

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programlama Laboratuvarı II

Proje 3

BANKA YÖNETİM SİSTEMİ UYGULAMASI

Muhammad Abdan SYAKURA (200201147) – Robera Tadesse GOBOSHO (190201141)

prof.syakur@gmail.com – robtad318@gmail.com

Özet

Bu rapor Programlama Laboratuvarı II dersi 3. Projesinin çözümünü açıklamak üzere hazırlanmıştır.

Projenin amacı üzerinden bir bankanın yönetim sistemi için bir veritabanı tasarladık. Bu veritabanının üzerinde gerekli işlemleri gerçekleştirilebilir. Veritabanı tasarımı için rapor son kısmında ERD bulunmaktadır. Veritabanında bulunan tabloları da 3NF normalizasyonuna uyuyacak şekilde tasarlamaya çalıştık.

Bu proje C# diliyle veritabanı için de Microsoft SQL Server üzerinden tasarlanmıştır. Uygulamanın GUI için C# .NET Framework içinde bulunan WinForms kütüphanesini kullandık.

C# ile veritabanı arasında bağlantı için DbConnector diye bir sınıf oluşturduk. App.config içinde bağlantı string tanımlarını. Tekrar tekrar bağlantı kurmaya gerek kalmadan App.config içindeki bağlantı stringi direkt değiştirilebilir. Böylece kodları yazarken daha verimli olur.

Banka yönetim sisteminde müşteri, temsilci ve banka müdürü olmak üzere 3 adet rol bulunmaktadır. Bunların profil, giriş bilgileri gibi gerekli bilgiler veri tabanında saklanmaktadır. Ve her rol sistemde normal bankalardaki gibi kendi yapabileceği işlemler tanımlanmıştır ve role göre 3 adet panel bulunmaktadır. Banka müdürü bankadaki işlenen tüm işlemlerine bakabilir, temsilciler kendi müşterilerin taleplerini onaylayabilir, müşteriler başka hesaplara para gönderebilir gibi işlemleri yapabilecekler. Tüm işlemler veritabanına gönderilip kaydedilmektedir.

Giriş

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 saklanmaktadır. Bir müşterinin birden fazla hesabı bulunabilir. Hesaplar sistem içerisinde kayıtlı bulunan herhangi bir para birimi cinsinden açılabilir (TL varsayılan olarak gelmektedir). Hesaplar

arası para transferinde gerekli durumlarda kur dönüşümü otomatik olarak yapılmaktadır. Rollerin gerçekleştirdiği eylemler aşağıda belirtilmiştir. Tüm bu eylemlerin tasarlanan bir ara yüz üzerinden görsel bir şekilde yapabilmektedir.

1. Müşteriler;

- Hesaplarından para çekebilirler ve yatırabilirler.
- Yeni hesap açma ve var olan bir hesabı silme talebinde bulunabilirler. (Bakiyesi “0” olmayan bir hesap silinemez)
- Birbirleri arasında para transferi yapabilirler. (Farklı para birimlerine sahip hesaplar arası transferler sırasında gönderilen miktar hedef para birimine otomatik olarak çevrilmektedir)
- Bilgilerini güncelleyebilirler.
- Bankaya para transferi yapabilirler. (Kredi borcu ödeme)
- Bankadan kredi talep (kredi talebi için aşağıda ayrıntılarını anlatılacak) edebilirler.
- Aylık özetlerini görüntüleyebilirler. (Geçerli ay içerisinde yaptığı para gönderme, çekme, kredi borcu ödeme gibi işlemlerin özeti)

2. Müşteri Temsilcisi;

- Her müşterinin bir temsilcisi vardır.
- Müşteri ekleme, silme ve düzenleme yapabilir (silme ve düzenleme işlemleri sadece

kendi müşterileri için geçerlidir).

- Müşteri bilgilerini güncelleyebilirler.
- İlgilendikleri müşterilerin genel durumlarını (gelir, gider ve toplam bakiye) görüntüleyebilmektedir.
- Müşterilerden gelen hesap açma, silme ve kredi taleplerini görüntüleme ve onaylama sorumluluğu temsilcilere aittir.
- İlgilendikleri müşterilerin işlemlerini (para çekme, yatırma ve transfer) görüntüleyebilmektedir.

