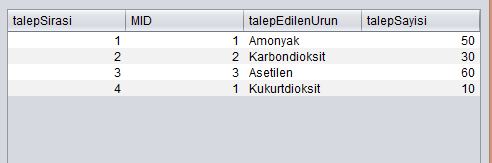
***5.5- Normalizasyon Kurallarının Tablolara Uygulanması***

******

***Müşteri Bilgileri Tablosu (Şekil 1)***

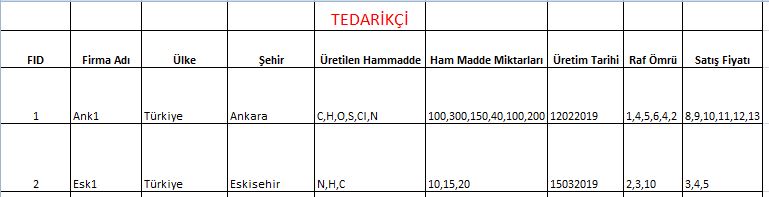
**Şekil 1 'deki** tablo **1** **NF kuralına uymamaktadır.** Çünkü **Adres** adındaki sütunda yer alan bilgiler **çekirdek nitelikte değildir.** Adres özelliği birden fazla özellik ile nitelendirilmiştir. Bu yüzden tablonun çekirdek nitelikte olması sağlanmalıdır.



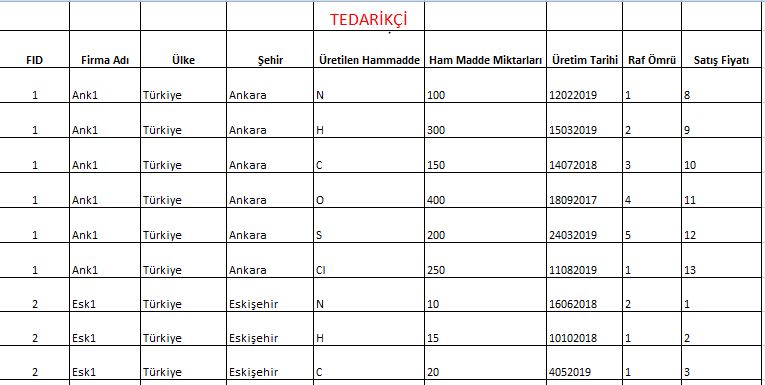
***Müşteri Bilgileri Tablosu (Şekil 2)***

***Müşteri Bilgileri Tablosu (Şekil 3)***

**Şekil 2'de** 1 NF kuralına uymayan **Adres** bilgileri **mahalle, cadde ve sehir** olarak ayrılmıştır. Aynı zamanda **talepEdilenUrun** özelliği **kısmi bağımlılık** içerdiği için **2 NF kuralına uymamaktadır.** Bu sorunu düzeltmek için **Şekil 3'te** gösterilen, **musteriTalepleri** adlı bir tablo oluşturulmuştur.

***Tedarikçi Tablosu (Şekil 4)***

**Şekil 4 'deki** tablo **1** **NF kuralına uymamaktadır.** Çünkü **Uretilen Hammadde, Hammadde Miktarları, Uretim Tarihi, Raf Omru** ve **Satış Fiyatı** adındaki sütunlarda yer alan bilgiler **çekirdek nitelikte değildir.** Bu sütunlar birden fazla özellik ile nitelendirilmiştir. Bu yüzden tablonun çekirdek nitelikte olması sağlanmalıdır.

