



B&T Buluş ve Tanış

Nihai Proje Gereksinim Dökümantasyon Raporu

Grup12

VERSION: 1.0.1

05/06/2023

0.0	İçindekiler	2
1.0	Giriş.....	3
2.0	Proje Kapsam ve Hedefleri.....	3
3.0	İşlevsel Gereksinimler	3
3.1	Kullanıcı Adı ve Kimlik Doğrulama	3
3.2	İletişim Yönetimi	3
3.3	Mesaj Değişimi/Mesajlaşma	3
4.0	İşlevsel Olmayan Gereksinimler	4
4.1	Performans.....	4
4.2	Güvenlik	3
4.3	Kullanılabilirlik	3
4.4	Ölçeklenebilirlik	3
5.0	Sistem Mimarisi	Error! Bookmark not defined.
5.1	Front-End Mimarisi	3
5.2	Back-End Mimarisi	5
6.0	Kullanıcı Arayüz Tasarımı.....	6
6.1	Giriş/Login Sayfası.....	6
6.2	Kayıt/Register Sayfası	7
6.3	Kullanıcı/Userpage Sayfası	8
6.4	Profil Sayfası	9
7.0	Veri Yönetimi	10
7.1	Veri Tabanı Mimarisi	10
8.0	Proje Zaman Çizelgesi ve Kaynakları	11

1.0 GİRİŞ

Bu belgenin amacı, B&T Buluş ve Tanış projesine kapsamlı bir genel bakış sağlamaktır. Kapsamı, hedefleri, işlevsel ve işlevsel olmayan gereksinimleri, sistem mimarisini, kullanıcı arabirimi tasarımını, veri yönetimini, test ve kalite güvencesini, dağıtım ve bakımı, proje zaman çizelgesini ve kaynaklarını, ayrıca varsayımları ve kısıtlamaları ana hatlarıyla belirtir.

2.0 KAPSAM VE HEDEFLER

B&T Buluş ve Tanış, kullanıcıların gerçek zamanlı olarak birbirleriyle iletişim kurmasını sağlayan bir çevrimiçi mesajlaşma web uygulamasıdır. Temel amaç, kesintisiz ve kullanıcı dostu bir mesajlaşma deneyimi sunmaktır. Hedef kitle, web uygulaması aracılığıyla etkileşimde bulunmak ve mesaj alışverişinde bulunmak isteyen bireyler ve gruplardır.

B&T Buluş ve Tanış projesi, hızlı, güvenli ve kullanıcı dostu bir mesajlaşma deneyimi sunan bir web tabanlı uygulamanın geliştirilmesini hedeflemektedir.

Projenin amaçları şunlardır:

- Gerçek zamanlı iletişim sağlayan bir mesajlaşma sistemi oluşturmak.
- Kullanıcıların kolayca mesaj gönderip alabileceği bir arayüz tasarlamak.
- Veri iletimi ve kullanıcı doğrulama mekanizmaları gibi güvenlik önlemlerini sağlamak.
- Uygulamanın yüksek performanslı, ölçeklenebilir ve istikrarlı bir şekilde çalışmasını sağlamak.

Bu şekilde, B&T Buluş ve Tanış, kullanıcıların güvenli ve etkileşimli bir şekilde iletişim kurabileceği bir platform olmayı hedeflemektedir.

3.0 İŞLEVSEL GEREKSİNİMLER

3.1 Kullanıcı Kaydı ve Kimlik Doğrulama

Kullanıcılar bir hesap oluşturabilir ve mesajlaşma özelliklerine erişmek için kimliklerini doğrulayabilir.

Input: Kullanıcı Adı , E-Mail , Şifre

Output: Kullanıcı hesabının oluşturulması ve kimlik doğrulanması

3.2 İletişim Yönetimi

Kullanıcılar , kişi listelerine ilgili kişiyi ekleyerek , kaldırarak veya düzenleyerek kişilerini yönetebilir.

Input: İletişim bilgileri , İletişim grupları

Output: Kişilerin eklenmesi veya kaldırılması , kişi gruplarının düzenlenmesi

3.3 Mesaj Değişimi/Mesajlaşma

Kullanıcılar gerçek zamanlı olarak mesaj gönderebilir, alabilir ve görüntüleyebilir.

Girişler: Mesaj içeriği , Alıcı(lar)

Çıkışlar: Gönderilen, alınan ve görüntülenen mesajlar.

4.0 İŞLEVSEL OLMAYAN GEREKSİNİMLER

4.1 Performans

Web uygulaması bir çok sayıda eşzamanlı kullanıcıyı desteklemeli ve mesajları neredeyse gerçek zamanlı olarak iletebilmelidir.

4.2 Güvenlik

Kullanıcı verileri, şifreleme protokolleri kullanılarak güvenli bir şekilde saklanmalı ve iletilmelidir.

