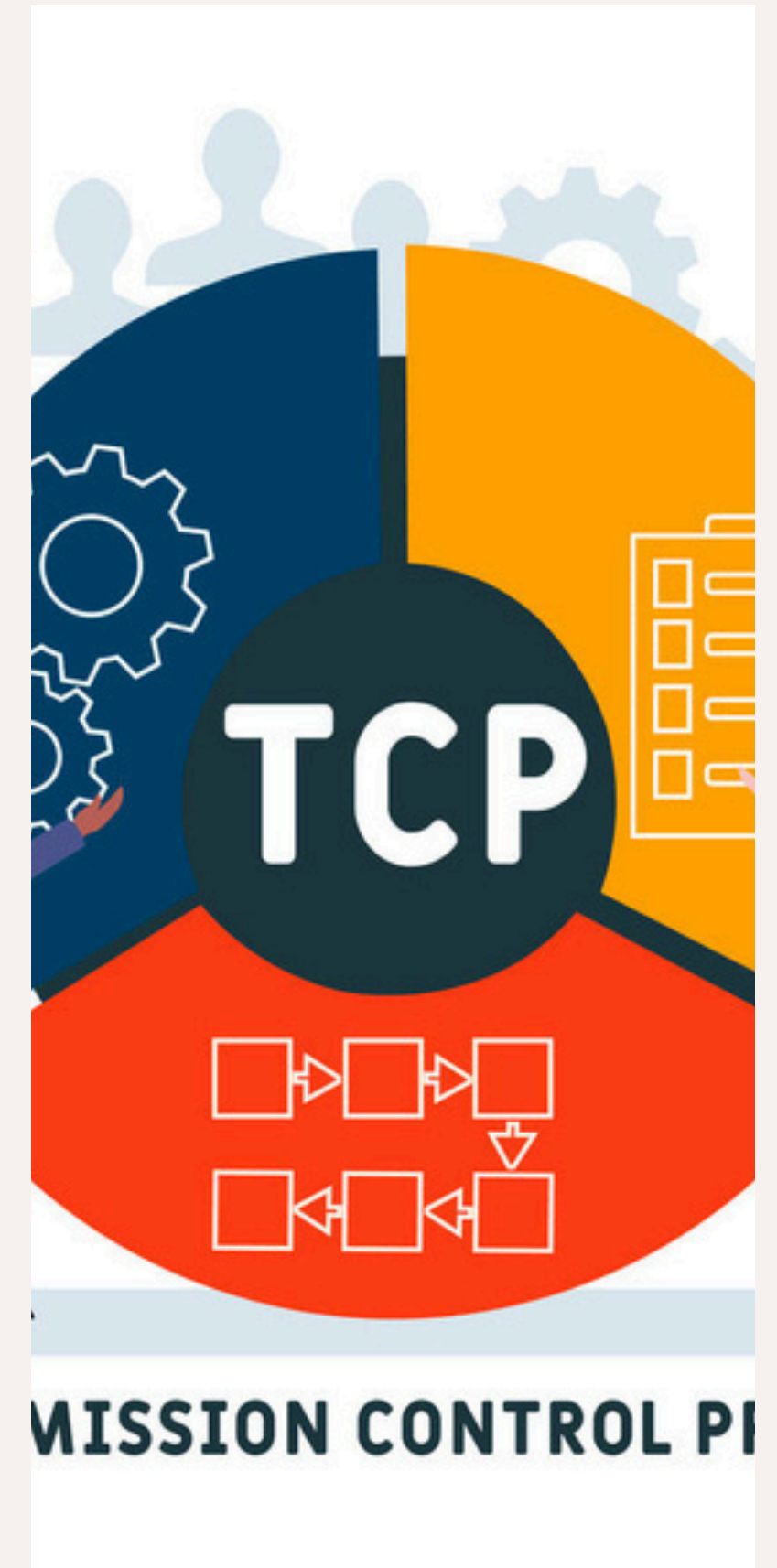
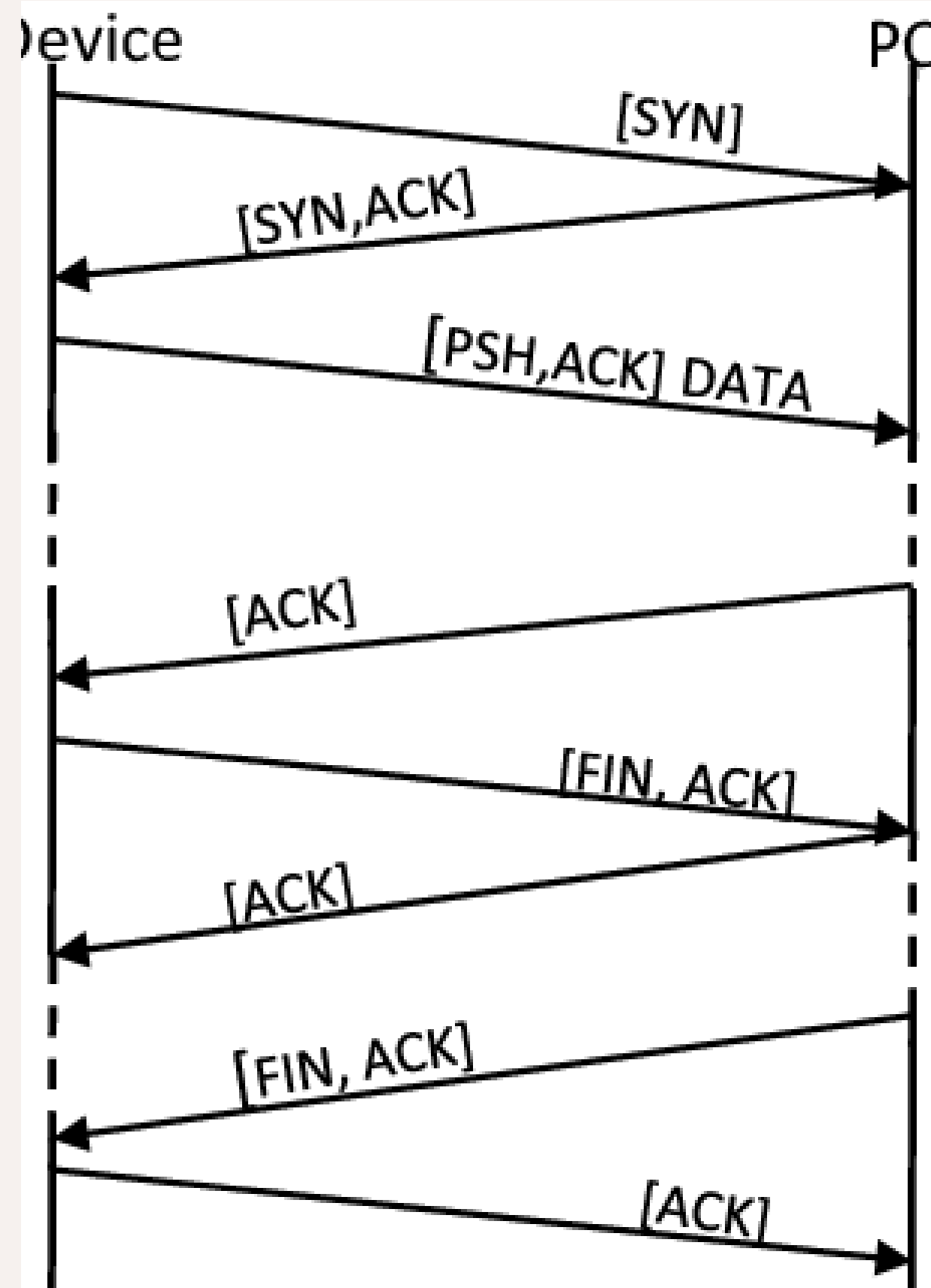


Soket Programlama Sunumu



01

Login Ekranı

Kullanıcı login olma işlemlerini bu sayfadan yapacak, eğer kullanıcı hesabı yoksa create new acoount kısmından yeni hesap oluşturabilecek.

