**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

AĞ PROGRAMLAMA

ÖDEV RAPORU

**REAL TİME CHAT APP**

**G191210305 - Adem YILMAZ –** [**adem.yilmaz10@ogr.sakarya.edu.tr**](mailto:adem.yilmaz10@ogr.sakarya.edu.tr)

**B201210351 - Eray BAYSAL-eray.baysal@ogr.sakarya.edu.tr**

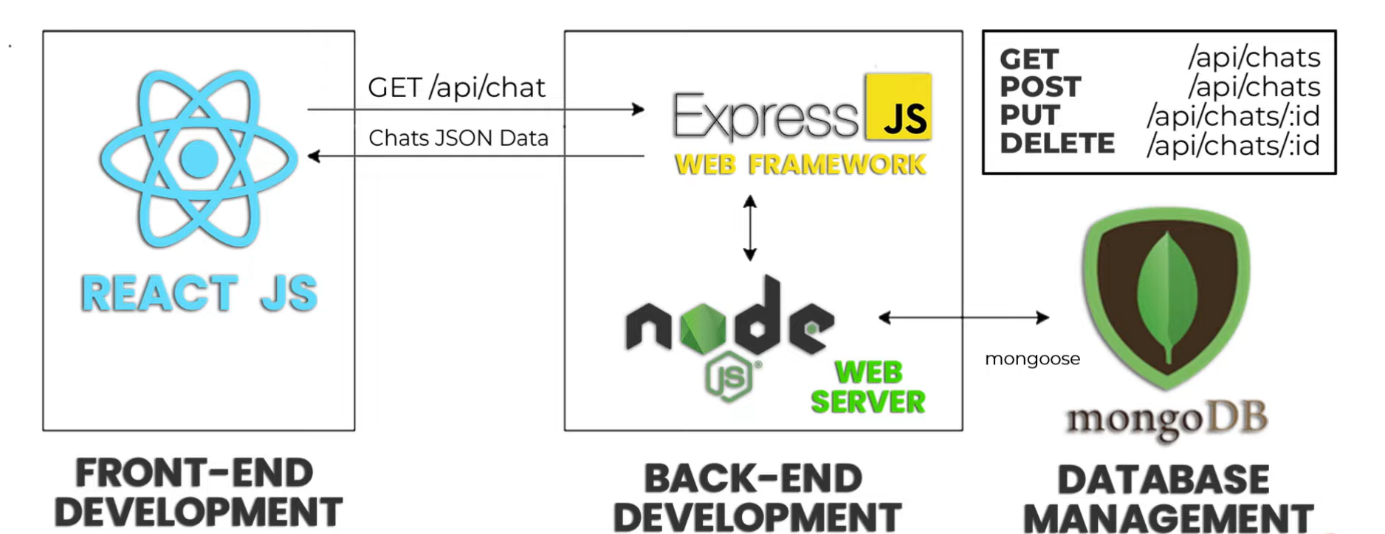
**SAKARYA**

**Nisan, 2023**

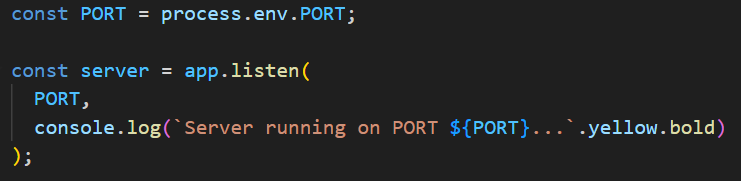
Ağ Programlama Dersi

**PROJE AÇIKLAMA**

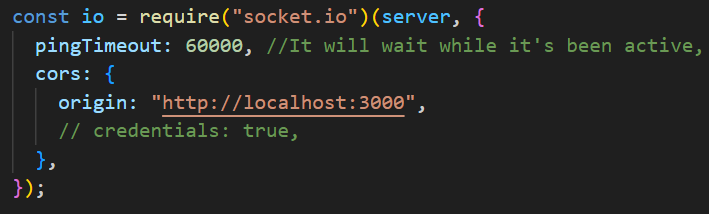
Gerçekleştirmiş olduğumuz ödev de soket programlama mimarisini kullanarak bir gerçek zamanlı mesajlaşma uygulaması yaptık. Bu projeyi bir web uygulaması şeklinde backend tarafında node.js, frontend tarafında react.js, veritabanı olarak mongodb ve node.js de bazı işlerimizi kolaylaştırmak için Express.js framework kullanarak gerçekleştirdik. Kullanıcılar günümüzde bazı mobil uygulamalarda olduğu gibi gerçek zamanlı olarak birbirleriyle haberleşebilmektedir. Kullanıcılar sisteme kayıt olarak üye olabilmektedir. Herhangi kayıtlı bir kullanıcı sisteme giriş yaptığında o sistemde kayıtlı kullanıcılar ile mesajlaşabilmektedir. Bir kullanıcı online olmadığında bile ona gelen mesajı online olunca görebilmektedir. Aynı zamanda grup oluşturulup mesajlaşma yapılabilmektedir. Gerçekleştirmiş olduğumuz uygulamadaki bazı özellikler günümüzde sık kullanılan uygulamalara benzetilmeye çalışılmıştır. Uygulamada soket programlama mimarisi kullanmak için bir node.js kütüphanesi olan socket.io