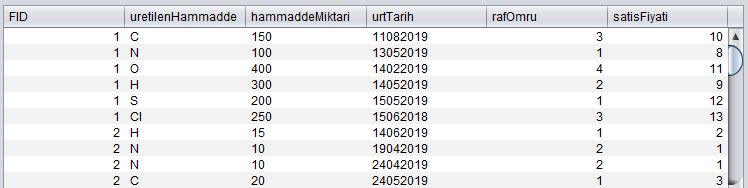


***Tedarikçi Tablosu (Şekil 5)***

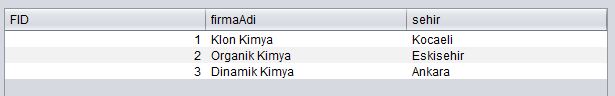
**Şekil 5'teki** tedarikçi tablosu 1 NF kuralına uymaktadır. Ancak tablo kısmi bağımlılıklar içerdiği için **2 NF kuralına uymamaktadır.** Tablo, **TedarikciBilgileri (Şekil 6)** ve **TedarikEdilenUrunBilgileri** **(Şekil 7)** olmak üzere ikiye ayrılmıştır.

******

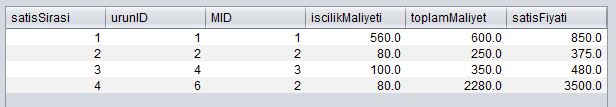
***TedarikciBilgileri (Şekil 6)***

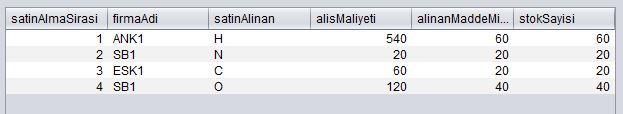
***TedarikEdilenUrunBilgileri (Şekil 7)***

***Uretici Tablosu (Şekil 8)***

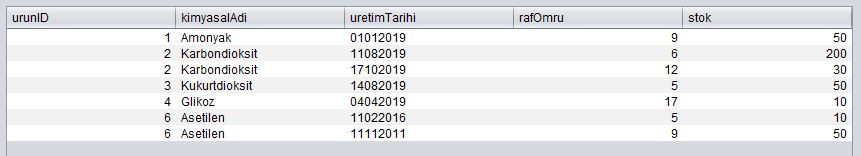
**Şekil 8 'deki** tablo **1** **NF kuralına uymamaktadır.** Çünkü **Satın Alınan Hammaddeler, Alış Maliyeti, Stok Bilgileri, Geliştirilen Kimyasal Urun Adı ve Kimyasal Ürün Bileşenleri** adındaki sütunlarda yer alan bilgiler **çekirdek nitelikte değildir.**  Bu sütunlar birden fazla özellik ile nitelendirilmiştir. Bu yüzden tablonun çekirdek nitelikte olması sağlanmalıdır. Tablo aynı zamanda kısmi bağımlılıklar içerdiği için **UreticiBilgileri (Şekil 9), UreticiSatis (Şekil 10), UreticiAlim (Şekil 11),** ve **UreticiUretilen (Şekil 12)** adlı 4 tabloya ayrılmıştır.

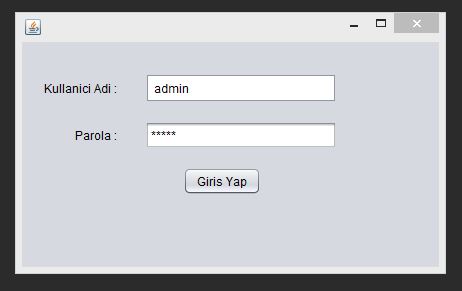
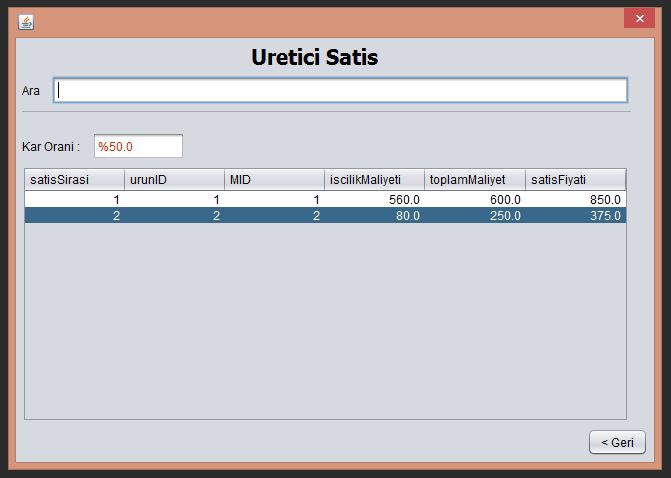
***UreticiBilgileri (Şekil 9)***

