

Programlama Laboratuvarı 2

Umut SÜTCÜ 200202038

Ömer ARAN 190202012

ÖZET

Bu doküman programlama laboratuvarı 2 dersinin 3.projesi olan banka yönetim sistemi ve veritabanının üzerinde gerekli işlemleri gerçekleyen bir uygulama yapmamız istenmiştir. Bu dökümanda proje tanıtımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, geliştirme ortamı ve kullanılan kaynaklara yer verilmiştir.

PROJENİN TANIMI

Tasarlanan veritabanı, bankanın müşterilere, çalışanlara, hesaplara ve işlemlere ilişkin bilgileri organize bir şekilde işlemesine yardımcı olacaktır. Bu şekilde bankanın ve müşterilerin ihtiyacı olan bilgilere daha kolay ulaşabilmesi sağlanmış olacaktır. Ayrıca, veritabanı kullanarak müşteri istekleri ve bankanın ihtiyaçları doğrultusunda raporlar hazırlanabilecektir. Bir veritabanı tasarımının ilk aşamasında sistemin ihtiyaçlarının belirlenmesi ve depolanacak bilgi türlerinin tanımlanması için Varlık-İlişki (ER) diyagramı oluşturulmalıdır. ER diyagramı sistem içerisinde var olabilecek varlıkların ve aralarındaki ilişkilerin görsel olarak ifade edilmesi için kullanılır. Geliştirme sırasında, ER diyagramı gereksinimlerin daha açık ve özlü bir şekilde haritalanmasına yardımcı olmaktadır. Sizlerden problemlerin çözümü için bir veritabanı tasarımı yapmanız istenmektedir. Tablo sayısı tüm tablolar en az 3NF normalizasyon formuna uyacak şekilde size bırakılmıştır. Projede oluşturulacak veritabanının ER diyagramının oluşturulması ve arayüzde gösterilmesi gerekmektedir. Diyagram üzerinden gerçekleştirilen normalizasyon işlemlerinin adım adım gösterilmelidir

Banka içerisinde müşteri, temsilci ve banka müdürü olmak üzere 3 adet rol bulunmaktadır. Müşteriler ve çalışanlar için gerekli tanımlayıcı bilgiler (Ad Soyad, Telefon, TC No, Adres, E-posta) veri tabanında saklanmalıdır. Bir müşterinin birden fazla hesabı bulunabilir. Hesaplar sistem içerisinde kayıtlı bulunan herhangi bir para birimi cinsinden açılabilir (TL varsayılan olarak gelmelidir). Hesaplar arası para transferinde gerekli durumlarda kur dönüşümü otomatik olarak yapılmalıdır.

Müşteriler; o Hesaplarından para çekebilirler ve yatırabilirler. o Yeni hesap açma ve var olan bir hesabı silme talebinde bulunabilirler. Bakiyesi "0" olmayan bir hesap silinemez. o Birbirleri arasında para transferi yapabilirler. Farklı para birimlerine sahip hesaplar arası transferler sırasında gönderilen miktar hedef para birimine

otomatik olarak çevrilmelidir. o Bilgilerini güncelleyebilirler. (Adres, Telefon vs.) o Bankaya para transferi yapabilirler. (Kredi borcu ödeme) o Bankadan kredi talep edebilirler. Kredi sadece TL cinsinden talep edilebilmektedir. Bankanın kredi talebini onaylaması durumunda istenilen vade oranınca (faiz ve anapara toplamı) bölünerek aylara borç olarak yansıtılır. Aylık özet görüntülemeye kredi borcu ödemeleri için ödenen faiz ve anapara ayrı ayrı görüntülenmelidir. Müşterinin aylık borcunun tamamını ödememesi durumunda kalan borç ek faiz hesaplanarak bir sonraki aya devreder. Faiz ve gecikme faiz oranı banka müdürü tarafından belirlenir. Aylık borç ve kalan borç ayrı ayrı görüntülenmeli. (müşteri isterse tüm borcunu tek seferde ödeyebilir) Erken ödeme durumlarında gelecek aylar için faiz alınmayacaktır. o Aylık özetlerini görüntüleyebilirler. (Geçerli ay içerisinde yaptığı para gönderme, çekme, kredi borcu ödeme gibi işlemlerin özeti)

