# UDP Mesaj Alıcı / Gönderici Tasarım Raporu

UDP mesaj alıcı / gönderici uygulaması kullanıcı tarafından belirlenen bir IP adresi ve port numarasını dinleyerek UDP protokolü üzerinden gönderilen mesajları dinler ve yine kullanıcı tarafından belirlenen bir IP adresi ve port numarasına mesaj gönderir. Gönderilen ve alınan mesajlar veritabanında detay bilgileri ile kaydedilir. Kullanıcı bu mesajlardan istediğini silebilir veya tarih aralığı girerek kayıtlı mesajları filtreleyebilir.

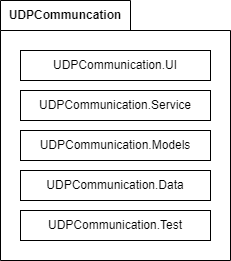
## Kullanılan Teknolojiler

Uygulamanın geliştirilmesinde N katmanlı mimari esas alınmıştır. Geliştirme ortamı olarak Windows 10 Education 64 bit işletim sistemi üzerinde Visual Studio 2022 Community Edition kullanılmıştır. Kaynak kod geliştirilmesi sırasında kullanılan teknolojiler:

* Visual C#
* .NET Framework 6.0
* NHibernate 5.4.6
* PostgreSQL 16
* SHA-256 şifreleme algoritması

## Kaynak Kodu Oluşturan Projeler

Uygulamanın kaynak kod yapısı, Visual Studio üzerinde açılan “UDPCommunication” isimli solution içine eklenen 5 adet projeden oluşmaktadır. Projeler Çizim 1’de gösterilmektedir.



Çizim 1: Proje Yapısı

**UDPCommunication.UI**

Uygulamaya ait arayüz bileşenlerinin bulunduğu projedir. Arayüzler XAML isimli XML temeline dayanan teknoloji ile geliştirilmiştir. Bu katmanda kullanılacak servisler Dependency Injection (DI) kullanılarak projeye dahil edilmiştir.

**UDPCommunication.Service**

Kullanıcının arayüz üzerinde kullanacağı fonksiyonları içeren sınıfların bulunduğu projedir. Ayrıca veritabanı işlemlerinde arayüz ile veri katmanı arasında köprü görevi görür.

**UDPCommunication.Models**

Uygulama seviyesinde kullanılacak özelliklerin gereksinimlerini tanımlayan model sınıflarının bulunduğu projedir. Ayrıca “EventArgs” olarak kullanılacak sınıf ve “Enumeration” sınıfları da bu projede bulunur.

**UDPCommunication.Data**

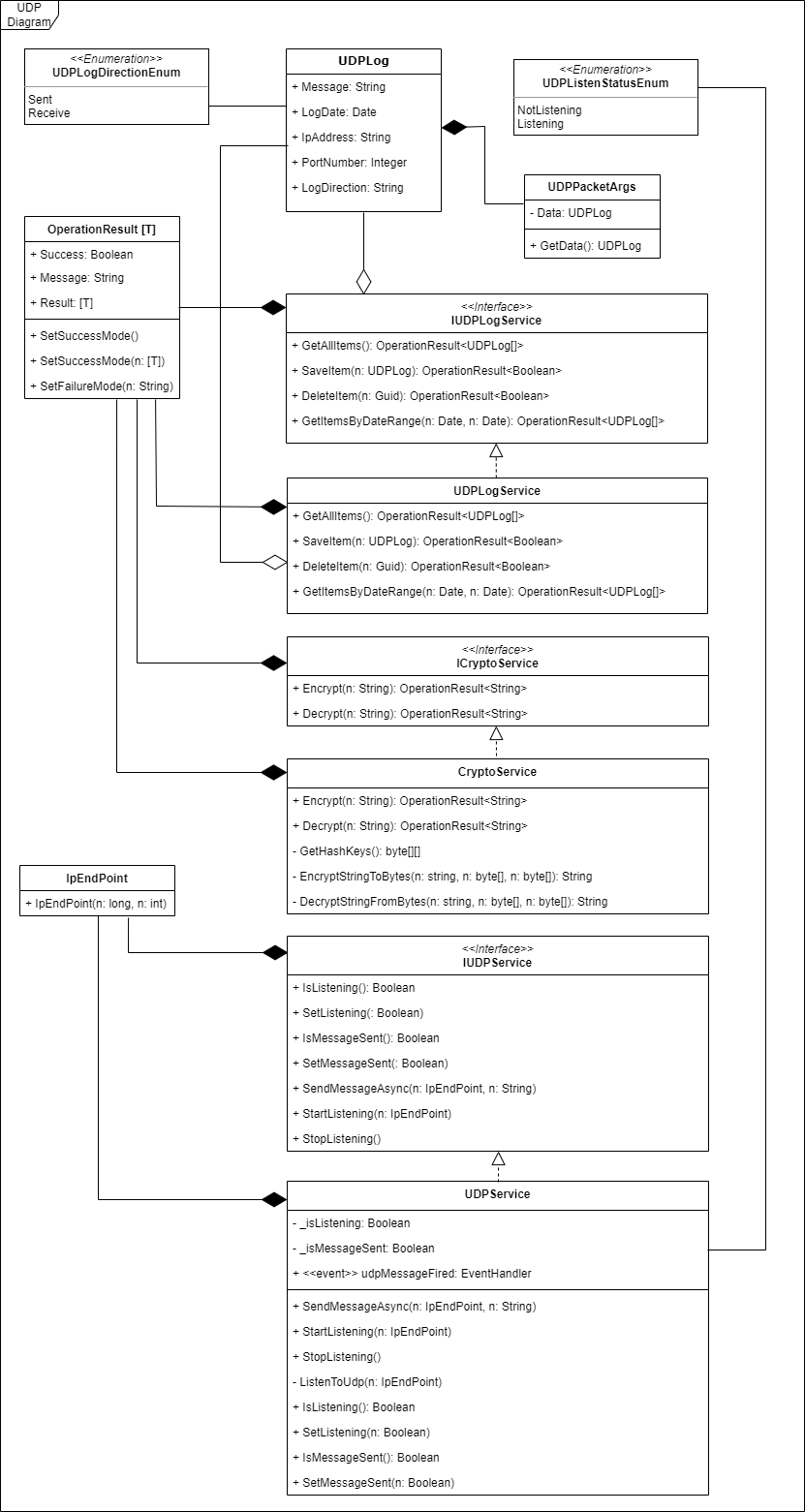
Uygulama üzerinde veri okuma, veri silme ve veri kaydetme işlemlerinin yapıldığı sınıfları içeren projedir. Projeye NuGet referansı olarak eklenen NHibernate teknolojisi ile PostgreSQL veritabanı ile iletişim kurulabilir.