Login

— □ ×

LOGIN

kullanıcı adı:

şifre:

login

create a new account

02

Registration

Üye olma ekranımız. Kullanıcı bu ekranda kullanıcı adı, şifre, ve dinleme yapması için port numarasını belirleyebilecek. standart olarak 5000 portu seçilmiştir.

Registration

—□×

login

REGISTRATION

kullanıcı adı:

şifre:

onayla:

port:

5000

create a new account

03

Registration

Herhangi bir textBox boş bırakılmadan (hepsinin hata kısmı yazıldı) create new account kısmına basıldığında kullanıcının hesabı kaydediliyor. Kaydedilen kullanıcının bilgileri database de bir sonraki kısımdaki gibi görülecek.

Registration

— □ ×

login

REGISTRATION

Kayıt Başarılı !!

×

Kaydınız başarıyla alınmıştır. Aramıza Hoşgeldin aabb

Tamam

create a new account

04

Users Table

	id	username	password	isOnline	ipAddress	port
1	1	myakn55	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	5000
2	2	esra	02d20bbd7e394ad5999a4cebabac9619732c343a4cac9947...	0	192.168.56.1	5200
3	3	deneme	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	5100
4	4	portdeneme	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	4900
5	5	aabb	961b6dd3ede3cb8ecbaacbd68de040cd78eb2ed5889130cc...	0	NULL	5000

Burada görüldüğü gibi users tablomuz şu sütunlardan oluşuyor: ID, USERNAME, PASSWORD, ISONLINE, IPADRESS, PORT. Password kısmı projenin amacı güvenli bir mesajlaşma uygulaması yapmak olduğu için SHA-256 ile şifrelenerek kaydediliyor. Ayrıca ilk kayıt olduğunda kullanıcı ip adresi null olarak belirleniyor, fakat kullanıcı giriş yaptığında ipAdress kısmı güncelleniyor ve kullanıcının ip adresi getLocalIpAdress() fonksiyonuyla alınıp kullanıcının ip adres kısmı güncellenmiş oluyor. Kullanıcı her yeni giriş yaptığında bu sorgu çalışıyor ve kullanıcının ip adresi sürekli güncel kalmış oluyor.

Users Table

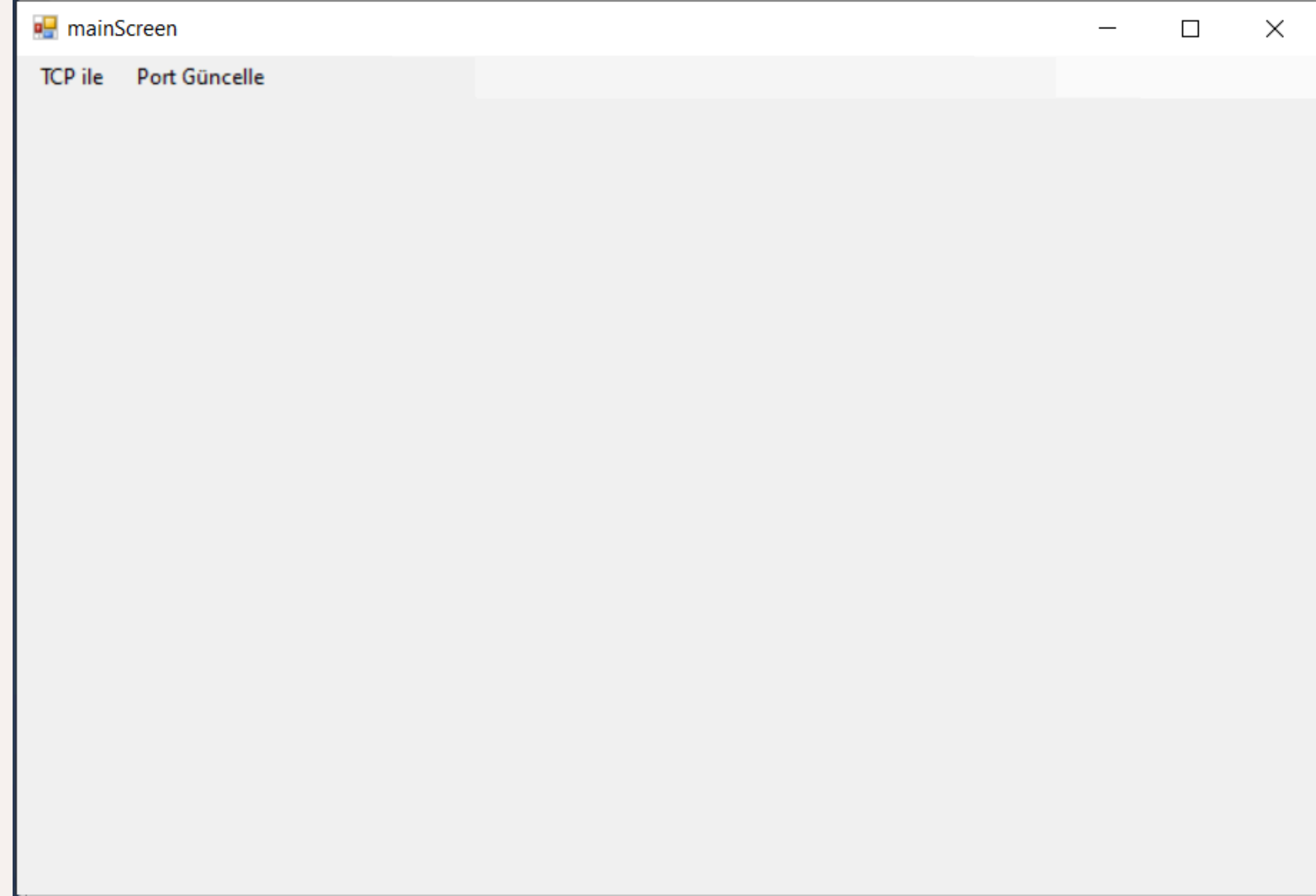
	id	username	password	isOnline	ipAddress	port
1	1	myakn55	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	5000
2	2	esra	02d20bbd7e394ad5999a4cebabac9619732c343a4cac9947...	0	192.168.56.1	5200
3	3	deneme	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	5100
4	4	portdeneme	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	4900
5	5	aabb	961b6dd3ede3cb8ecbaacbd68de040cd78eb2ed5889130cc...	0	NULL	5000