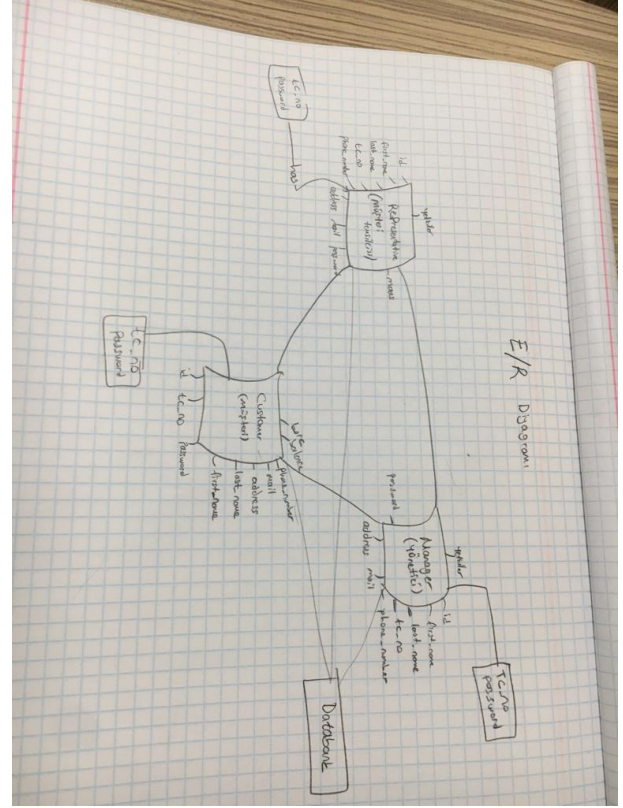
Banka müdürü; o Bankanın genel durumunu (gelir, gider, kar ve toplam bakiye) görüntüleyebilmektedir. o Yeni para birimi (Dolar, Euro, Sterling vs.) ekleyebilir ve kur değerlerini güncelleyebilir. o Çalışanların maaş ücretlerini belirleyebilecektir. Tek bir çalışan türü vardır (müşteri temsilcisi). Hepsinin maaş miktarı aynıdır. o Kredi ve gecikme faiz oranını belirler. o Müşteri ekleyebilir. Sisteme yeni bir müşteri eklenmesi durumunda en az müşteriye sahip olan temsilciye atanır. o Sistemi bir ay ilerletebilir. İsterlerin test edilebilmesi için sizlerden uygulama tarihini bir ay ileriye öteleyebilmeniz istenmektedir. Bu ilerletme işlemi sonucunda -ç maaşların ödenmesi, gelir-gider durumlarının güncellenmesi ve müşterilerin bir sonraki aya ait borçlarının kendilerine yansıtılması gerekmektedir. o Bankada gerçekleşen tüm işlemleri (para çekme, yatırma ve transfer) görüntüleyebilmektedir. İşlemleri listelerken "son X adet işlemi listele" şeklinde bir seçenek sunulmalıdır. Örnek olarak "son 5 işlemi listele" sorgusunun çıktısı Tablo 1'de gösterilmiştir. o Listelenen işlemlerin aynı anda başlatılması durumunda deadlock oluşup, oluşmadığının analizinin yapılabilmektedir. Deadlock analizi ayrı bölümde açıklanacaktır.

Müşteri temsilcisi; o Her müşterinin bir temsilcisi vardır. o Müşteri ekleme, silme ve düzenleme yapabilir (silme ve düzenleme işlemleri sadece kendi müşterileri için geçerlidir). o Müşteri bilgilerini güncelleyebilirler. (Adres, Telefon vs.) o İlgilendikleri müşterilerin genel durumlarını (gelir, gider ve toplam bakiye) görüntüleyebilmektedir. o Müşterilerden

gelen hesap açma, silme ve kredi taleplerini görüntüleme ve onaylama sorumluluğu temsilcilere aittir. o İlgilendikleri müşterilerin işlemlerini (para çekme, yatırma ve transfer) görüntüleyebilmektedir.