**UDPCommunication.Test**

Servis katmanında ihtiyaç duyulan tüm metotların birim testlerinin bulunduğu test projesidir. Bu katmanda kullanılacak servisler Dependency Injection (DI) kullanılarak projeye dahil edilmiştir.

## UML Sınıf Diyagramı ve Açıklaması

Projeye ait UML sınıf diyagramı Çizim 2’deki gibidir. Devamında diyagramda kullanılan sınıflar detaylı olarak tanımlanmıştır.



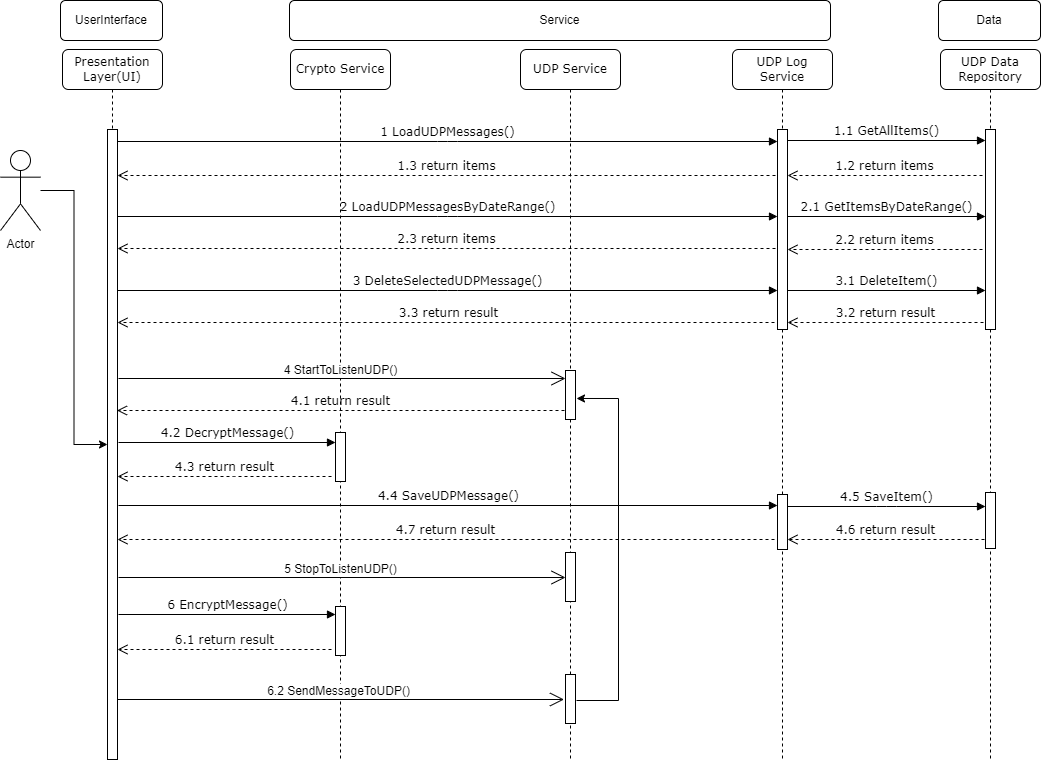
Çizim 2: UML Sınıf Diyagramı

Sınıf diyagramında tanımlanan sınıflar ve detaylı açıklamaları aşağıdaki gibidir:

