

Turkcell İin Kredi Modl

Zeynep Alıcı

Geleceęi Yazacak Kadın Mhendisler Java Kampı

Marmara niversitesi Bilgisayar Mhendislięi

Projeye Genel Bakış

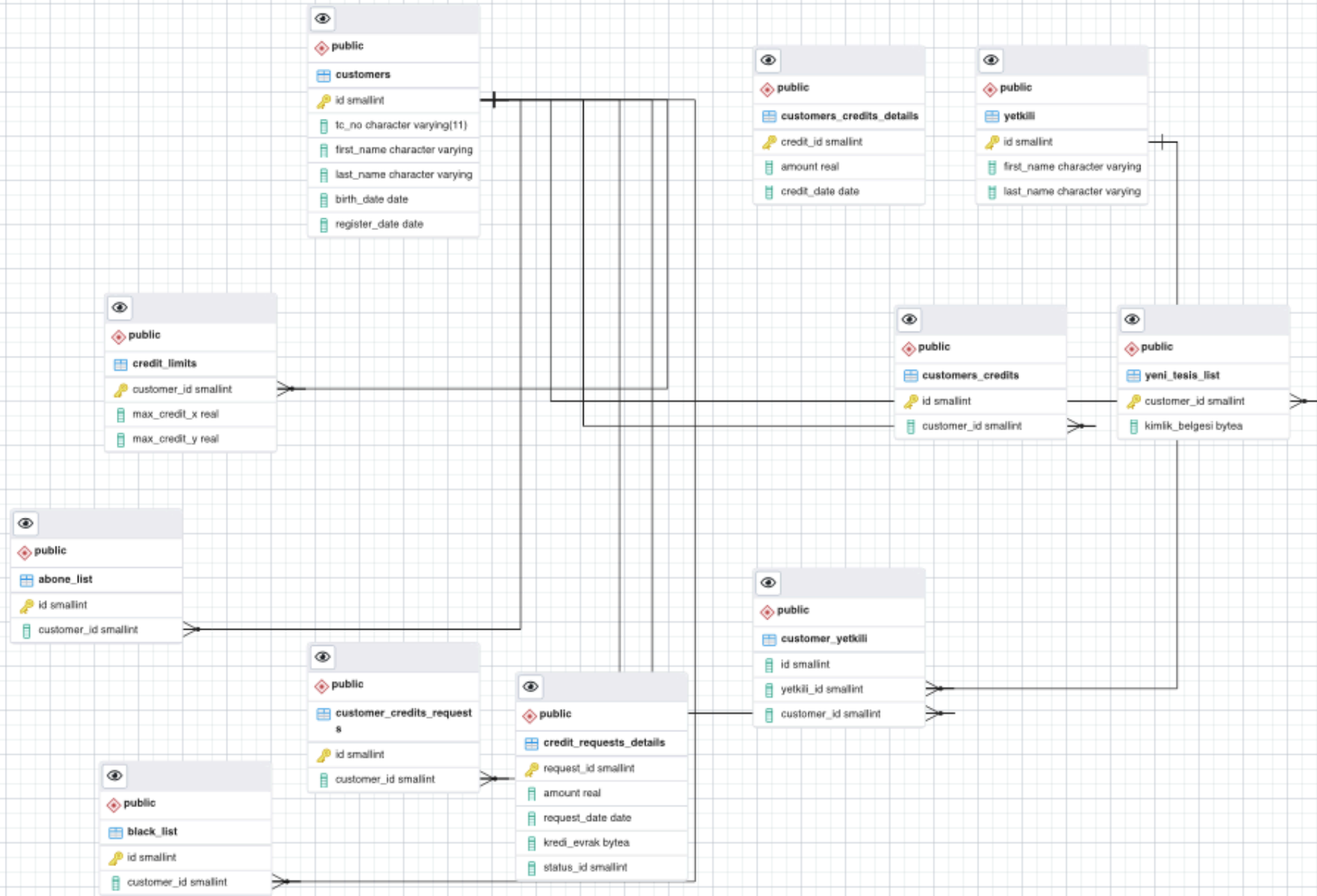
- Turkcell Kredi Modülü, katmanlı mimariye uygun bir şekilde geliştirildi.
- Programlama dili olarak Java 13 ve bununla birlikte Spring Boot kullanılarak bağımsız Spring tabanlı bir uygulama olarak geliştirildi.