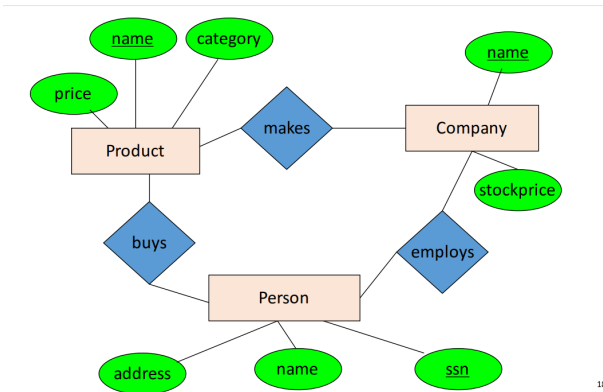
ARAŞTIRMA VE YÖNTEMLER

İlk önceliğimiz ER diyagramı ve normalizyon üzerinde çalışmak oldu bunun için birçok farklı yerden ve hocalarımızın kaynaklarından çalıştık. Dil olarak daha çok tecrübemiz olduğu için java seçtik ve arayüz tasarlamak için de java swing kullandık. Java swing için youtube videoları ve deneme yanılmalarımız yardımcı oldu. IDE olarak da beraber yaptığımız için birden çok IDE'yi aynı anda kullanma şansımız oldu. Bu IDE'ler visual studio code , IntelliJ IDEA ve netbeans'dir.

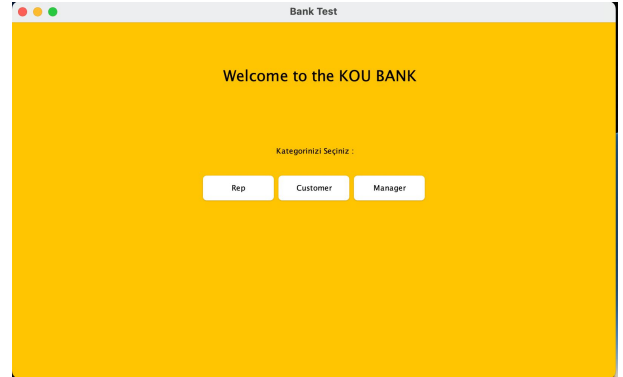


Projemizin E/R diyagramı

En çok sıkıntı çektiğimiz database ile javayı bağlamakta arayüzde yapılan bir şeyi javada işleyip database'e göndermek bizim için uğraştırıcı oldu.



Yukarıdaki fotoğraf bir E/R diyagramı örneği



Yukarıdaki fotoğraf swing ile yaptığımız bir arayüz

Swing ile çalışma yaparken intelliJ idea idesinde yaptığımızda bizim için yanda ne yaptığımızı canlı olarak gösteriyordu ve sürükleyip bırak yöntemi ile yerlerini ayarlabiliyorduk bu sayede işimizi çok hızlandırdı.

```
package view;

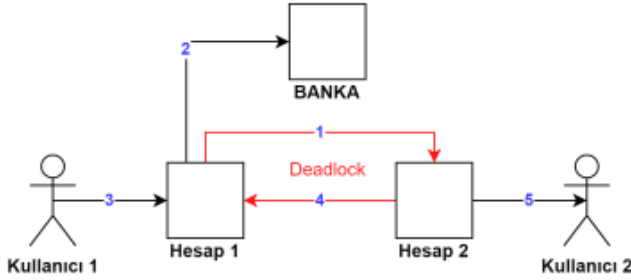
import javax.swing.*;

public class Main {

    public static void main(String[] args) {
        JFrame arayuz = new JFrame("Bank Test");
        JPanel panel = new GuiMain(arayuz);
        arayuz.setContentPane(panel);
        arayuz.setSize(800,500);
        arayuz.setVisible(true);
        arayuz.setDefaultCloseOperation(arayuz.DISPOSE_ON_CLOSE);
    }
}
```

Arayüz için yazdığımız kodun küçük bir kısmı

Para gönderimi sırasında hedefin işlem yapması engellenmektedir. Bu nedenle para almakta olan bir hesap para gönderimi yapamamaktadır. Deadlock analizi için tüm işlemlerin aynı anda çalışmaya başladığı ve paralel şekilde çalıştığı kabul edilebilir.



Deadlock örneği

Şekilde görüldüğü üzere aynı anda başlamış olan 1 ve 4 işlemleri birbirlerini kilitlemektedir. Sistem çıktı olarak deadlock sayısını ve deadlock olan olan işlemler döndüreceklerdir.