isOnline, kısmıyla projenin bizden istemiş olduğu sunucuya bağlı tüm kullanıcılara mesaj at kısmını ve kullanıcı offline ise TCP ile değil database yoluyla mesaj gönderilmesi sorgusunu yapmış olmayı hedefledik. Ayrıca kullanıcı offline ise ona mesajı database yoluyla göstereceğiz. Eğer online ise karşı tarafın az sonra göreceğimiz mainscreen ekranında bir sunucu başlatmış olmasını bekleyeceğiz. Eğer kullanıcı online ise, ve bir diğer kullanıcı bu online kullanıcıya mesaj atmak istiyorsa karşı tarafa erişemeyeceği için bir hata mesajı dönecek. Fakat proje bizden offline kullanıcılara da mesaj atmamızı istediği için bu sorgulamaları yapmamız gerekecekti.

06

Ana Ekran

Kullanıcı giriş yaptığında onu karşılayacak ekran böyle. yukarıda menuStripimiz var ve kullanıcı bu menustripler ile aralarında geçiş yapabilecek.



Users Table

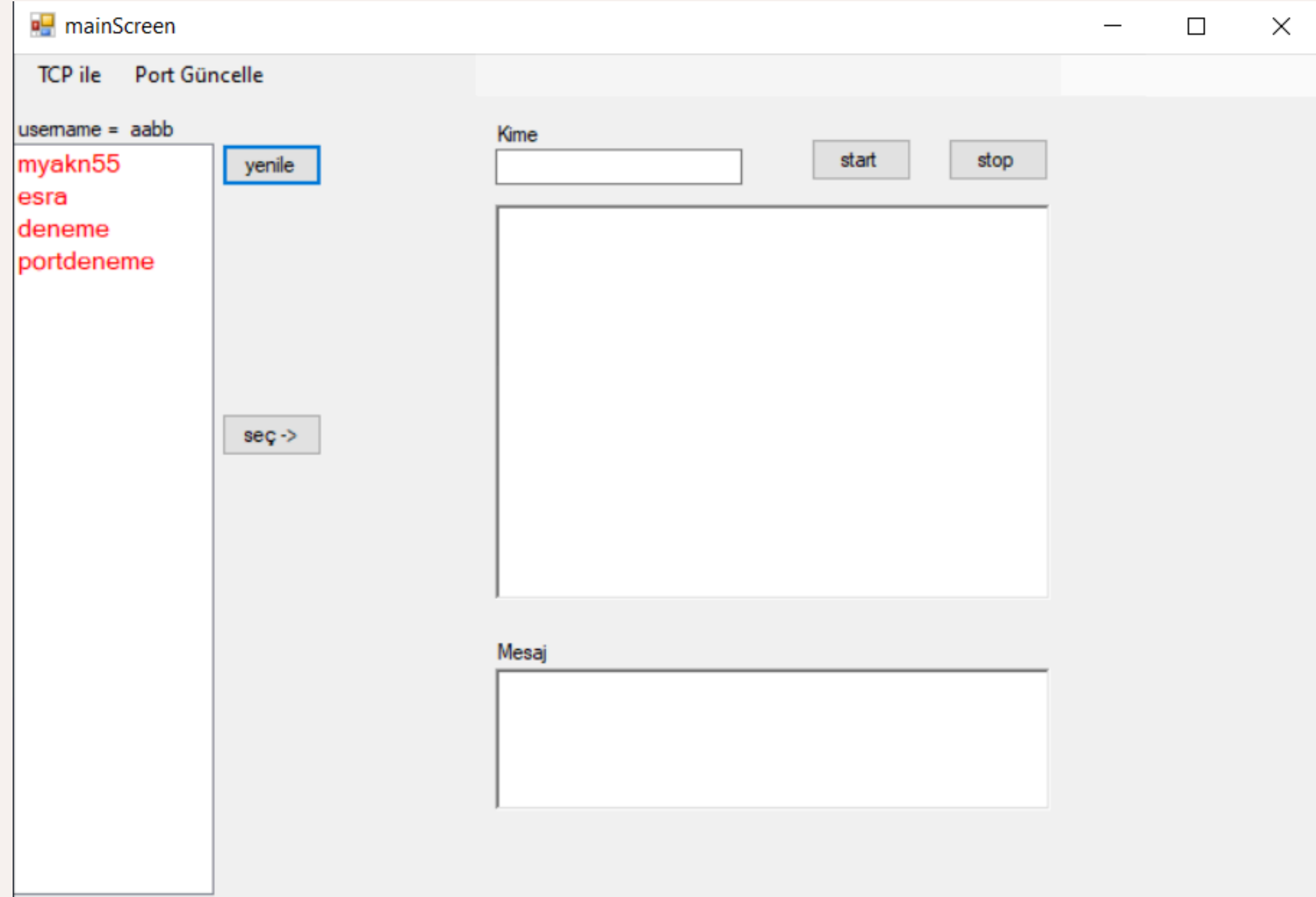
	id	username	password	isOnline	ipAddress	port
1	1	myakn55	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	5000
2	2	esra	02d20bbd7e394ad5999a4cebabac9619732c343a4cac9947...	0	192.168.56.1	5200
3	3	deneme	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	5100
4	4	portdeneme	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb3206...	0	192.168.56.1	4900
5	5	aabb	961b6dd3ede3cb8ecbaacbd68de040cd78eb2ed5889130cc...	1	192.168.56.1	5000

Kullanıcı giriş yaptıktan sonra isOnline kısmı ve ipAddress kısmı güncelleniyor. Ayrıca bu isOnline kısmını FormClosing eventlerinde sürekli kontrol ediyoruz eğer bir form kapanacak olursa her sayfada tutmuş olduğumuz userID'nin online kısmını 0 a çekiyoruz.

08

Ana Ekran

Kullanıcı yenile butonuna bastığında users tablosundaki tüm kullanıcıları görebilecek. Offline olan kullanıcılar kırmızı ile online olan kullanıcılar ise Yeşil renkte görünecek. Sol kısmı oluştururken listView kullandık. Mesaj görüntüleme ve mesaj yazma kısımları ise richTextBox kullanılarak yapılmıştır. Ayrıca kullanıcı seç butonuna basmadan altta gonder butonu görülmeyecek şekilde tasarım yapılmıştır.



09

Tek Kullanıcı

Tek kullanıcı seçildiğinde ekran böyle görünecek. Üstte kime gönderileceği görünecek.