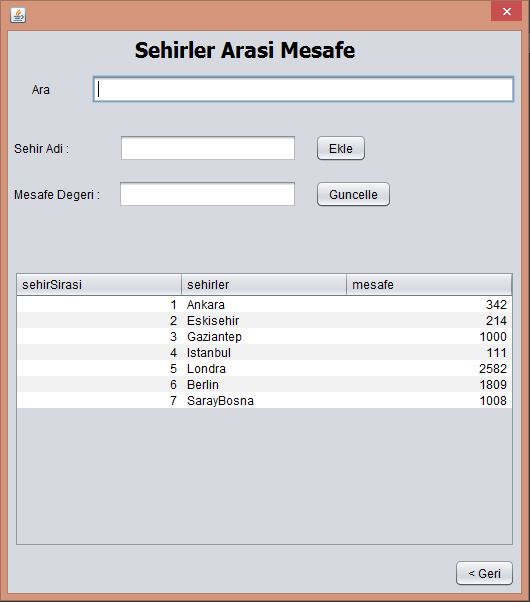
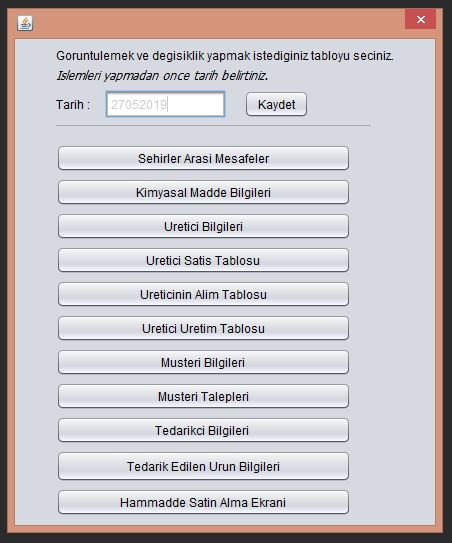
***UreticiSatis (Şekil 10)***