kullanılmıştır. Frontend kısmında ise uygulamanın arayüzünü güzelleştirmek için çeşitli kütüphaneler(chakra-ui vs.) kullanılmıştır. Backend de ise mongodb veritabanına erişim sağlamak ve CRUD işlemleri yapmak için mongoose kütüphanesi kullanılmıştır. Sunucu backend 5000 nolu port da çalışacaktır. Fakat burada sorun frontend port numarası 3000. Biz bu port numarası ile frontend den backend apisine erişemeyiz. Dolayısıyla arada bir Proxy kullanmamız gerekmektedir. Dolayısıyla frontend de packege.json dosyasının içine Proxy olarak 127.0.0.1:5000 eklemeliyiz.



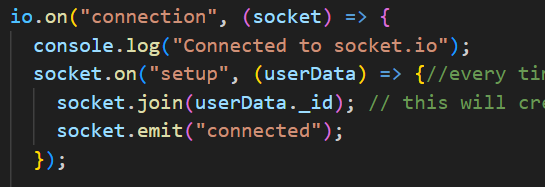
Backend klasörü içerisinde server.js dosyasında 5000 nolu portdan haberleşme yapacağımızdan bunu soket yapısında kullanmak için server değişkenine atıyoruz.



İo isimli değişkende require ile socket.io kütüphanesini çağırıp tanımlamış olduğumuz server bilgisini giriyoruz ve sonrasında ise pingTimeout yani sunucunun istemciyi aktif iken kaç saniye bekleyeceğini belirtiyoruz. Origin ile de sunucuya istemcinin frontend tarafında soket adresini bildiriyoruz.



Sonrasında ise io.on ile connection isminde soket tanımlıyoruz ve sunucu aktif hale getiriyoruz. Herhangi bir kullanıcı kendi hesabına giriş yaptığında ekrana Connected to socket.io yazdırılacaktır. Setup isminde soket oluşturulduktan sonra socket.join ile her kullanıcıya kendine özel ayrı bir soket tanımlaması yapılacaktır. Socket.emit ile de sunucudan istemciye connected mesajı gönderilecektir.