3. Banka Müdürü;

- Bankanın genel durumunu (gelir, gider, kar ve toplam bakiye) görüntüleyebilmektedir.
- Yeni para birimi (Dolar, Euro, Sterling vs.) ekleyebilir ve kur değerlerini güncelleyebilir.
- Çalışanların maaş ücretlerini belirleyebilir. (Tek bir çalışan türü vardır (müşteri temsilcisi). Hepsinin maaş miktarı aynıdır.)
- Kredi ve gecikme faiz oranını belirler.
- Müşteri ekleyebilir. (Sisteme yeni bir müşteri eklenmesi durumunda en az müşteriye sahip olan temsilciye atanır)
- Sistemi bir ay ilerletebilir. (İsterlerin test edilebilmesi için uygulama tarihini bir ay ileriye öteleyebilmektedir. Bu

ilerletme işlemi sonucunda -> maaşların ödenmesi, gelir-gider durumlarının güncellenecek ve müşterilerin bir sonraki aya ait borçlarının kendilerine yansıtılacaktır)

- Bankada gerçekleşen tüm işlemleri (para çekme, yatırma ve transfer) görüntüleyebilmektedir. (İşlemleri listelerken “son X adet işlemi listele” şeklinde bir seçenek sunulmaktadır)
- Listelenen işlemlerin aynı anda başlatılması durumunda deadlock oluşup, oluşmadığının analizinin yapabilmektedir. (Deadlock analizi ayrı bölümde açıklanacaktır.)

Müşterilerin kredi talepleri için talimatlar aşağıdaki gibidir;

- 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.

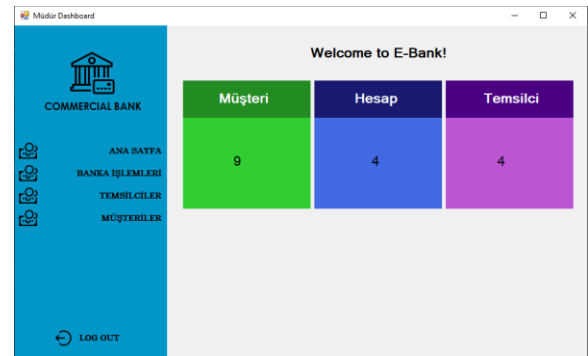
- Aylık borç ve kalan borç ayrı ayrı görüntülenmeli. (Müşteri isterse tüm borcunu tek seferde ödeyebilir)
- Erken ödeme durumlarında gelecek aylar için faiz alınmayacaktır.

Yöntem

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 edilecektir. Bu deadlock analiz için de kendi algoritmamızı geliştirdik. Ve tüm deadlockları gösterecektir.

Deneysel Sonuçlar

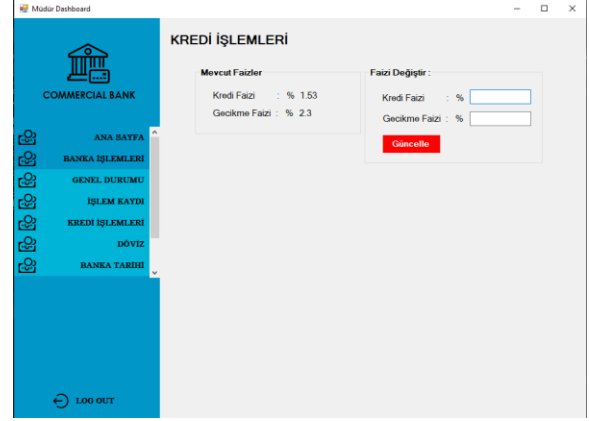
Aşağıdaki uygulamanın bazı arayüz görüntüleri. Önce uygulama başlatıldığında login paneli çıkacaktır. Ondan sonra giriş yapılır. Ve giriş bilgilerine göre panel açılacaktır. Aşağıki müdür panelinden birkaç görüntüler;