4.3 Kullanılabilirlik

Kullanıcı arayüzü, sezgisel ve gezinmesi kolay olmalı ve kesintisiz bir mesajlaşma deneyimi sağlamalıdır.

4.4 Ölçeklenebilirlik

Sistem, gelecekteki büyümeyi ve artan kullanıcı taleplerini karşılamak için ölçeklenebilir olmalıdır.

5.0 SİSTEM MİMARİSİ

5.1 Front-End Mimarisi

- `@popperjs/core`
"2.11.6" - Bu kütüphane, modern web uygulamalarında pop-up pencereleri, açılır menüleri ve bilgi kutularını düzenlemek için kullanılır. Popper, referans elemanı ve hedef eleman arasındaki ilişkiyi hesaplamak ve yerleştirmek için kullanılır. Özellikle Vue, React veya Angular gibi popüler framework'lerle kullanılan bileşenlerin oluşturulmasında sıklıkla kullanılır.
- `axios`
"1.4.0"
Axios, HTTP istekleri göndermek ve almak için kullanılan bir JavaScript kütüphanesidir. Bu, modern web uygulamalarının çoğunda bir ihtiyaçtır, çünkü bir API'ye bağlanmak ve verileri almak veya göndermek sık sık yapılması gereken işlemlerdir. Axios, özellikle büyük veri dosyaları veya resimler gibi büyük dosyaları yükleme işlemleri için kullanışlıdır.
- `bootstrap`
"5.2.3"
Bootstrap, CSS, JavaScript ve HTML'den oluşan popüler bir frontend framework'üdür. Bootstrap, web uygulamalarının ve mobil uygulamaların hızlı ve kolay bir şekilde geliştirilmesine yardımcı olmak için tasarlanmıştır. Bootstrap, özellikle kullanıcı arayüzü (UI) geliştirmesi için yaygın olarak kullanılır.
- `core-js`
"3.8.3"
Core-js, modern JavaScript özelliklerini (ES6+ özellikleri) desteklemek için tasarlanmış bir JavaScript kütüphanesidir. Core-js, web tarayıcılarındaki eksiklikleri doldurarak, web uygulamalarının ve yazılım projelerinin daha modern ve güncel JavaScript özelliklerini kullanmasına olanak tanır.

- ``vue`: "^3.2.13"`
Vue, modern web uygulamaları için açık kaynaklı bir JavaScript framework'üdür. Vue, basit, hafif ve esnek bir yapıya sahiptir ve bileşen tabanlı bir yapı kullanır. Vue, özellikle SPA (tek sayfa uygulamaları) oluşturmak için kullanılır.
- ``vue-router`: "^4.1.6"`
Bu, VueJS uygulamalarında yönlendirmeyi yönetmek için kullanılan bir kütüphanedir. Vue Router, tek sayfa uygulamalarında (SPA) URL yönetimini kolaylaştıran birçok özellik sunar. Örneğin, URL parametreleri, yolları, vurguları, middleware'leri ve daha birçok özelliği destekler. Bu, VueJS projelerinde sayfa yönetimini daha kolay ve esnek hale getirir.
- ``vuex`: "^4.1.0"`
Vuex, VueJS projelerinde durum yönetimi için kullanılan bir kütüphanedir. Vuex, web uygulamalarında verilerin nasıl depolanacağı, yönetileceği ve paylaşılacağına ilişkin en iyi uygulama yöntemlerini sağlar. Vuex, özellikle büyük ve karmaşık uygulamalarda durum yönetimini kolaylaştırmak için kullanılır. Vuex, uygulama durumunu merkezi bir yerden yönetir ve bu sayede uygulama bileşenleri arasında daha iyi bir koordinasyon sağlar.

5.2 Back-End Mimarisi

- `bcrypt: "^5.1.0"`
bcrypt kütüphanesi, kullanıcı şifrelerini güvenli bir şekilde hashlemek için kullanılır. Bu, kullanıcıların şifrelerinin güvenli bir şekilde saklanmasını ve bir saldırganın şifreleri çözmesini önler. Bu nedenle, web uygulaması gibi güvenli bir proje geliştirirken bu kütüphane kullanılabilir.
- `cors: "^2.8.5"`
CORS (Cross-Origin Resource Sharing) kütüphanesi, farklı kaynaklardan (domain, protokol veya port) gelen HTTP isteklerinin güvenli bir şekilde işlenmesine yardımcı olur. Özellikle modern web uygulamaları, örneğin API'ler, özellikle farklı kaynaklardan istekleri kabul ederek çalışırlar. Bu kütüphane, bu tür isteklerin güvenli bir şekilde yönetilmesine yardımcı olur.
- `express: "^4.18.2"`
Express, Node.js tabanlı web uygulamaları oluşturmak için yaygın olarak kullanılan bir web uygulaması çerçevesidir. Bu kütüphane, HTTP isteklerini işlemek, yönlendirmek ve yanıtlamak için bir dizi kolay kullanımlı API sağlar. Express, web uygulaması geliştirirken temel bir kütüphanedir.
- `jsonwebtoken: "^8.5.1"`
Jsonwebtoken, JSON tabanlı kimlik doğrulama işlemlerinde kullanılır. Bu kütüphane, JWT (JSON Web Token) üretmek ve doğrulamak için bir dizi API sağlar. JWT'ler, bir kullanıcının kimliğini doğrulamak veya yetkilendirmek için sıklıkla kullanılır. Bu nedenle, güvenli web uygulamaları geliştirirken bu kütüphane kullanılabilir.