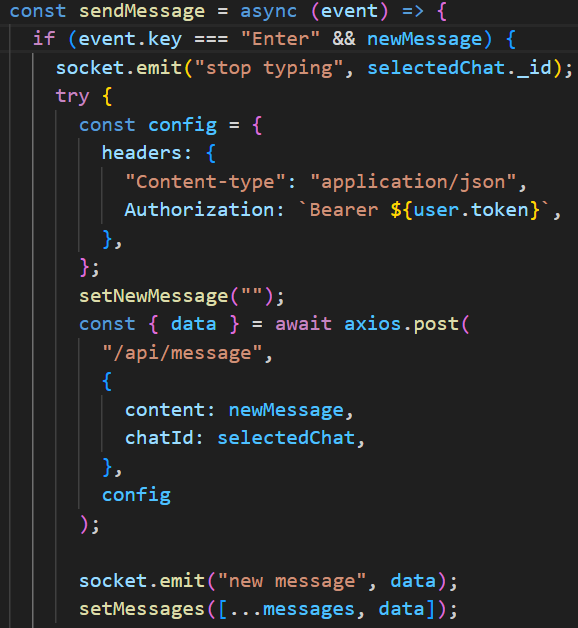


Frontend tarafında ise React.js ile çalıştığımızdan her bir değişiklik algılandığında useEffect ile gereken işlemler yapılacaktır. Burada sunucu ile bağlantı kurulması için soket tanımlaması yapılır. Endpoint sunucunun soket adresidir yani localhost:5000. Burada frontend klasörü içerisinde singlechat.js içerisinde her kullanıcının mesajının olduğu bölgede kullanıcı herhangi bir kullanıcı ile olan mesaj alanına girdiğinde socket.emit ile sunucuya setup mesajını ve aynı zamanda o kullanıcının bilgilerini gönderecektir. Sunucu tarafında ise ilgili kullanıcının id si ile o kullanıcıya ait bir soket tanımlanacaktır. Ve artık kullanıcı kendisine ayrılmış soket ile diğer kullanıcılarla mesajlaşabilecektir. Sunucu tarafında gereken işlem yapıldığında istemci tarafına connected mesajı gönderilecek ve artık istemci ve sunucu arasında bağlantı kurulacaktır. Typing alanları ise kullanıcılar chat de mesaj yazdıklarında karşı tarafta yazıyor simgesinin gözükmesi amacıyla oluşturulmuştur.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

İstemci tarafında yani frontend de kullanıcı herhangi bir kullanıcı veya grup chat içerisinde mesaj göndereceği zaman mesajı yazıp enter tuşuna bastığı zaman sunucuya socket.emit ile new message isimde mesaj ve aynı zamanda mesaj bilgilerinin olduğu data yı gönderecektir.



Sunucu tarafında socket.on ile “new message” beklenmektedir. Data bilgisi geldiğinde data içerisinden mesaj bilgisinin tutulduğu chat bilgisi alınır. Mesajı gönderen kişi haricinde mesaj gönderenin user id si ile chat içerisindeki tüm kullanıcılara gönderilir. Bu chat bir grup ise tüm gruba , iki kişi arasında bir chat ise diğer kişiye mesaj iletilir.