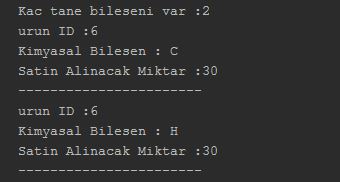
******

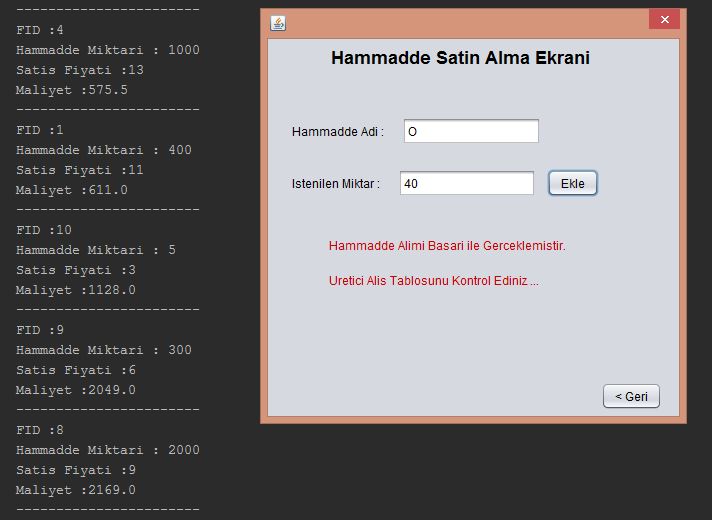
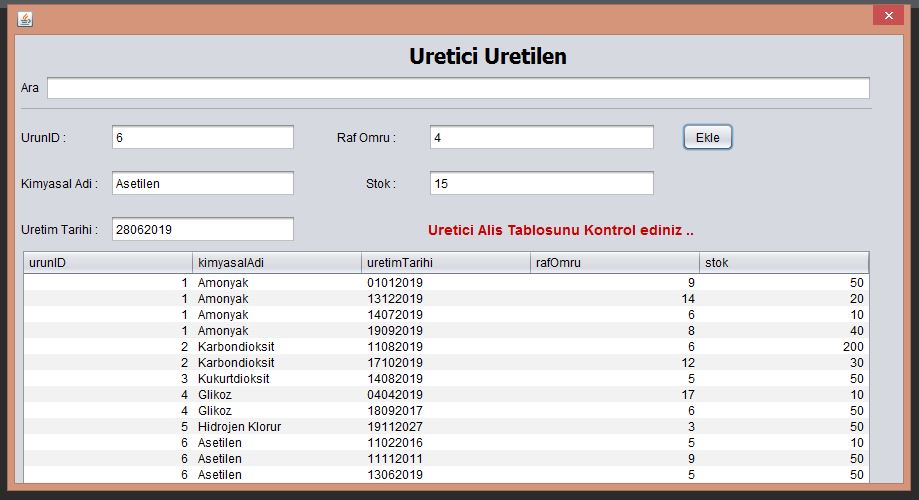
**UreticiAlim (Şekil 11)**