- ▼ src
 - ▼ main
 - ▼ java
 - ▼ com.example.turkcellcreditmodule
 - ▼ api.controllers
 - CustomerCreditRequestController
 - YetkiliController
 - ▼ business
 - ▼ abstracts
 - IBlackListService
 - ICreditRequestDetailsService
 - ICreditRequestService
 - ICustomerCreditDetailsService
 - ICustomerCreditLimitService
 - ICustomerCreditService
 - ICustomerService
 - ICustomerYetkiliService
 - IGeneralCreditRequestService
 - ITurkcellAbonesiService
 - IYeniTesisCustomerCreditLimitService
 - IYeniTesisService
 - IYetkiliService
 - ▼ concretes
 - BlackListManager
 - CreditRequestDetailsManager
 - CreditRequestManager
 - CreditRequestManagerForTurkcellAbonesi
 - CreditRequestManagerForYeniTesis
 - CustomerCreditDetailsManager
 - CustomerCreditLimitManager
 - CustomerCreditManager
 - CustomerManager
 - CustomerYetkiliManager

- GeneralCreditRequestManager
- TurkcellAbonesiManager
- YeniTesisCustomerCreditLimitManager
- YeniTesisManager
- YetkiliManager
- ▼ dataAccess
 - ▼ abstracts
 - BlackListRepository
 - CreditRequestDetailsRepository
 - CustomerCreditDetailsRepository
 - CustomerCreditLimitRepository
 - CustomerCreditRepository
 - CustomerCreditRequestRepository
 - CustomerRepository
 - CustomerYetkiliRepository
 - TurkcellAbonesiRepository
 - YeniTesisRepository
 - YetkiliRepository
 - concretes
- ▼ entities
 - ▼ abstracts
 - IEntity
 - ▼ concretes
 - BlackList
 - CreditRequestsDetails
 - Customer
 - CustomerCredit
 - CustomerCreditDetails
 - CustomerCreditLimit
 - CustomerCreditRequest
 - CustomerYetkili
 - TurkcellAbonesi
 - YeniTesis
 - Yetkili
- TurkcellCreditModuleApplication

Database Management

- Database Management aracı olarak PostgreSQL kullanıldı.
- Database tabloları oluşturulurken normalizasyon kurallarına dikkat edildi.



İş Kuralları

- İş kurallarının tamamı business katmanında bulunmaktadır.
- GeneralCreditRequestManager, müşterinin Turkcell abonesi olup olmadığını kontrol ederek iki farklı (ancak abstraction'ı aynı) servisten hangisinin kullanılacağına karar verir.

```
@Service
public class GeneralCreditRequestManager implements IGeneralCreditRequestService {
    @Autowired
    TurkcellAbonesiManager turkcellAbonesiManager;

    @Autowired
    CreditRequestManagerForTurkcellAbonesi creditRequestManagerForTurkcellAbonesi;

    @Autowired
    CreditRequestManagerForYeniTesis creditRequestManagerForYeniTesis;

    private boolean isTurkcellAbonesi(int customerId) {
        return turkcellAbonesiManager.getByCustomerId(customerId) != null;
    }

    public CustomerCreditRequest add(String tcNo, CreditRequestsDetails creditRequestsDetails, byte[] kimlikBelgesi) {
        int customerId = creditRequestManagerForTurkcellAbonesi.getCustomerManager().getByTcNo(tcNo).getId();

        if (isTurkcellAbonesi(customerId))
            return creditRequestManagerForTurkcellAbonesi.add(creditRequestsDetails, tcNo);
        else
            return creditRequestManagerForYeniTesis.add(creditRequestsDetails, tcNo, kimlikBelgesi);
    }
}
```