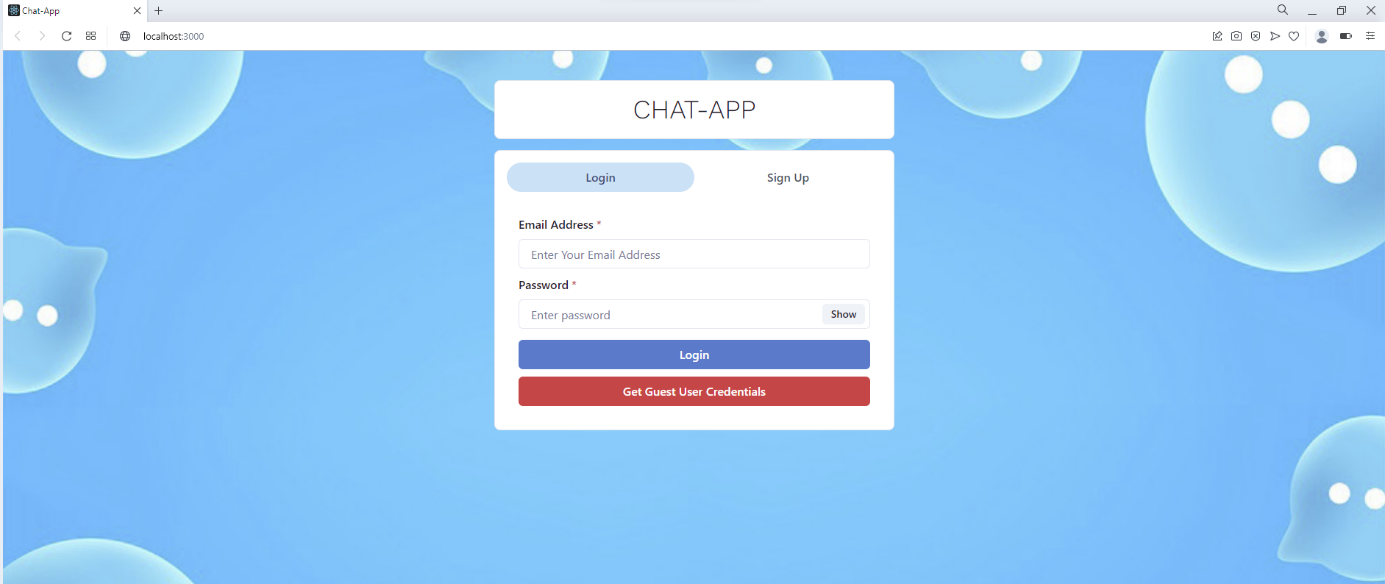


Sunucudan ilgili kullanıcıdan gönderilen message recieved bilgisi istemci tarafında socket.on ile beklenmektedir. Bu bilgi geldiğinde öncelikle mesajın kullanıcının diğer chatlerinde gözükmemesi sadece ilgili chat de gözükmesi için gerekli kontroller yapılır. Sonrasında ise notification listesinde gelen mesaj yoksa notification listesine eklenir ve bildirimler güncellenir. Aynı zamanda tekrardan chat kutusu yenilenir. Diğer durumda son alınan mesaj setMessages ile kullanıcıya ait mesajlar listesinin sonuna eklenir ki en son o mesaj gözüksün.