The screenshot shows a Java Swing window titled "mainScreen" with a standard Windows-style title bar (minimize, maximize, close buttons). The window has a light gray background and is divided into several sections. At the top, there are two tabs: "TCP ile" (selected) and "Port Güncelle". Below the tabs, on the left, is a text area labeled "username = aabb" containing a list of usernames: "myakn55", "esra", "deneme", and "portdeneme". To the right of this list is a "yenile" (refresh) button. Below the list is a "seç ->" (select) button. On the right side of the window, there is a section labeled "Kime" (To) with a text input field containing "myakn55". To the right of this field are "start" and "stop" buttons. Below the "Kime" section is a large, empty rectangular area. At the bottom right, there is a section labeled "Mesaj" (Message) with a text input field and a "gonder" (send) button.

10

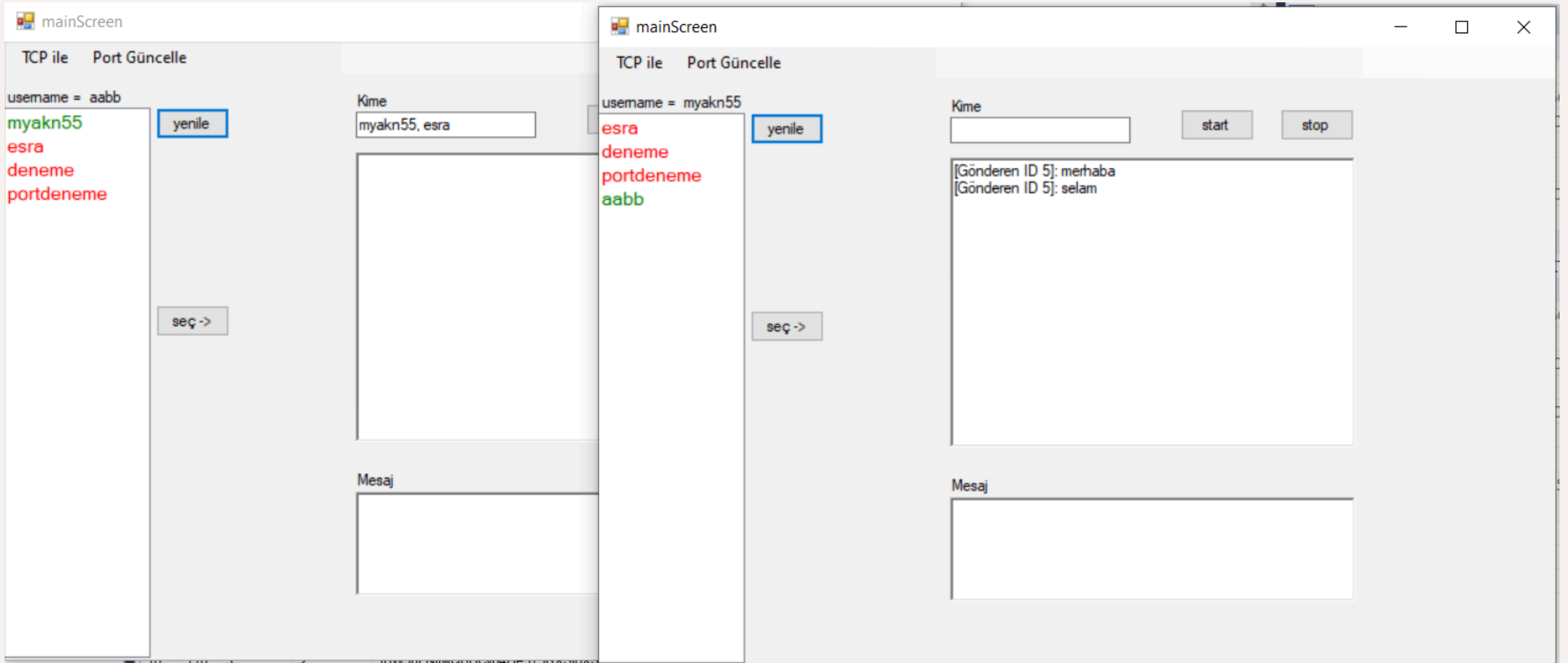
Çok Kullanıcı

Çok kullanıcı listview kısmında seçilip yukarıdaki textboxa yazılacak. Bu kısım sadece görüntü olarak oluşturulmamış, yukarıdaki kullanıcıları trim fonksiyonuyla birlikte ID leri alınacak bir fonksiyon da yazılmıştır.

The screenshot shows a Windows application window titled "mainScreen". The interface is divided into several sections:

- Top Bar:** Contains the text "TCP ile" and "Port Güncelle".
- User List:** A listview on the left containing the following items: "myakn55", "esra", "deneme", and "portdeneme". The last two items are highlighted in red.
- Buttons:** A "yenile" button is next to the listview. A "seç ->" button is located below the listview.
- Time Input:** A section labeled "Kime" with a text input field containing "myakn55, esra".
- Time Controls:** "start" and "stop" buttons are located next to the time input field.
- Message Input:** A section labeled "Mesaj" with a large text input field.
- Send Button:** A "gonder" button is located at the bottom right of the message input field.

11



Offline Kullanıcıya Mesaj

Offline kullanıcıya mesaj gönderildiğinde TCP kısmı çalışmayacak fakat databaseye kaydı tutulacaktır. Database kısmında yaptığımız tasarımı aşağıda görebilirsiniz. Mesaj gönderilmeden önce kullanıcının online olup olmadığı users tablosunda kontrol ediliyor daha sonra kullanıcı offline ise database kısmına kaydetmek için fonksiyonumuz devreye giriyor, ve isDelivered kısmı 0 olarak işaretleniyor. Offline olan kullanıcı daha sonra online olup ekrana giriş yaptığında ilk çalıştırılan fonksiyonlarımızdan biri bu mesajları var mı kontrol etmek oluyor. messages tablosunda tutulan bilgiler şunlardır: ID, SENDER_ID, RECEIVER_ID, MESSAGE, SENT_AT, ISDELIVERED. sender_id ile gönderen kişinin idsini receiver_id ile kime gönderildiği message ile AES formatında şifrelenmiş mesajı ve sent_at kısmında tarihi isDelivered ile kullanıcıya ulaşmış olup olmadığını kontrol ediyoruz. Son mesajları buradan görebilirsiniz offline kullanıcıya gönderildiği için isDelivered kısmı 0 olarak işaretlenmiştir.