Turkcell abonesinin kredi talebi eklenirken kara listede olup olmadığı, kredi limitleri ve son bir ayda kredi çekip çekmediği kontrol edilir.

```
@Override
public CustomerCreditRequest add(CreditRequestsDetails creditRequestsDetails, String tcNo) {
    int customerId = getCustomerManager().getByTcNo(tcNo).getId();

    boolean inBlackList = turkcellAbonesiManager.inBlackList(tcNo);
    boolean exceedOneTimeCreditLimit = customerCreditLimitManager.checkOneTimeCreditLimit(customerId) < creditRequestsDetails.getAmount();
    boolean exceedOneYearCreditLimit = customerCreditLimitManager.checkAvailableCreditLimit(customerId) < creditRequestsDetails.getAmount();
    boolean takeCreditWithinLastMonth = customerCreditLimitManager.takeCreditWithinMonth(customerId);

    if (inBlackList || exceedOneTimeCreditLimit || exceedOneYearCreditLimit || takeCreditWithinLastMonth)
        return null;

    return super.add(creditRequestsDetails, tcNo);
}
```

single responsibility
principle

Turkcell abonesi olmayan bir müşterinin kredi talebi eklenirken yeni tesis işlemi de gerçekleşir. Farklı sınıflara ait operasyonlar class içinde injection aracılığıyla yapılır.

```
public CustomerCreditRequest add(CreditRequestsDetails creditRequestsDetails, String tcNo, byte[] kimlikBelgesi) {  
    YeniTesis yeniTesis = new YeniTesis();  
    yeniTesis.setKimlikBelgesi(kimlikBelgesi);  
    yeniTesis.setCustomerId(getCustomerManager().getByTcNo(tcNo).getId());  
    yeniTesisManager.add(yeniTesis);  
    return super.add(creditRequestsDetails, tcNo);  
}
```

```
@Service  
public class CreditRequestManagerForYeniTesis extends CreditRequestManager {  
  
    @Autowired  
    YeniTesisManager yeniTesisManager;  
}
```


Yetkili, customerCreditManager'ı kullanarak müşterinin kredi talebini onaylayabilir, güncellemesini isteyebilir veya reddedebilir.

Update etmesi durumunda, eğer müşteri kredisinin güncellenmiş halini onaylarsa kredi talebi onaylanır, aksi halde kredi istediği silinir.

```
@Service
public class YetkiliManager implements IYetkiliService {

    @Autowired
    YetkiliRepository yetkiliRepository;

    @Autowired
    CustomerYetkiliManager customerYetkiliManager;

    @Autowired
    CreditRequestManager creditRequestManager;

    @Autowired
    CustomerCreditManager customerCreditManager;
```

```
public CustomerCredit add(int requestId) {
    return customerCreditManager.add(creditRequestManager.getByRequestId(requestId));
}
```

```
public String update(CreditRequestsDetails updatedCreditRequestDetails, boolean response) {
    return customerCreditManager.update(updatedCreditRequestDetails, response);
}
```

```
public String reject(int requestId) {
    return customerCreditManager.reject(creditRequestManager.getByRequestId(requestId));
}
```

Yetkili kendisine gelen talepleri listeleyebilir.

```
public List<CustomerCreditRequest> getAllRequestsByCustomerId(int yetkiliId) {  
    List<CustomerCreditRequest> customerCreditRequests = new ArrayList<>();  
  
    for (int customerId : customerYetkiliManager.getAllCustomersByYetkiliId(yetkiliId)) {  
        customerCreditRequests.add(creditRequestManager.getByCustomerId(customerId));  
    }  
    return customerCreditRequests;  
}
```