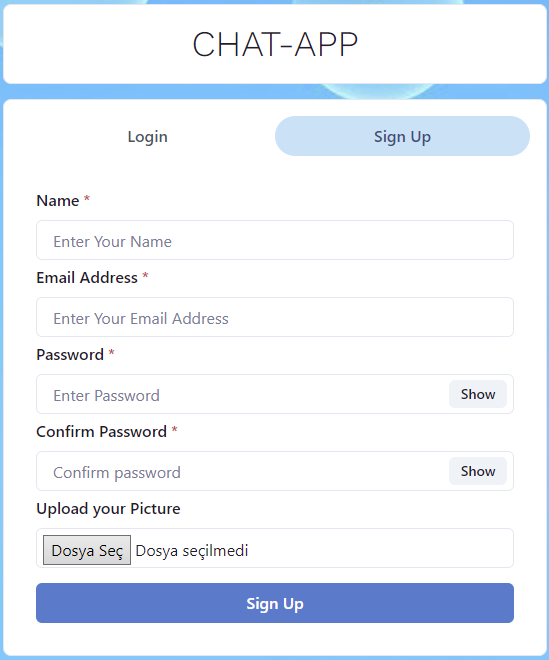


**Uygulama Ekran Görüntüleri**

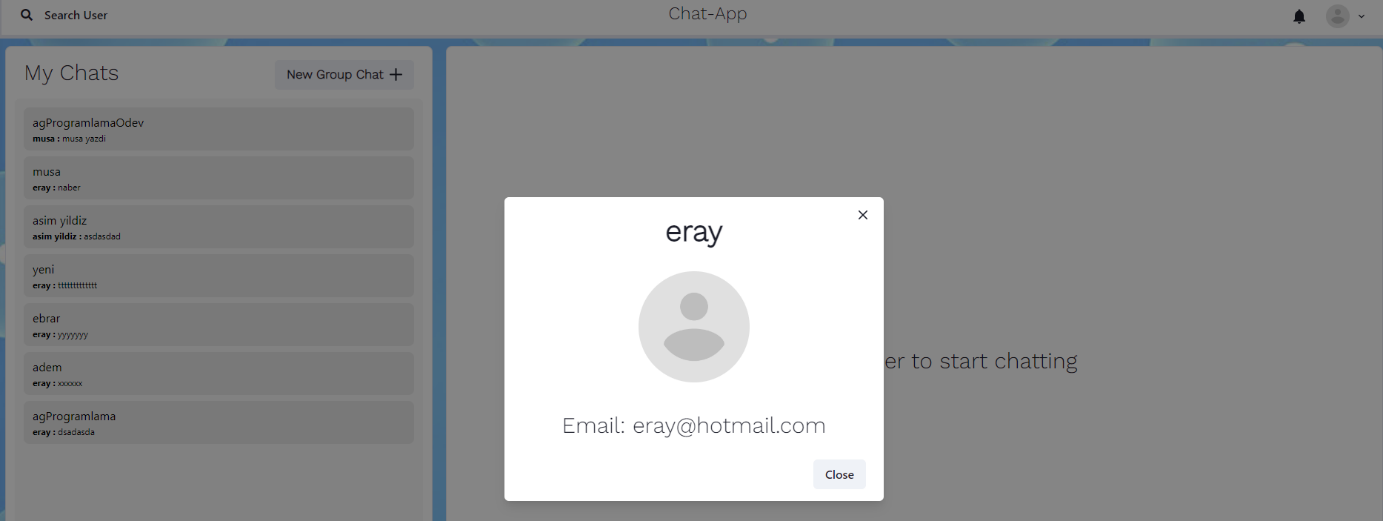
Uygulama ilk açıldığında karşımıza çıkacak olan sayfadır. Burada kullanıcılar sisteme üye ise giriş yapabilirler.



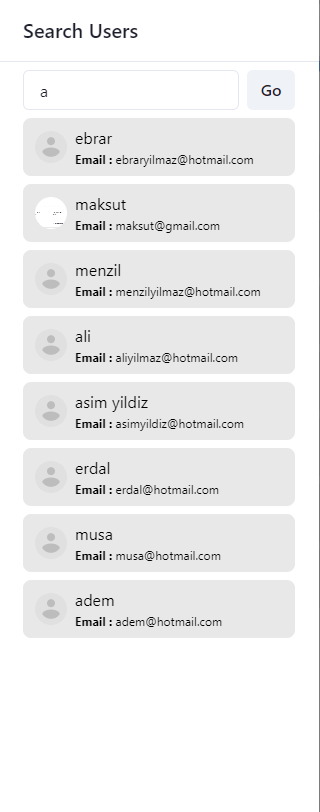
Kullanıcılar sisteme üye değil ise sign-up menüsünden üye olabilirler ve sisteme giriş yapabilirler.



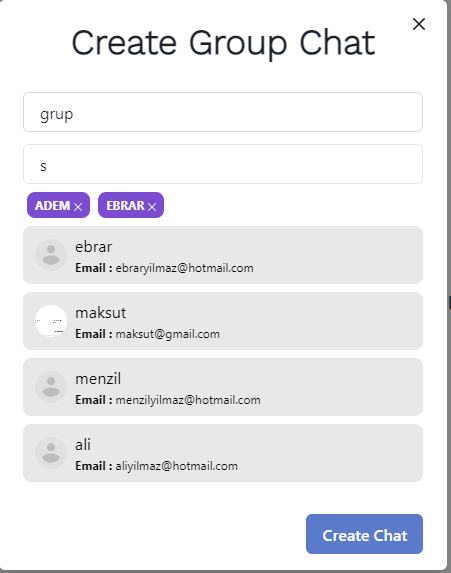
Eray isimli kullanıcının chat sayfası aşağıdaki resimde gözükmektedir.



