



# FATİH SULTAN MEHMET

## VAKIF ÜNİVERSİTESİ

### Öğrencinin;

**ADI:** Üveys

**SOYADI:** SARAÇ

**NO:** 1621221025

**BÖLÜM:** Bilgisayar Mühendisliği

### Projenin;

**KONUSU:** Çoklu client destekleyen chat uygulaması

### Dersin;

**ADI:** Bilgisayar Ağları

**EĞİTMEN:** Ali Yılmaz ÇAMURCU, Samet KAYA

## TCP :

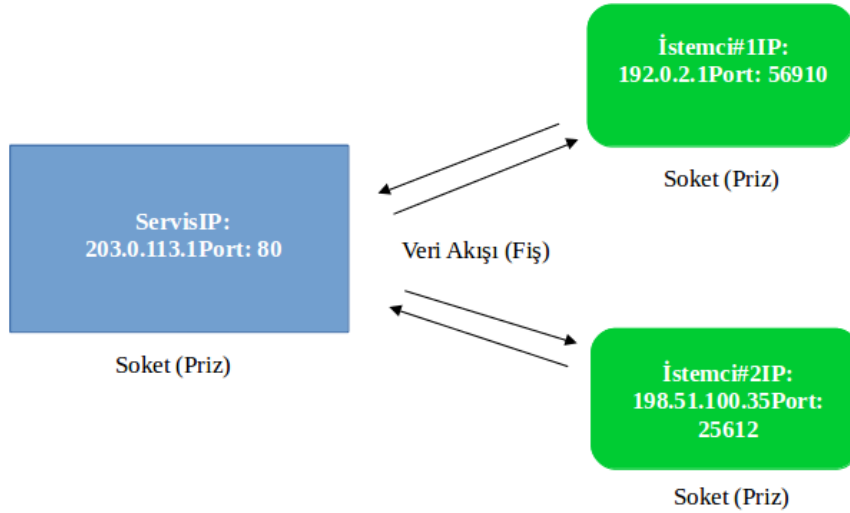
**TCP (Transmission Control Protocol)**, TCP/IP protokol takımının aktarım katmanı protokollerinden birisidir.

Gelişmiş bilgisayar ağlarında paket anahtarlama bilgisayar iletişimde kayıpsız veri gönderimi sağlayabilmek için TCP protokolü yazılmıştır. HTTP, HTTPS, POP3, SSH, SMTP, Telnet ve FTP gibi internet'in kullanıcı açısından en popüler protokollerinin veri iletimi TCP vasıtasıyla yapılır.

## Socket programlama :

Soketler, süreçler arası iletişim için kullanılır. **Interprocess iletişimi** istemci-sunucu modeline dayanır. Bu durumda, istemci-sunucu birbiriyle etkileşime giren uygulamalardır. İstemci ve sunucu arasındaki etkileşim bir bağlantı gereklidir. Soket programlaması, etkileşime girecek uygulamalar arasındaki bağlantıyı kurmaktan sorumludur. Kısacası ağ iletişimini temin eden bir uyarlamadır.

Çevrim içi kullanıcıların bulunduğu bir platform oluşturduunuz ve kullanıcılar arası anlık mesajlaşma özelliği getireceğinizi varsayalım, bir kullanıcıdan diğer bir kullanıcıya mesaj gönderildiğinde sunucu tarafında bir değişiklik olur ve sunucu bunu istemciye bildirir. Bu değişikliği algılayabilmek için polling, long polling ya da websocket gibi yapılar kullanılır. **Polling** belli aralıklarla sunucuya istek yapar, sunucu her isteğe bir cevap oluşturup gönderir ve bu çok fazla trafik yaratarak, bu da anlık konseptinin dışına çıkar. Long polling ise sunucuya isteği atar fakat sunucu tarafından cevap dönmesi için yeni bir istek yapılmasını bekler. Websocketler ise **HTTP protokolüne** uygun olmayan eş zamanlı web uygulamalarındaki karmaşık yapının basitleştirilmesini sağlar. Websocket'ler polling'e göre daha az band genişliğine ihtiyaç duyar. Websocket ile kalıcı bir bağlantıyla oluşturduğumuz port üzerinden kullanıcılar arası iletişim portunu dinleyebilir ve anlık olarak kullanıcılar arası iletişimi ucuz ve hızlı yoldan halletmiş olursunuz.