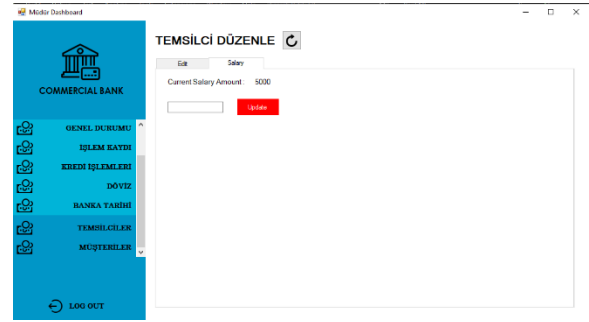
Şekil 1Müdür Panel Girişi



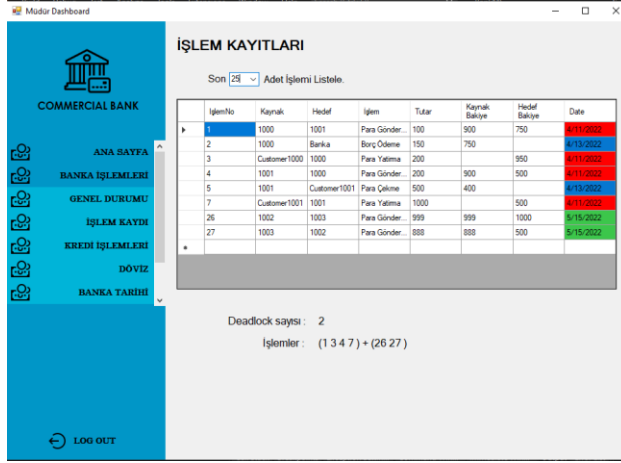
Şekil 2 Bazı Müdür işlemleri



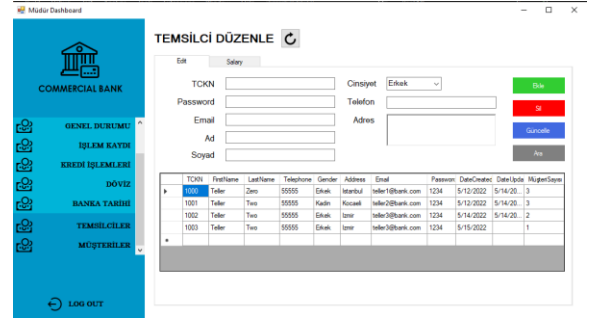
Şekil 5 Kredi Faiz İşlemleri



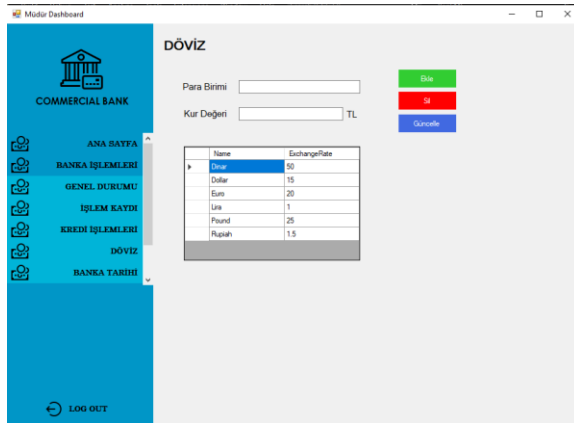
Şekil 6 Maaş İşlemi



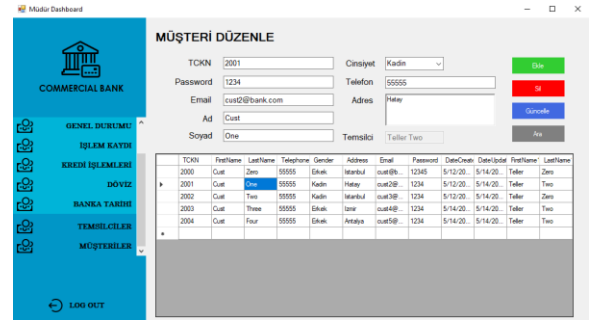
Şekil 3 Deadlock analizi



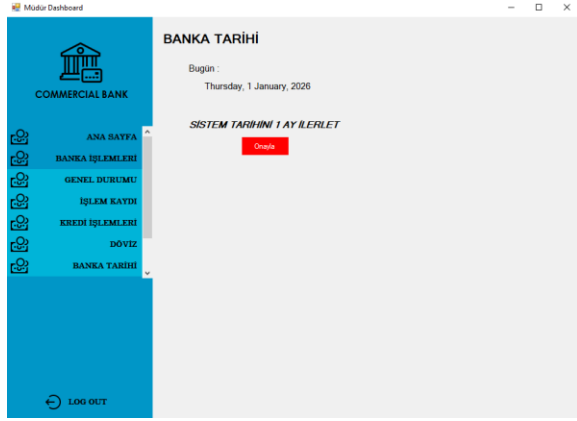
Şekil 7 Temsilci İşlemleri



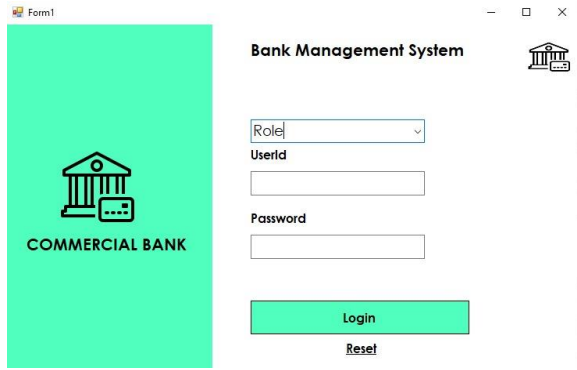