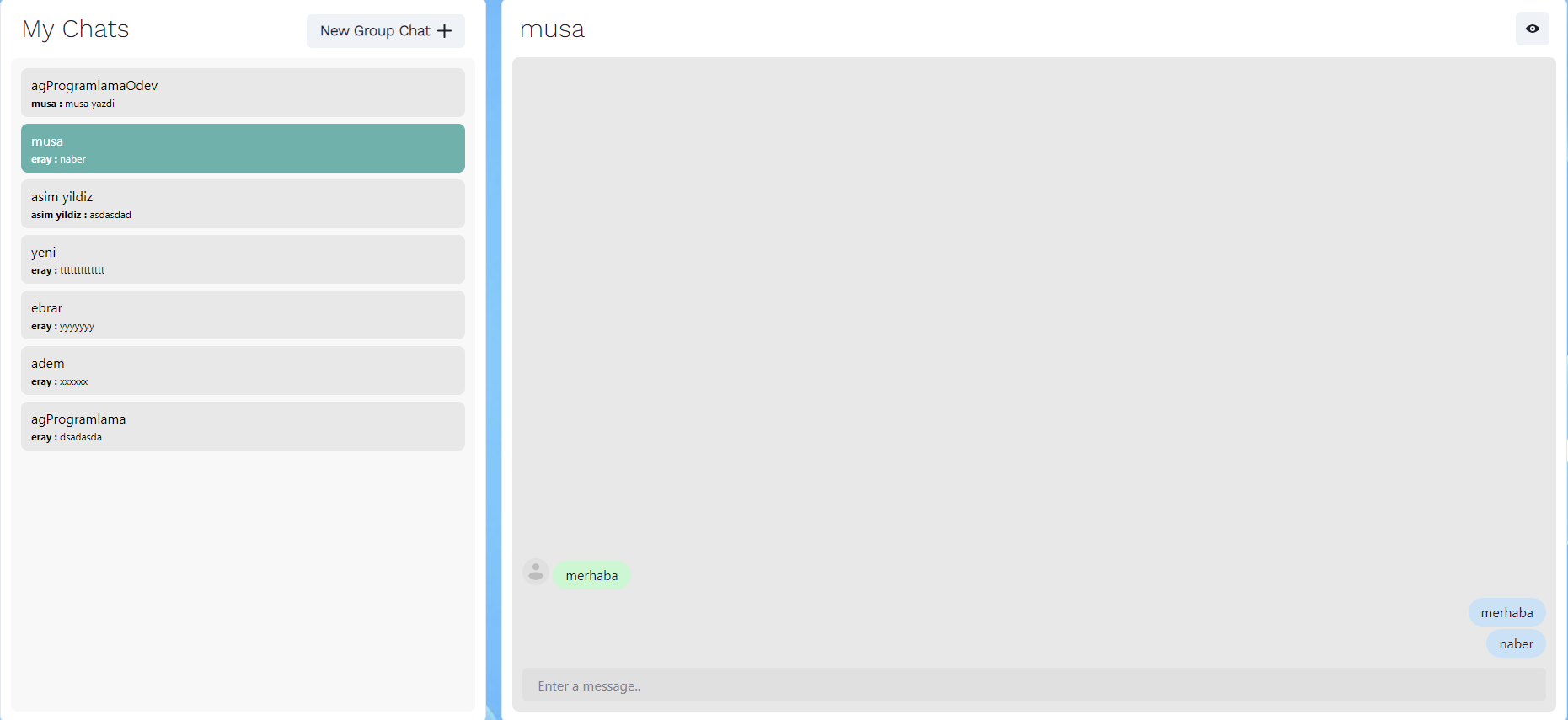
Search users kısmında kullanıcılar sisteme kayıtlı olan diğer kullanıcıları bulup onlarla mesajlaşabilmektedir.