## Multi-Threading :

Çok akışlı ya da birden fazla iş parçacığı ile programlama denilebilir. Yani bir programın, uygulamanın aynı anda birden fazla iş yapabilme yeteneğidir. Çok akışlı programlama javanın bir kütüphanesi olmayıp yazılım dilinin temel özelliklerinden birisidir. **Thread**, iş parçacığı uygulama içindeki yapılması gereken bir işlemin baştan sona eksiksiz yerine getirilmesidir.

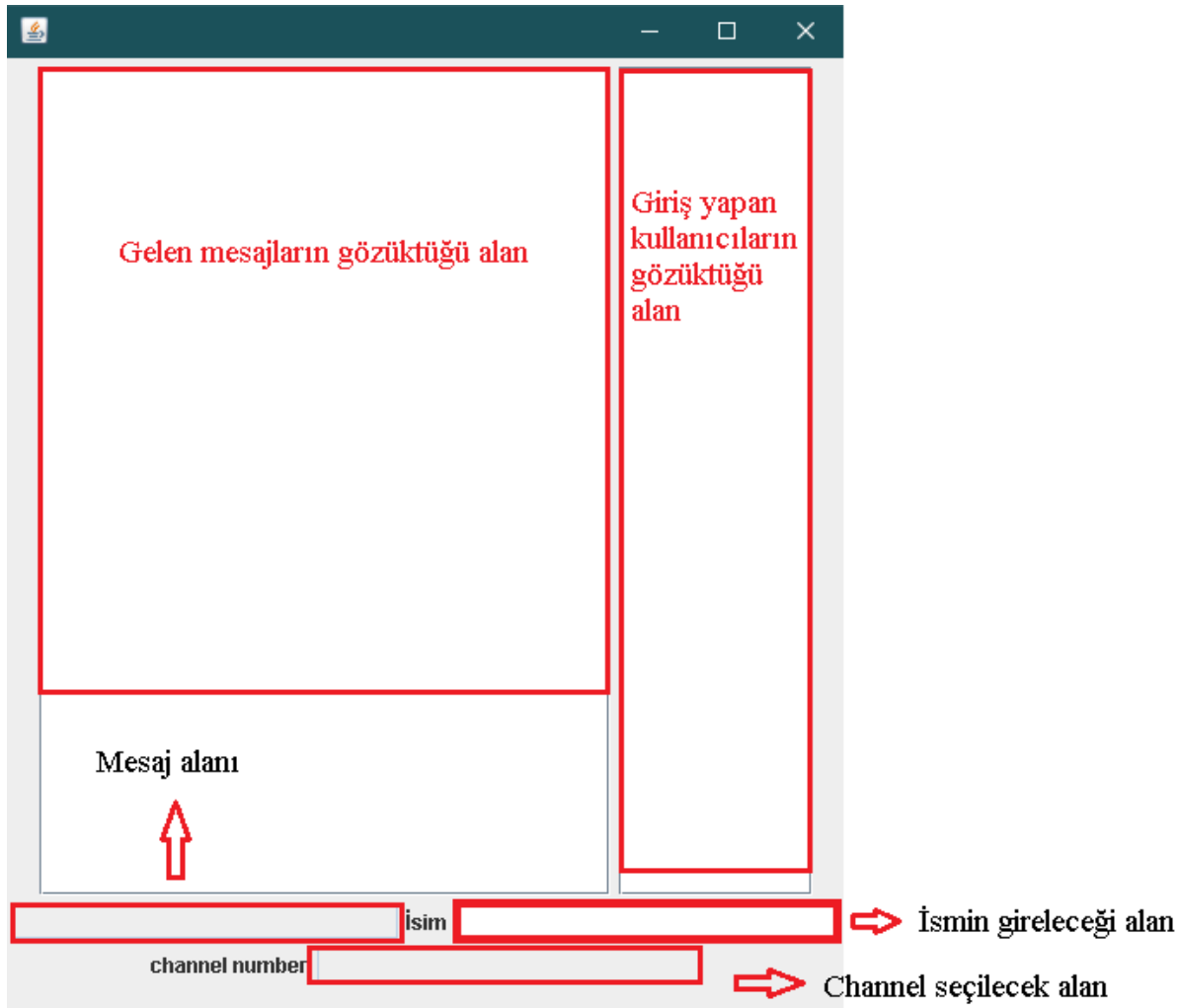
**Thread sınıfının bazı metotları:**

- **run metodu:** Çok akışlılığı sağlamaktadır. Runnable arabirimini uygulayan sınıflar bu metodu override etmektedir.
- **start metodu:** Ana akıştan-işten farklı bir akış-iş parçacığı çağırır ve çağırılan iş parçacığını başlatır.
- **sleep(long tipinde değer):** Parametrede verilen milisaniye boyutundaki zaman aralığında app'nin duraksamasını sağlar.