***UreticiUretilen (Şekil 12)***

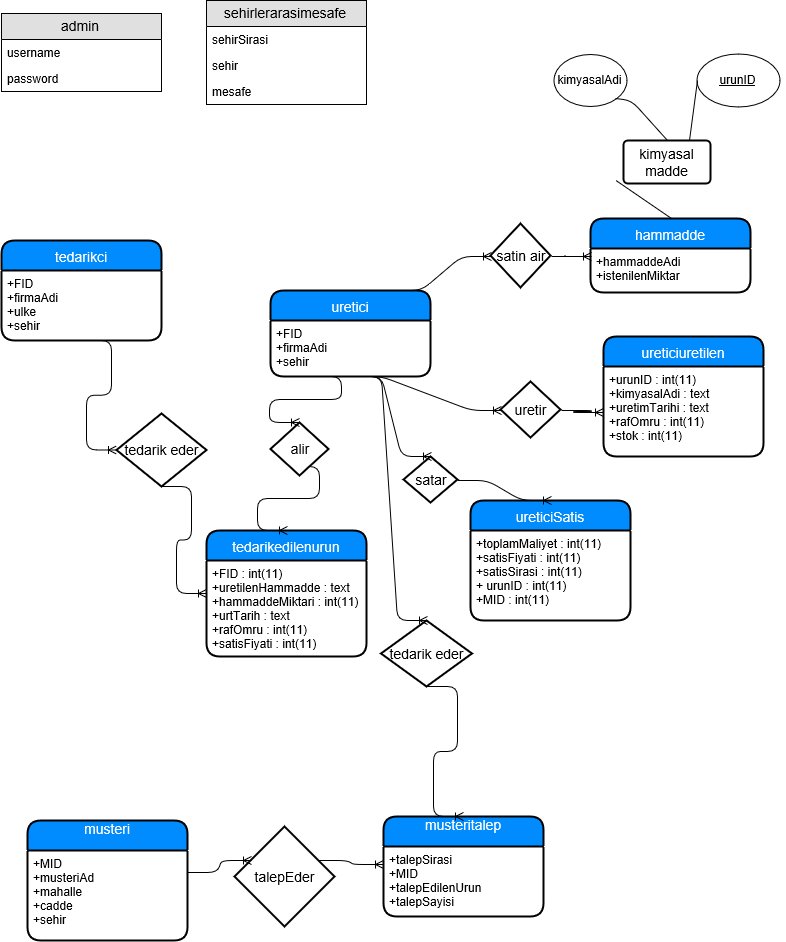
**6-Ekran Görüntüleri**

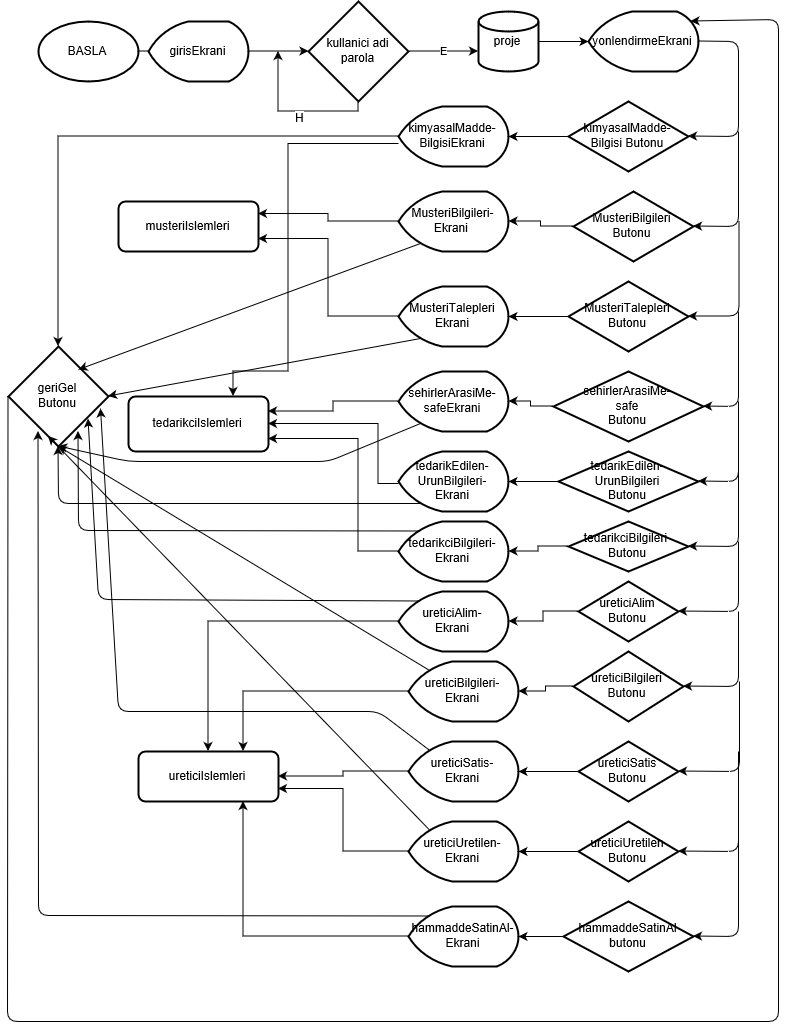
****

******

****

**7- ER Diyagramı**

****

**8- Akış Diyagramı**

***9.Kazanımlar***

1- MySql veri tabanı yönetim sistemini öğrenmek,

2- MySqlin sorgu yapısını anlamak ve pekiştirmek,

3-Normalizasyonu ve normalizasyon kurallarını öğrenmek,

4- Bir projede veri tabanının nasıl efektif kullanacağını öğrenmek,

5-Dinamik bir sistem oluşturmak,

6- Git sürümleri - GitHub Kullanımı

***10.Kaynakça***

[1]<https://kodcu.com/2012/12/jdbc-ile-mysql-baglantisi/>

[2]<https://dev.mysql.com/doc/refman/8.0/en/create-procedure.html>

[3]<https://www.youtube.com/watch?v=4h8VoRnRvnE>

[4]<https://www.youtube.com/watch?v=wTJjpH2RUcQ>

[5]<https://www.youtube.com/watch?v=379qkZTibZA>