Aes kısmını bir Class içinde oluşturduk.

	id	sender_id	receiver_id	message	sent_at	isDelivered
1	119	5	2	IZZHA0B48cUXbokpbAXpWg==	2024-12-27 17:39:41.403	0
2	118	5	1	IZZHA0B48cUXbokpbAXpWg==	2024-12-27 17:39:41.393	0
3	117	5	2	oW1bKCVOavmSMndjx/vnvQ==	2024-12-27 17:39:24.787	0
4	116	5	1	oW1bKCVOavmSMndjx/vnvQ==	2024-12-27 17:39:24.723	0
5	115	1	2	JB3Yii4pl/bmBBCJbOpfsenivZeAd4YMQo2LzREYQw=	2024-12-25 02:36:12.663	1
6	114	2	3	FH20tvt+FNo2RYLNCgfSLuN5IVMLDrepwwiu2C/qik=	2024-12-25 02:24:14.240	1
7	113	2	3	FH20tvt+FNo2RYLNCgfSLuN5IVMLDrepwwiu2C/qik=	2024-12-25 02:24:14.233	1
8	112	2	1	FH20tvt+FNo2RYLNCgfSLuN5IVMLDrepwwiu2C/qik=	2024-12-25 02:24:13.383	1
9	111	2	1	FH20tvt+FNo2RYLNCgfSLuN5IVMLDrepwwiu2C/qik=	2024-12-25 02:24:13.377	1
10	110	3	2	fBW5nLNjtwQBDCyp4He5+5RXSloXShl2DL+wWnRvASc=	2024-12-25 02:23:52.320	1
11	109	3	2	fBW5nLNjtwQBDCyp4He5+5RXSloXShl2DL+wWnRvASc=	2024-12-25 02:23:52.317	1

```

public class AesEncryption
{
    private static readonly string key = "12345678901234567890123456789012"; // 32 karakter uzunluğunda bir anaht
    private static readonly string iv = "1234567890123456"; // 16 karakter uzunluğunda bir IV

    1 başvuru
    public static string Encrypt(string plainText)
    {
        using (Aes aes = Aes.Create())
        {
            aes.Key = Encoding.UTF8.GetBytes(key);
            aes.IV = Encoding.UTF8.GetBytes(iv);

            using (MemoryStream ms = new MemoryStream())
            {
                using (CryptoStream cs = new CryptoStream(ms, aes.CreateEncryptor(), CryptoStreamMode.Write))
                {
                    byte[] plainBytes = Encoding.UTF8.GetBytes(plainText);
                    cs.Write(plainBytes, 0, plainBytes.Length);
                    cs.FlushFinalBlock();
                }
                return Convert.ToBase64String(ms.ToArray());
            }
        }
    }

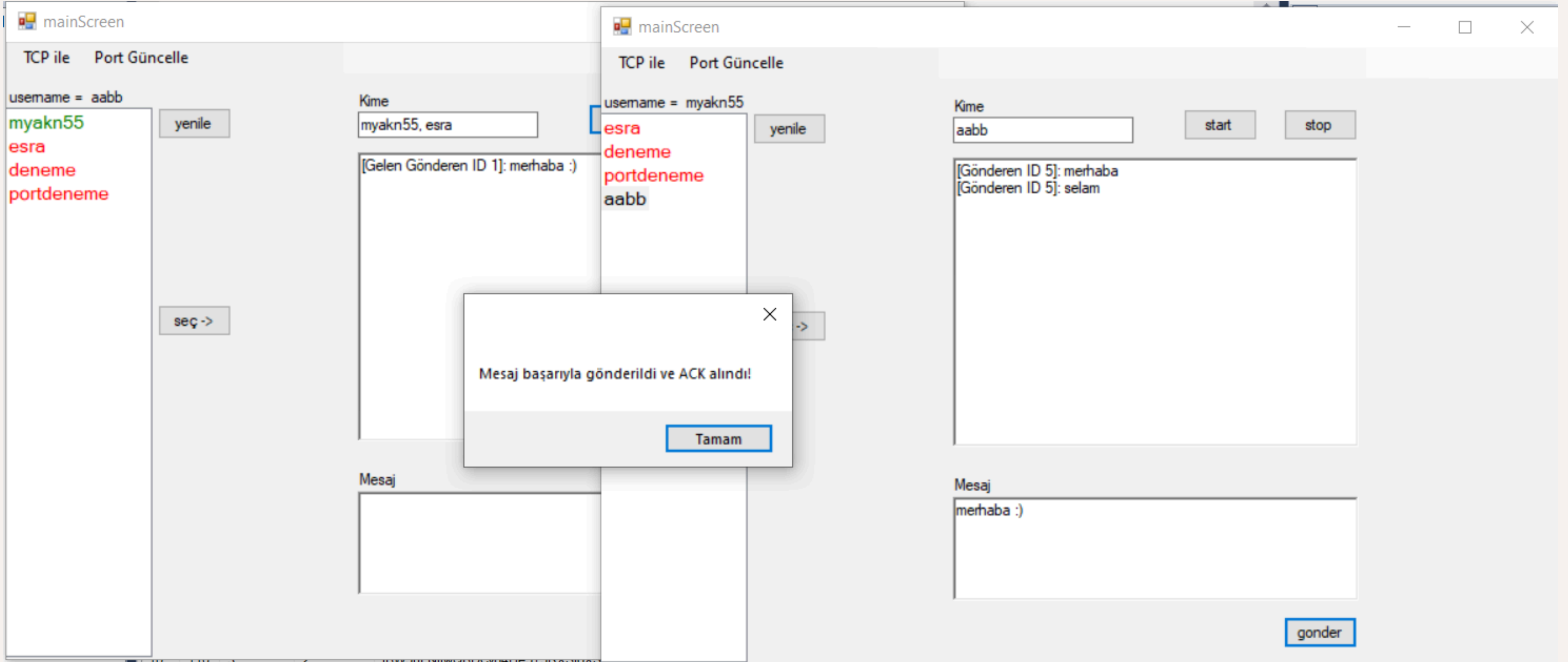
    1 başvuru
    public static string Decrypt(string encryptedText)
    {
        using (Aes aes = Aes.Create())
        {
            aes.Key = Encoding.UTF8.GetBytes(key);
            aes.IV = Encoding.UTF8.GetBytes(iv);

            using (MemoryStream ms = new MemoryStream(Convert.FromBase64String(encryptedText)))
            {
                using (CryptoStream cs = new CryptoStream(ms, aes.CreateDecryptor(), CryptoStreamMode.Read))
                {
                    using (StreamReader sr = new StreamReader(cs))
                    {
                        return sr.ReadToEnd();
                    }
                }
            }
        }
    }
}