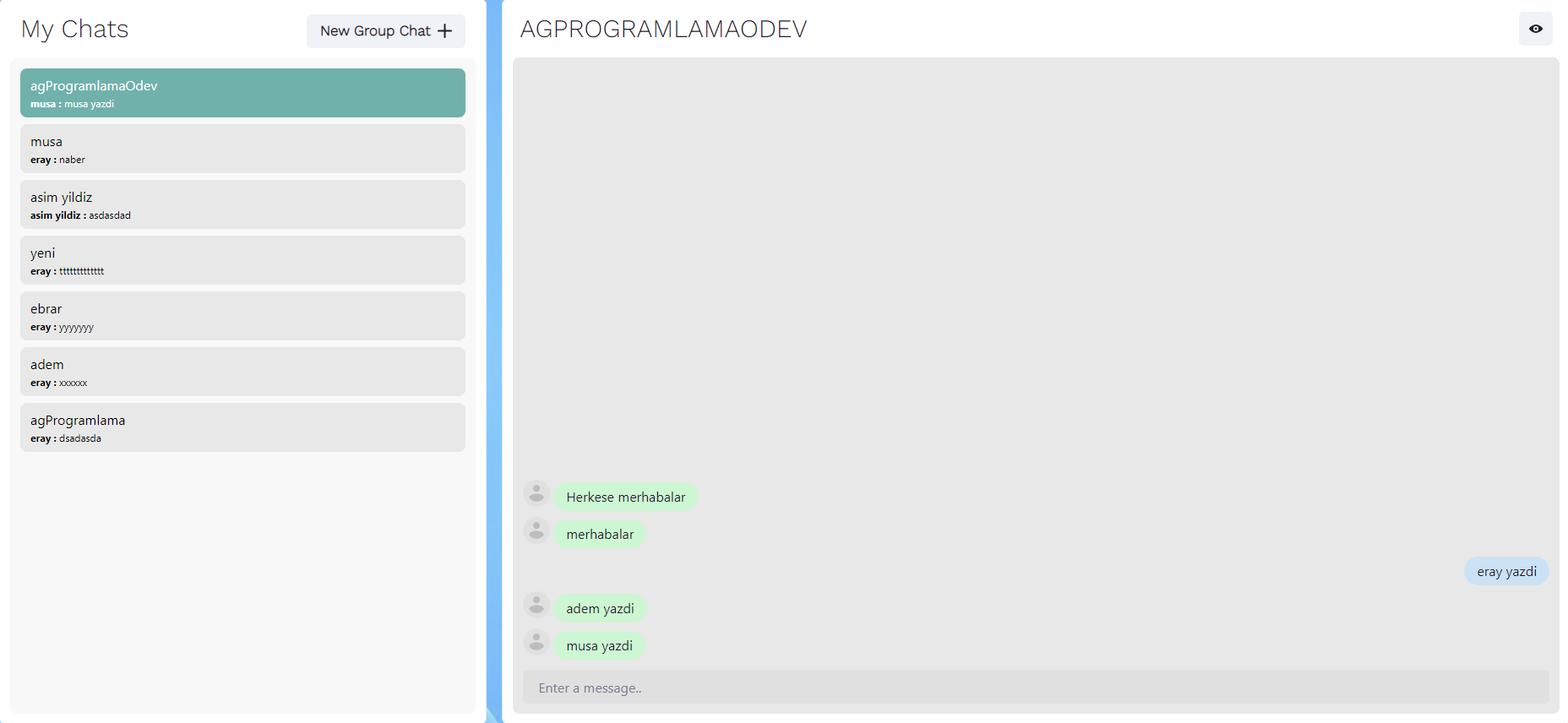
Create Group Chat kısmında ise kullanıcılar sisteme kayıtlı kullanıcılar ile grup oluşturup grup içerisinde mesajlaşabilmektedirler.



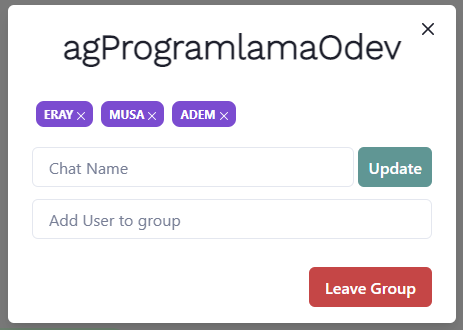
Musa isimli kullanıcının eray isimli kullanıcı ile olan mesajları aşağıdaki gibi gözükmektedir.