## Projede yapılanlar :

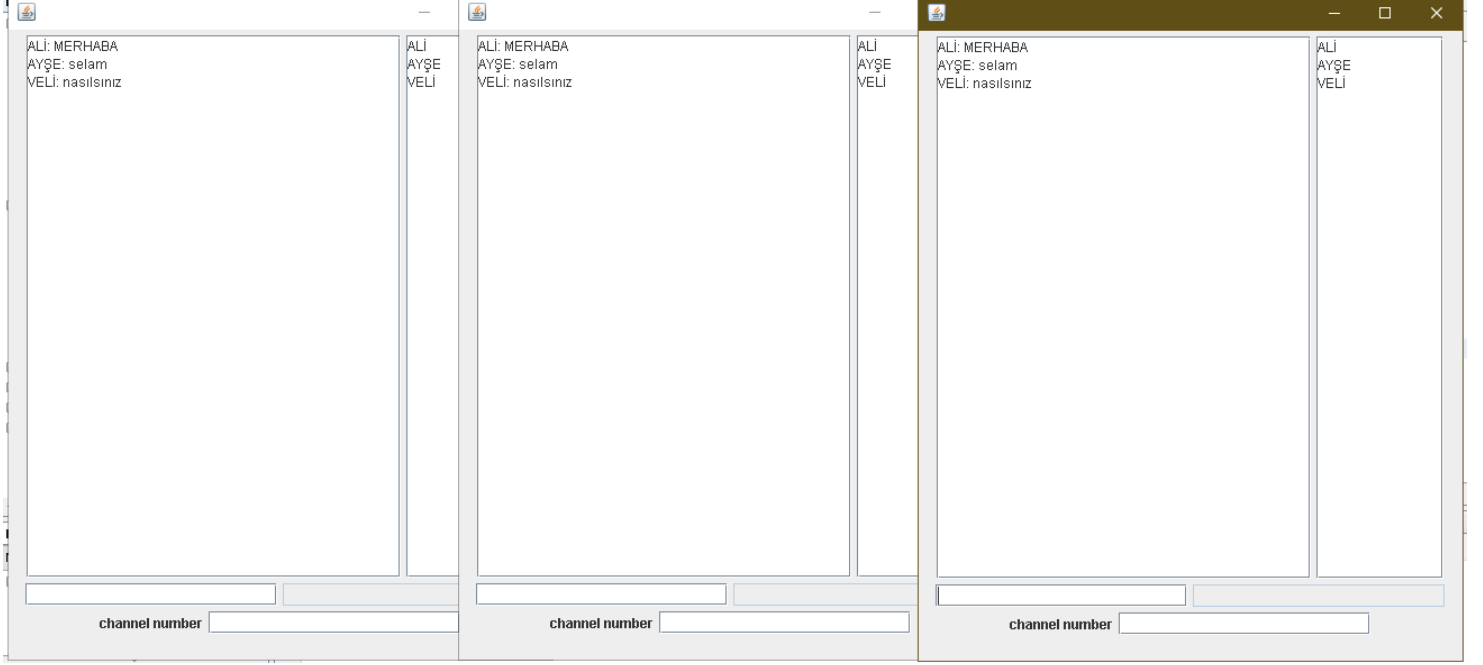
- Clientlar servera bir isimle bağlanır.
- Severa bağlanan client servera o an bağlı olan tüm clientları görür.
- Clientlar özel chat odası oluşturabilir(özel bir isimle)
- Clientlar kendi aralarında birebir(özel mesaj) mesajlaşabilir.

## Projeye ait ekran görüntüleri ve çalışma şekli :



İlk önce isim alanı açık olarak gelir. Channel ve mesaj alanları editable değildir. İsim girildikten sonra Channel ve mesaj alanı açılır.

Channel deęiřtirilmezse baęlanan herkesin birbiriyle mesajlařtıęı alanda mesajlařılır. 2 kullanıcı aynı channel'ı seęerse birlikte burada mesajlařırlar. Channel numbera 2 kullanıcı kriptolu bir metin girip burada birlikte özel olarak mesajlařabilir.



Resimde grldę zere 3 kullanıcı baęlanıp birbiri ile mesajlařabiliyor.