```

13

14



Online Kullanıcıya Mesaj

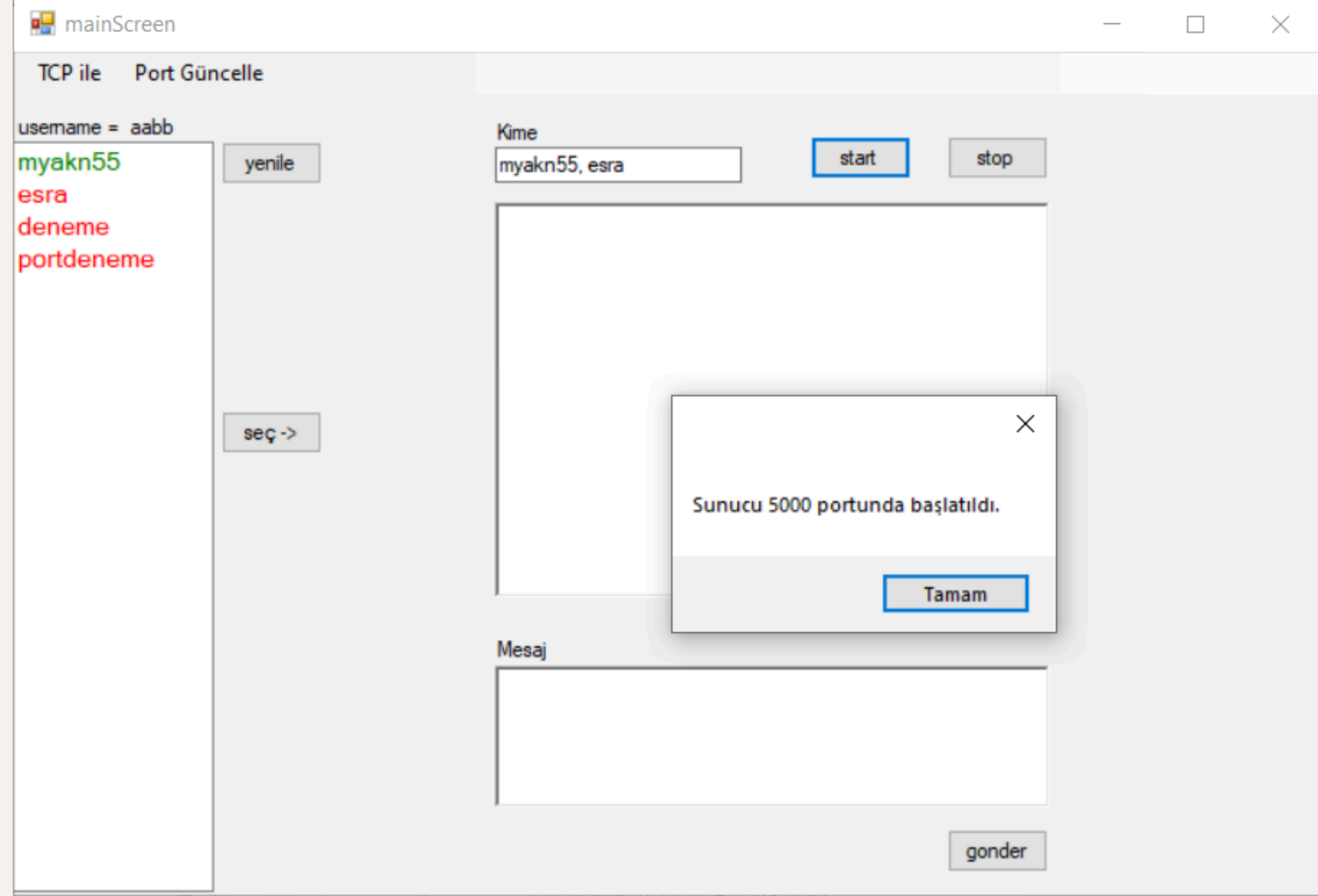
Eğer kullanıcı online ise burada izlenecek yol TCP ile mesaj gönderimidir. Burada TCPlistener ile client start butonua basarak bir portu sürekli dinliyor. Karşı taraf mesajı göndereceği ip adresi ve portu zaten users tablosundan çekebiliyor. Eğer karşı tarafın port bilgisi yanlışsa karşı tarafa iletilemediğine dair bir hata mesajı bize geri dönüyor, veya karşı taraf dinlemeyi başlatmadıysa eğer mesaj yine alıcı tarafa iletilemiyor. Eğer mesaj düzgün bir şekilde iletilirse bir ACK mesajı bize geri dönüyor. Bu kısmın çoğunu TCPlistener ile yaptık kodların bir kısmını yapay zeka ile bir kısmını ise stackoverflow üzerinden alarak yapmaya çalıştık. Ayrıca online bir kullanıcıya mesaj atarsak isdelivered kısmı 1 olduğunu görebiliyoruz.

	id	sender_id	receiver_id	message	sent_at	isDelivered
1	121	1	5	KNnt4hR8YeVJW8xnJacNNA==	2024-12-27 17:42:09.013	1
2	120	1	5	KNnt4hR8YeVJW8xnJacNNA==	2024-12-27 17:42:08.990	1
3	119	5	2	IZZHA0B48cUXbokpbAXpWg==	2024-12-27 17:39:41.403	0
4	118	5	1	IZZHA0B48cUXbokpbAXpWg==	2024-12-27 17:39:41.393	1
5	117	5	2	oW1hKCVQavmSMndix/vnvQ==	2024-12-27 17:39:24.787	0

16

Port Başlatma

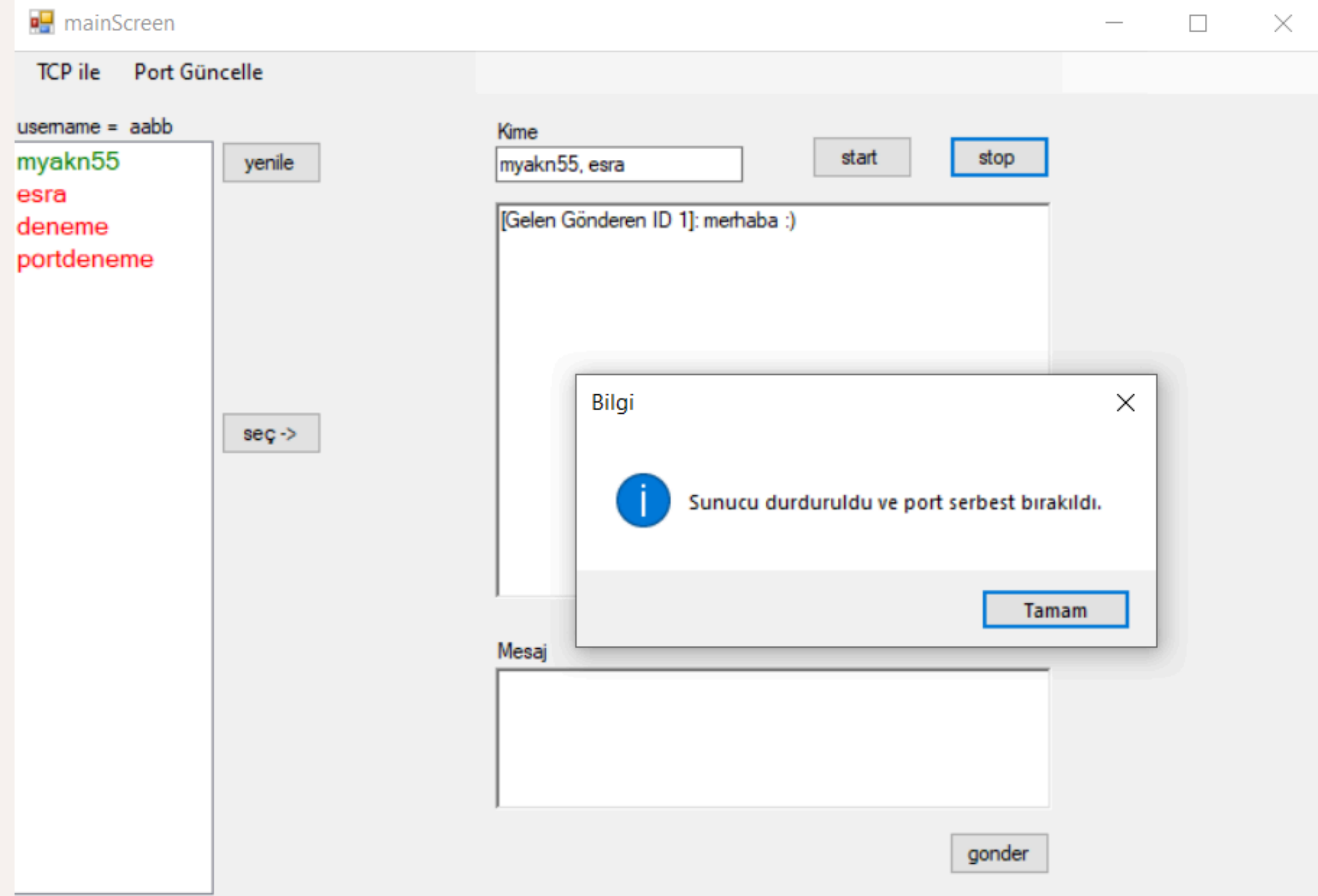
start butonu ile listener eventi başlatılıyor ve 5000 portundan dinleme yapılabilir. Eğer bunu yapmadan karşı taraftan mesaj almaya çalışırsak karşı tarafın mesajı bize ulaşmıyor.