* **UDPListenStatusEnum**: UDP Ip ve portunun dinleme modunda olup olmadığı tutan sınıftır.
* **UDPLogDirectionEnum**: Bir mesajın UDP portu üzerinden gönderildiği veya alındığı bilgisini tutan sınıftır.
* **UDPLog**: Uygulama seviyesinde gönderilen veya alınan mesajları detaylarıyla tanımlamaya yarayan nesne modelidir. Mesajın içeriği, gönderilme veya alınma zamanı, IP adresi, port numarası ve mesajın gönderilme / alınma bilgilerini tutar.
* **UDPPacketArgs**: UDP mesajı gönderildiği veya okunduğu zaman tetiklenen olayda parametre olarak kullanılan sınıf modelidir.
* **OperationResult[T]:** Uygulama bazında gerçekleştirilen tüm servis işlemlerinin dönüş modelini oluşturan sınıftır. Servisten dönen veri tipine göre “Generic Type” kullanımı ile geliştirilmiştir. Servis işlemi sırasında hata olup olmadığı bilgisi ile hata varsa detaylı hata mesajını da barındırır.
* **IpEndPoint**: Ip adresi ve port numarasını içeren adres modelidir. Bu sınıf .NET Framework seviyesinde tanımlıdır.
* **UDPLogService**: Gönderilen ve alınan mesajların veritabanı seviyesinde yönetimini yapmaya yarayan kayıt, silme ve sorgulama işlemlerini bulunduran sınıftır.
* **IUDPLogService**: UDPLogService servisi için tanımlanan arayüz sınıfıdır.
* **CryptoService**: Gönderilen ve alınan mesajların SHA-256 algoritması ile şifrelendiği ve şifreli mesajın ise tanımlı anahtar ile şifresinin çözüldüğü sınıftır.
* **ICryptoService**: CryptoService servisi için tanımlanan arayüz sınıfıdır.
* **UDPService**: UDP protokolü üzerinden mesaj gönderimi ve okunması işlemlerini yöneten sınıftır. Mesaj gönderimi, ilgili Ip adresi ve port numarasının dinlenmesi veya dinlenmesinin durdurulması gibi işlemler asenkron olarak tanımlanmıştır. Mesaj gönderildiğinde veya okunduğunda tetiklenen bir event(olay) mekanizması barındırır. Bu sayede mesaj gönderildiği veya okunduğu an, kullanıcı arayüz katmanından ilgili mesaja ulaşılabilir.
* **IUDPService**: UDPService servisi için tanımlanan arayüz sınıfıdır.

## UML Sıralama Diyagramı ve Açıklaması

Projeye ait UML sıralama diyagramı Çizim 3’teki gibidir. Devamında diyagramda kullanılan katmanlar ve işlemler detaylı olarak tanımlanmıştır.



Çizim 3: UML Sıralama Diyagramı

Sıralama diyagramında tanımlanan diyagram katmanlar ve detaylı açıklamaları aşağıdaki gibidir:

* **Presentation Layer (UI):** Kullanıcı arayüz ekranlarının bulunduğu katmandır. Kullanıcı uygulamayı bu arayüz üzerinden kullanır. Arayüze dahil olan tüm bileşenler (TextBox, Button, Grid vb.) ile bu bileşenleri yönetmeyi sağlayan kaynak kodlar bu katmandadır.

N katmanlı uygulama mimarisinde “UserInterface” katmanı olarak geçmektedir.

* **Crypto Service**: UDP protokolü üzerinden gönderilecek v alınacak mesajların SHA-256 algoritmasına göre şifrelendiği ve şifreli mesajın şifresinin çözüldüğü katmandır.

N katmanlı uygulama mimarisinde “Service” katmanında yer almaktadır.

* **UDP Service**: UDP protokolü üzerinden tanımlı Ip adresi ve port numarasını dinlemeyi, ilgili Ip adresi ve port numarasına mesaj göndermeyi yöneten katmandır.

N katmanlı uygulama mimarisinde “Service” katmanında yer almaktadır.

* **UDPLogService**: UDP protokolü üzerinden gönderilen ve alınan mesajların detayları ile birlikte tutulduğu “UDPLog” nesnesinin veritabanı okuma, yazma ve silme işlemlerini yapacak olan veri katmanı ile haberleşmesini sağlayan katmandır.

N katmanlı uygulama mimarisinde “Service” katmanında yer almaktadır.

* **UDPDataRepository**: UDPLog nesnesinin veritabanı okuma, yazma ve silme işlemlerini yapacak olan katmandır. Bu katman NHibernate isimli kalıcılık mekanizmasını kullanarak uygulama ile PostgreSQL veritabanı arasında haberleşmesi sağlar.

N katmanlı uygulama mimarisinde “Data” katmanında yer almaktadır.