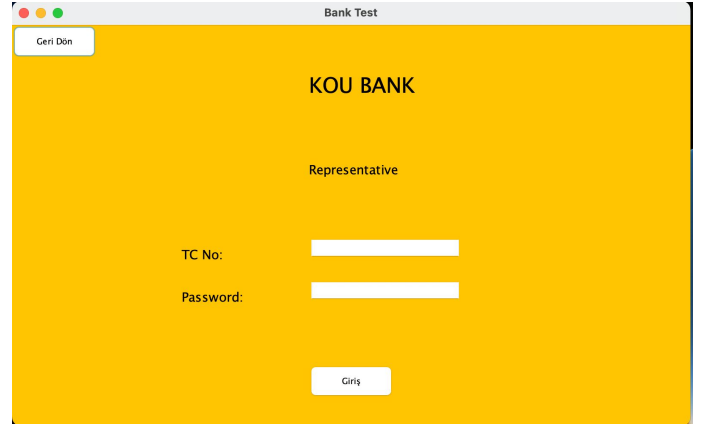
KOD BİLGİSİ(YALANCI KOD)

Kodu txt dosyası olarak ek olarak gönderdik.

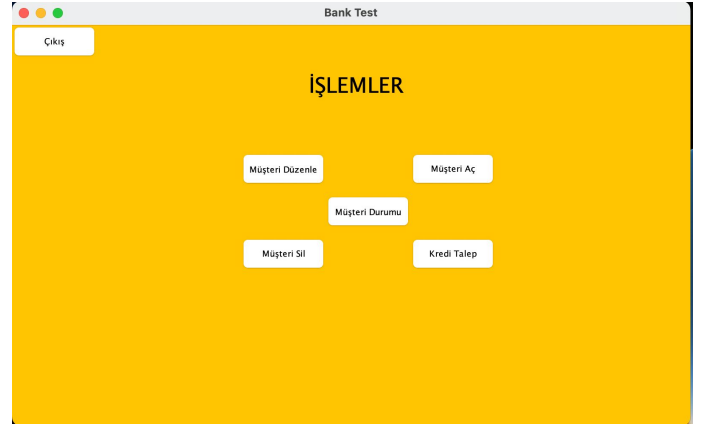
SONUÇ

Sonuç olarak java ve database bağlantıları nasıl yapılır. Yapılan database için ER diyagramı hazırlama ve kullanmanın nasıl olduğunu öğrendik. 3 nf kuralına göre bir database tablosu hazırlamanın nasıl bir şey olduğu gördük. Java swing ile de bir arayüz tasarlamının ne olduğunu öğrendik. En önemlisi de bunları birbirine bağlamayı ve arayüzde yapılan bir işlemin database üzerinde nasıl bir değişiklik yaşattığını gördük. Bunların kullanımı üzerine çalışmış olduk.

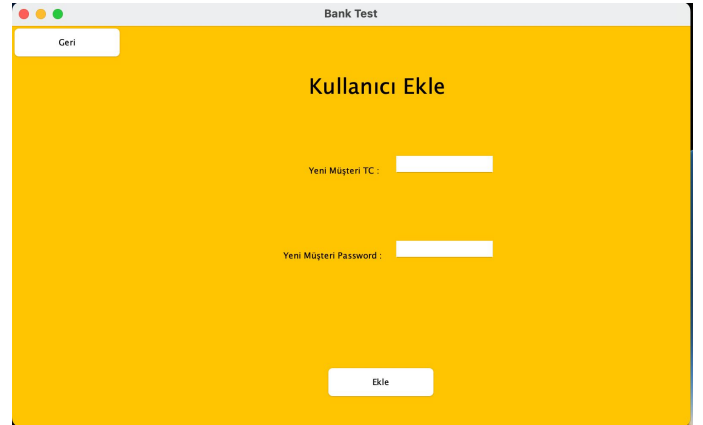
Aşağıda da projemize ait ekran görüntüleri:



Projemizin giriş ekranı



İşlemler menüsünün ekran görüntüsü



Kullanıcı ekleme işleminin ekran görüntüsü

KAYNAKÇA

<https://www.guru99.com/database-normalization.html>

<https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>

<https://www.javatpoint.com/java-swing>

<https://www.w3schools.com/java/>

<https://www.w3schools.com/mySQL/default.asp>

<https://sadienvrenseker.com/wp/>