17

Port Durdurma

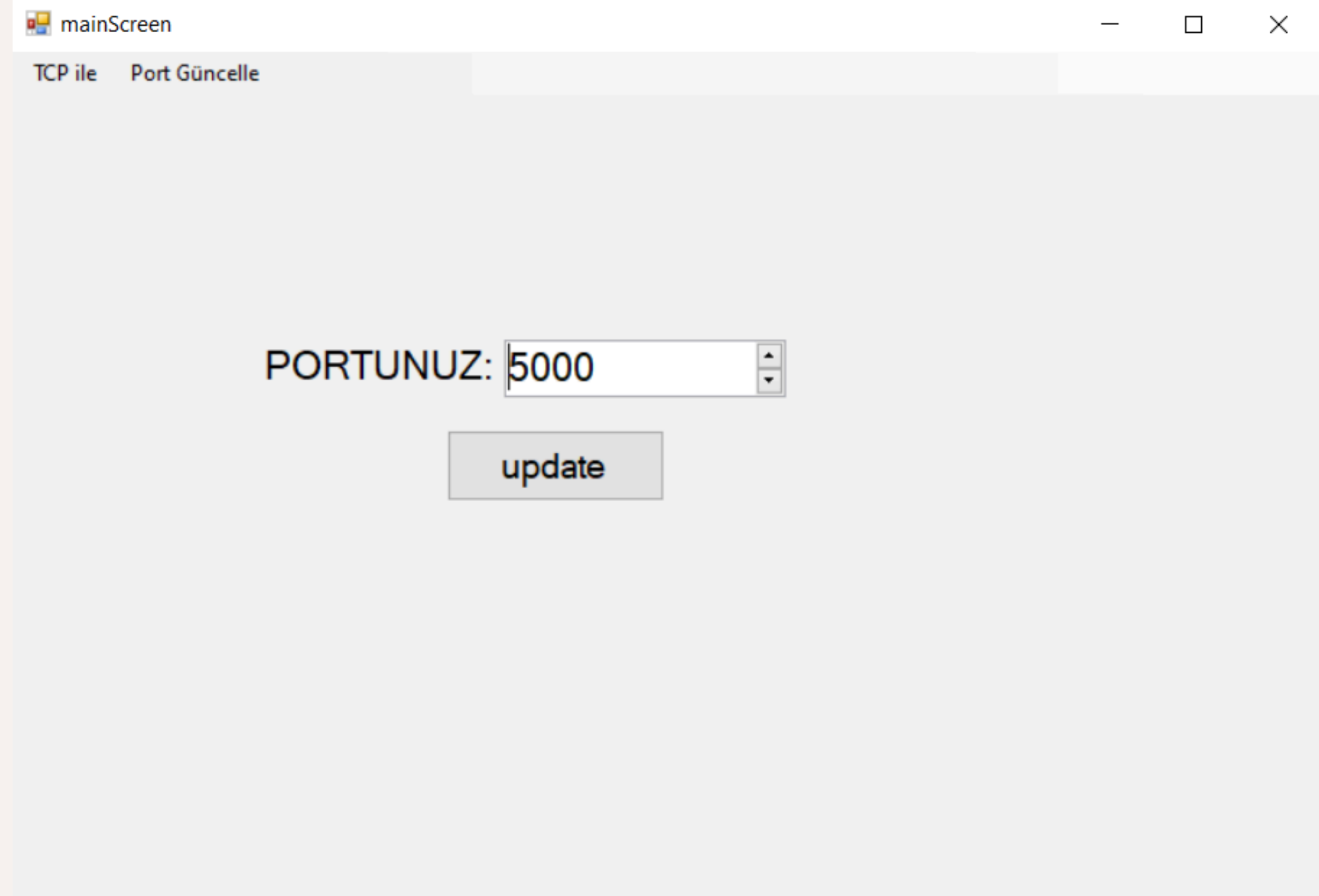
Stop butonu ile dinlemeye başladığımız portu serbest bırakabiliyoruz, bunu yapma amacımız iki farklı kullanıcı aynı portu dinleyemiyor TCP buna izin vermiyor. başka bir kullanıcı daha aynı port değerine sahipse start komutunu bastığında hata mesajıyla karşılaşılıyor. Ayrıca bu kısmı formclosing eventine de ekledim program kapatıldığında port serbest bırakılıyor.