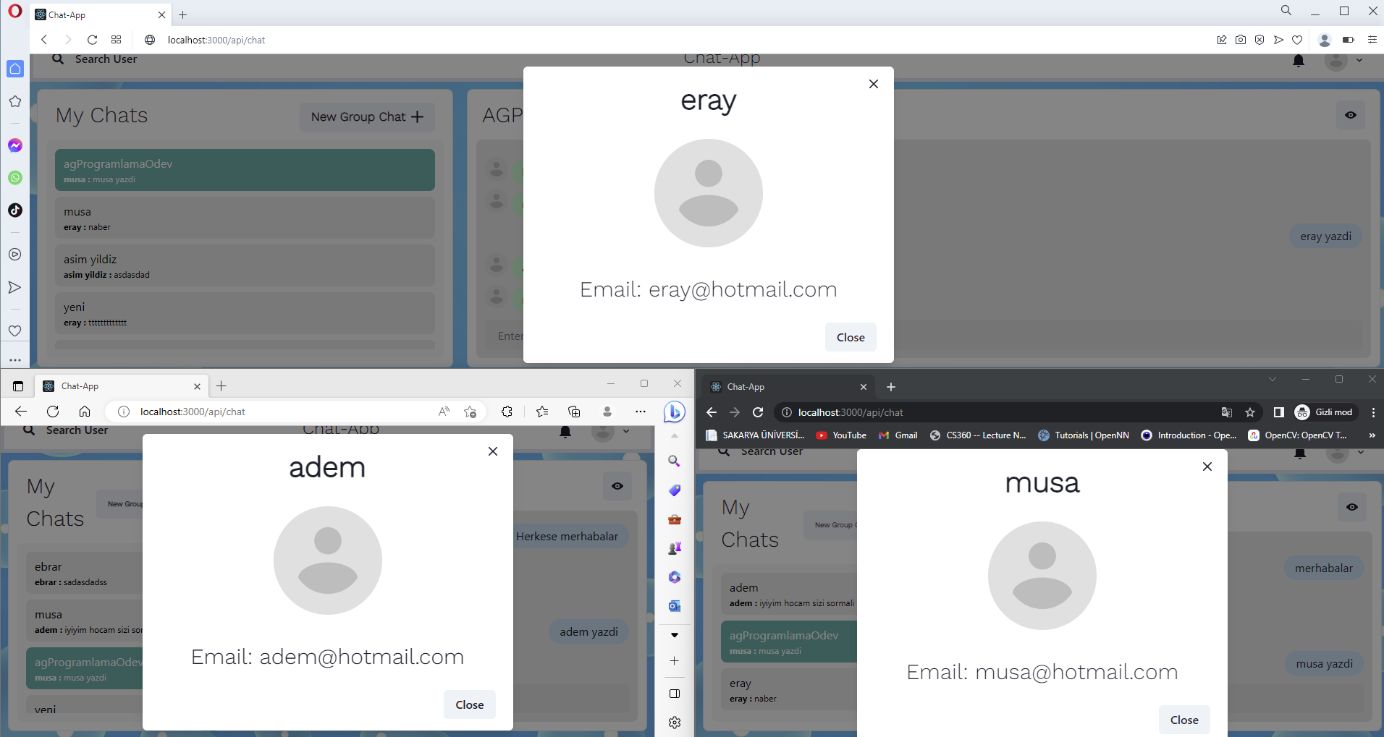
Grup üyeleri Eray, Musa, Adem olan AGPROGRAMLAMAODEV isimli grubun mesaj kutusu aşağıdaki gibi gözükmektedir.



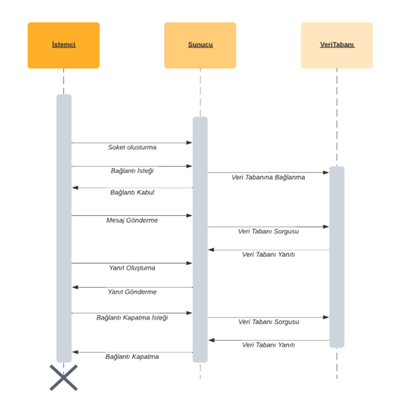
AgProgramlamaOdev isimli grubun üyeleri ve bu grupla yapılabilecek değişiklikler aşağıdaki gibi gözükmektedir.



Eray, Adem ve Musa isimli kullanıcıların chat sayfaları her biri ayrı bir tarayıcıda açılmak üzere görüntülenmiştir.



**Sequence Diyagram**

****