Şekil 4 Döviz İşlemleri



Şekil 8 Müşteri İşlemleri



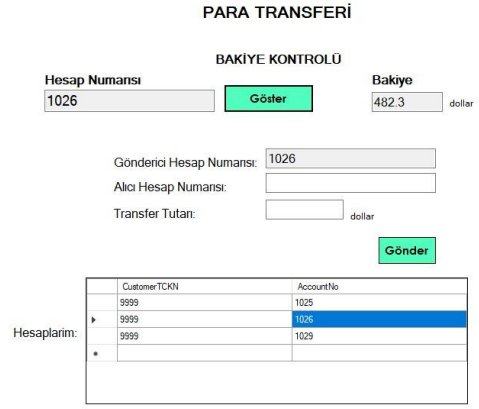
Şekil 9 Banka Tarihi İşlemi



Şekil 10 Login Paneli

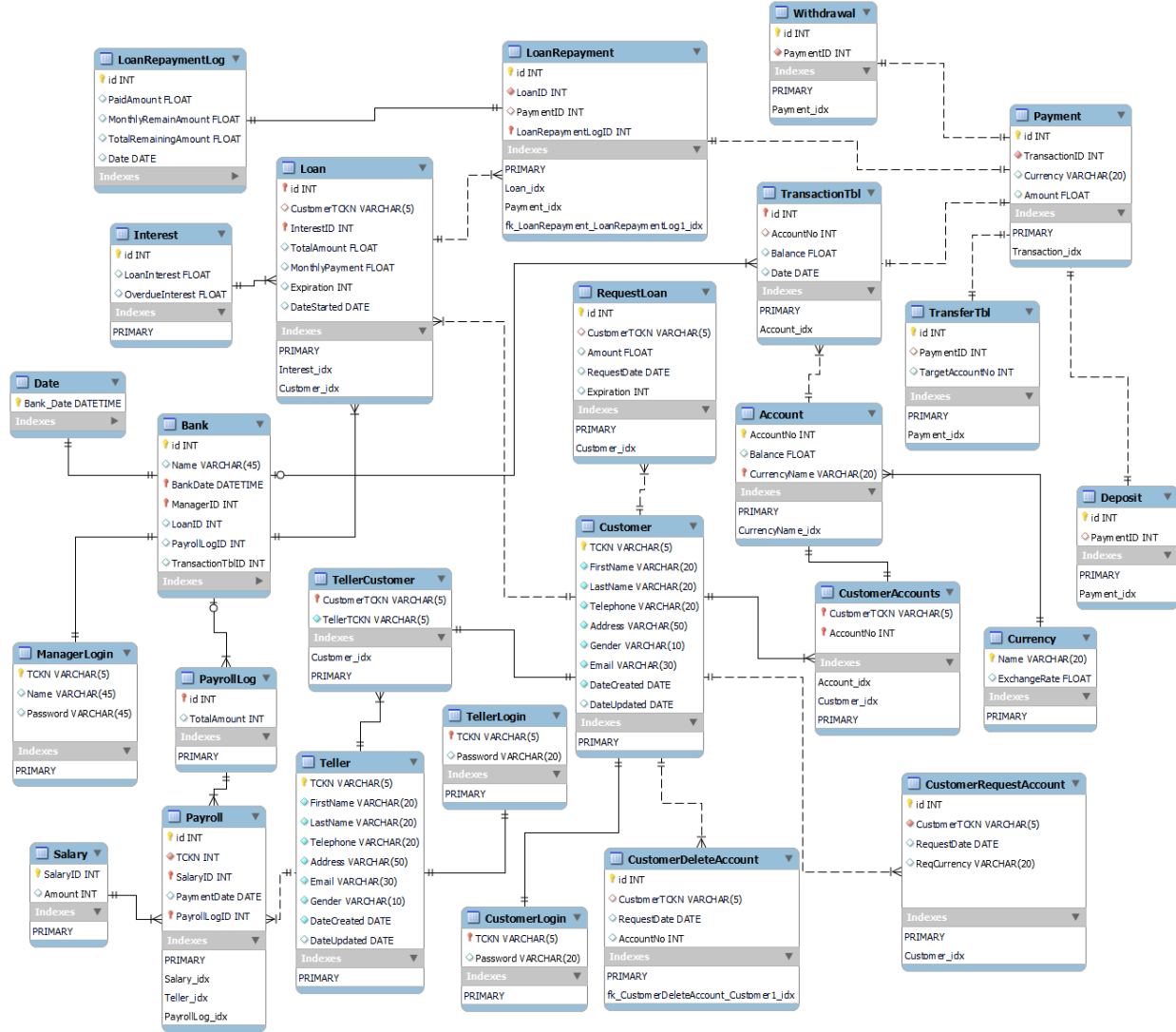


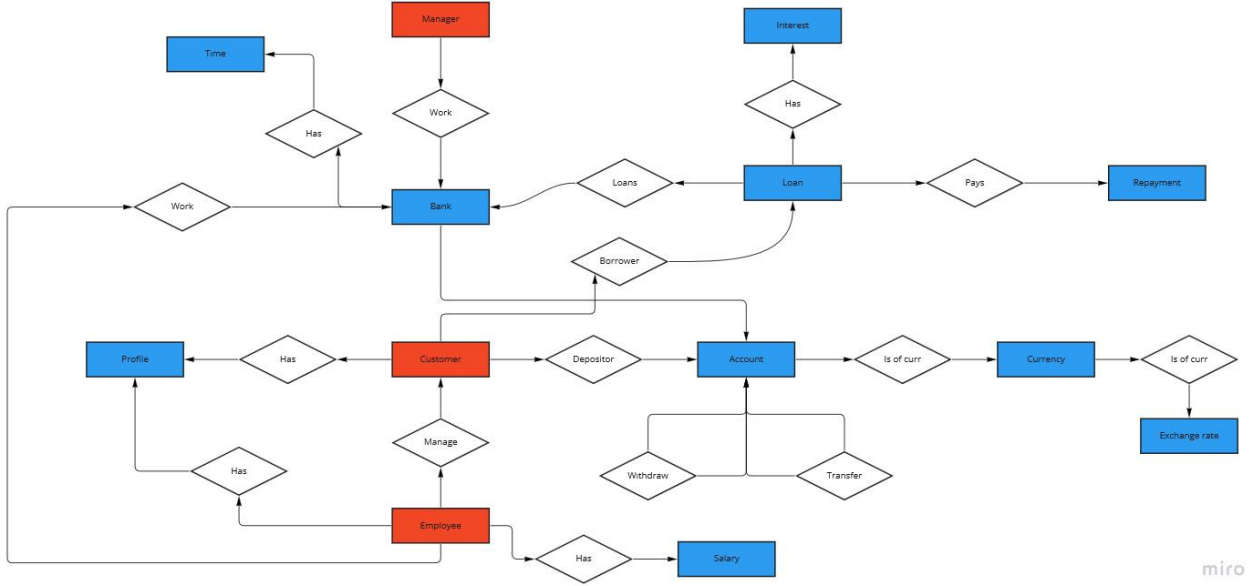
Şekil 11 Temsilci Paneli



Şekil 12 Para Transferi Ekranı

ER Diyagramı





Kaynakça

1. <https://www.guru99.com/database-normalization.html>
2. <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>
3. <https://www.edrawsoft.com/article/er-diagrams-for-banking-system.html>
4. <https://www.tutorialspoint.com/construct-an-er-diagram-for-the-banking-system-in-dbms>
5. <https://stackoverflow.com/questions/6536715/get-connection-string-from-app-config>
6. <https://stackoverflow.com/questions/919244/convert-a-string-to-datetime>
7. <https://www.youtube.com/watch?v=CQB1127dL0>
8. <https://stackoverflow.com/questions/176514/what-is-the-difference-between-char-nchar-varchar-and-nvarchar-in-sql-server>