18

Port Update

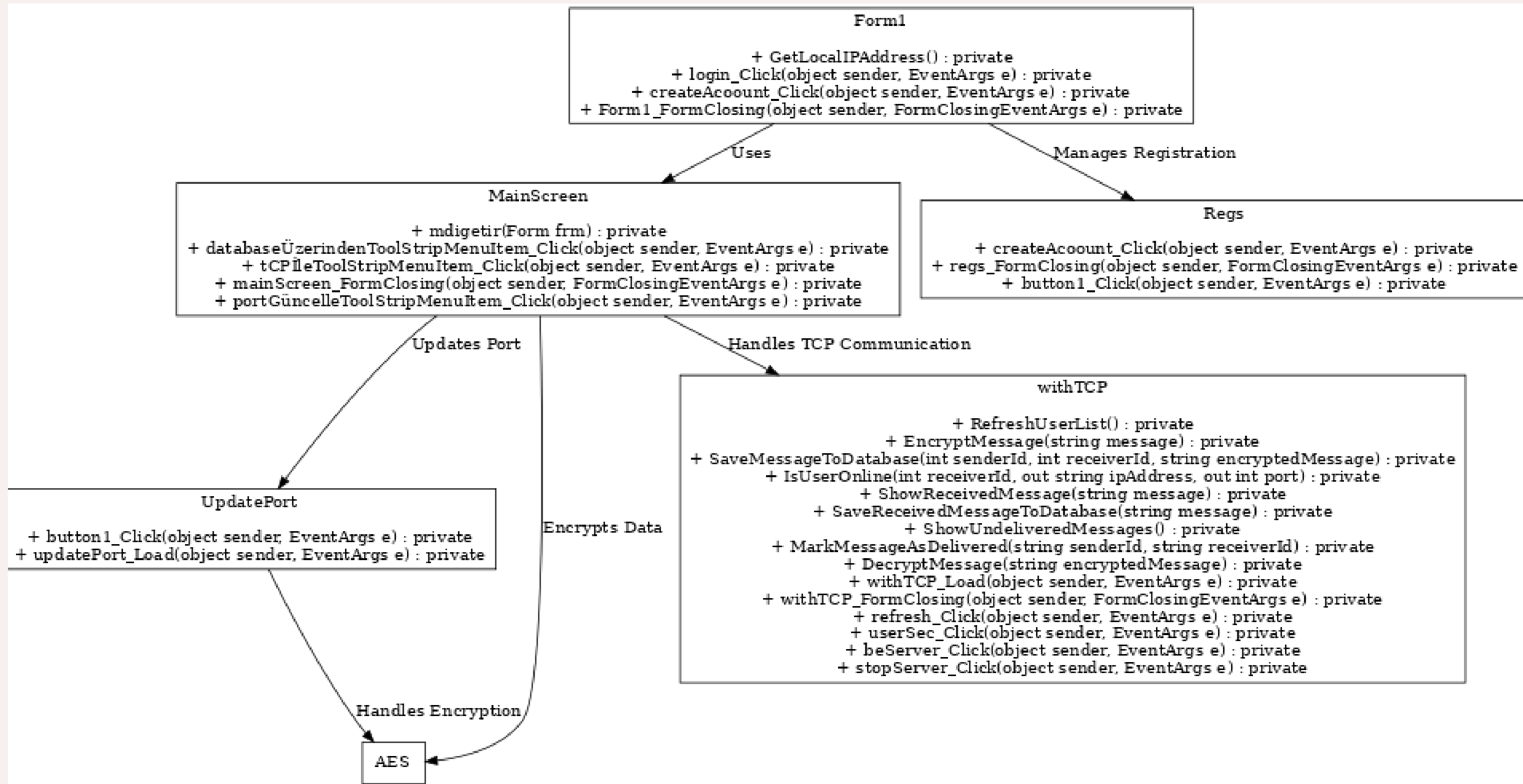
Port güncelleme kısmı ile kullanıcı kendi dinlemek istediği portu 100 100 değiştirebiliyor.



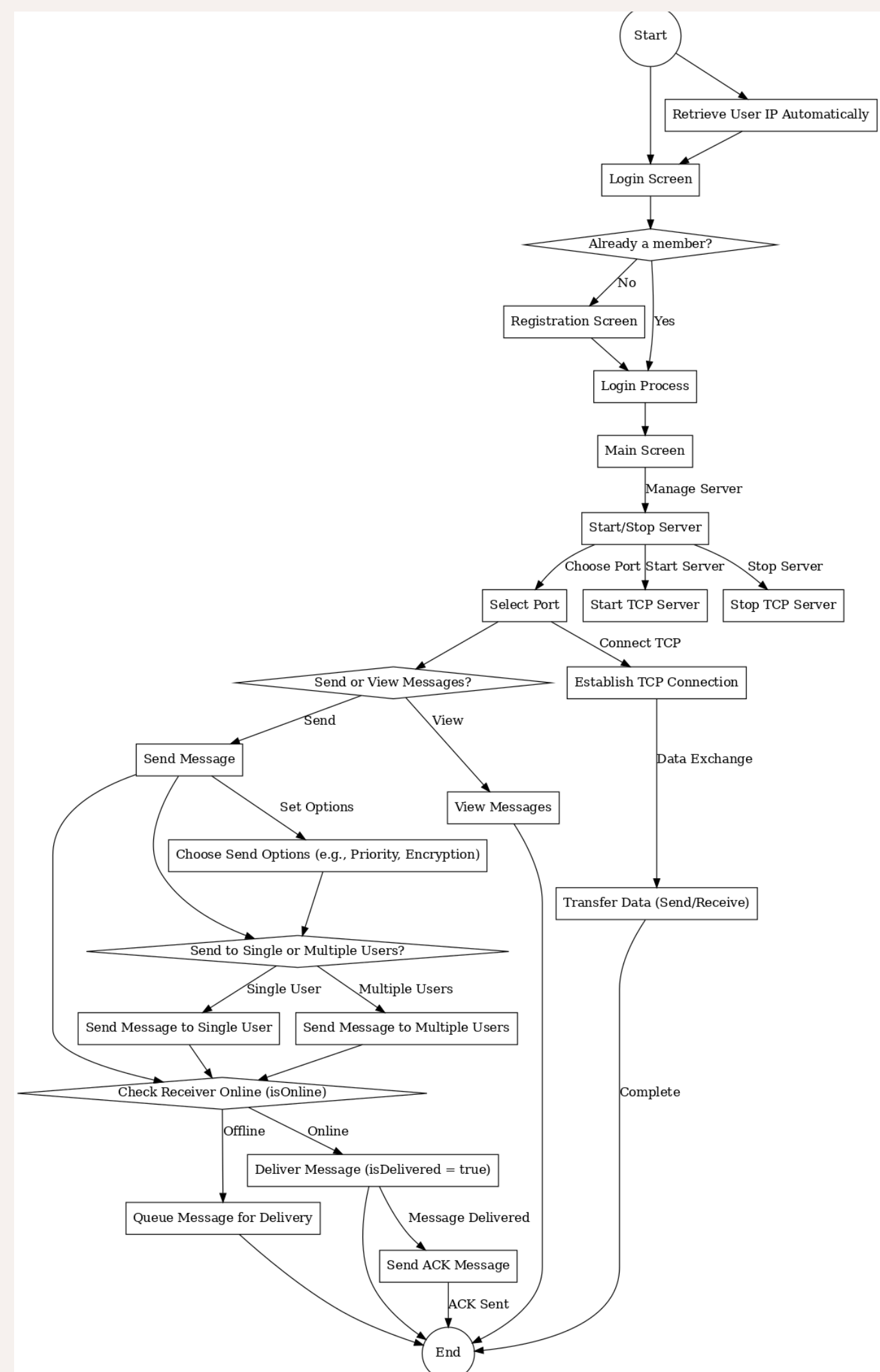
The screenshot shows a web application window with the title 'mainScreen'. It features a tabbed interface with two tabs: 'TCP ile' and 'Port Güncelle'. The 'Port Güncelle' tab is currently selected. Below the tabs, there is a text input field labeled 'PORTUNUZ:' which contains the value '5000'. To the right of the input field is a small vertical spinner control. Below the input field is a button labeled 'update'.

19

UML



20 FlowChart



Dinlediğiniz için teşekkürler!

Öğrenciler:

B211210079 - MUSTAFA YAKIN

G211210086 - MERT ÇALIŞKAN

github:

github.com/mustafayakin/AgProgramlama-Project