- mongoose: "^6.6.7"

Mongoose, MongoDB için bir ODM (Object Data Mapper) kütüphanesidir. Bu kütüphane, MongoDB veritabanında kolayca erişilebilir ve yönetilebilir veri modelleri oluşturmaya yardımcı olur. Mongoose, Node.js tabanlı uygulamalarla birlikte MongoDB kullanırken kullanılabilir.

- nodemailer: "^6.8.0"

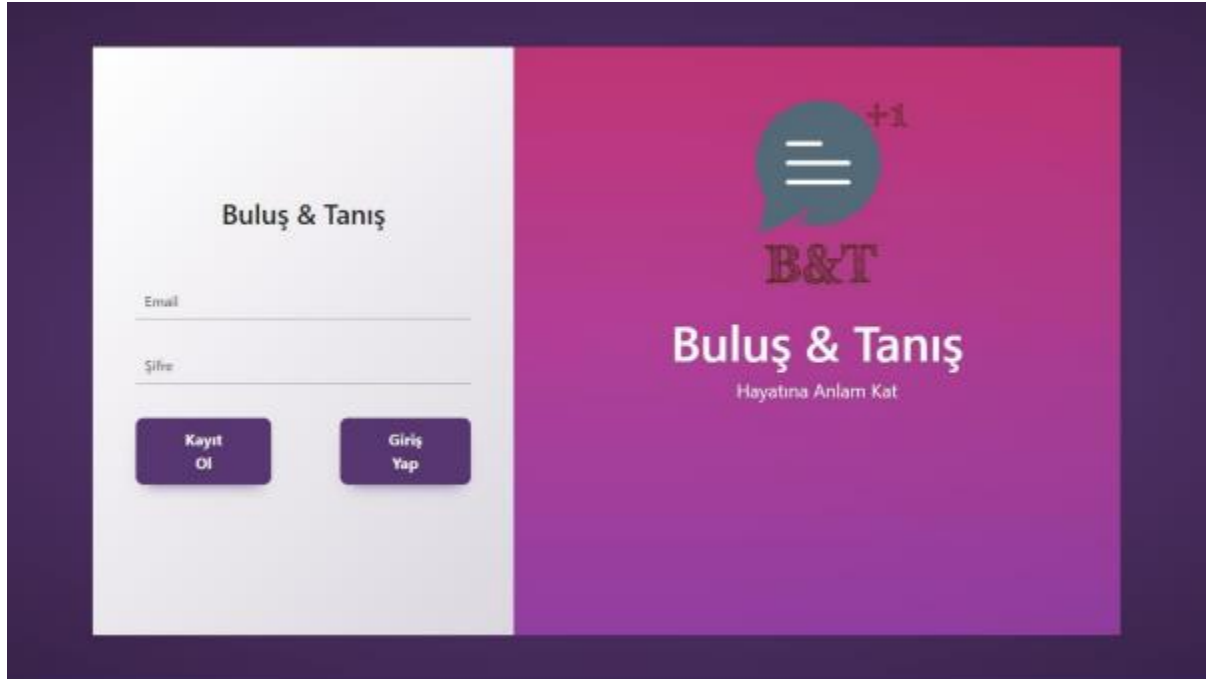
Nodemailer, Node.js tabanlı uygulamalar için bir e-posta gönderme kütüphanesidir. Bu kütüphane, SMTP protokolünü kullanarak e-posta göndermek için bir dizi kolay kullanımlı API sağlar. Bu, web uygulamasının e-posta doğrulaması vb. görevleri yerine getirmesine yardımcı olur.

- socket.io:

Socket.io, gerçek zamanlı, çift yönlü ve olay tabanlı iletişim sağlamak için kullanılan bir JavaScript kütüphanesidir. İki farklı tarayıcı veya tarayıcı-sunucu arasında anlık veri aktarımını mümkün kılar. Bu nedenle, uygulamaların canlı sohbet, anlık bildirimler, çevrimiçi oyunlar ve işbirlikçi uygulamalar gibi gerçek zamanlı iletişim özelliklerini uygulamak için yaygın olarak kullanılır. Socket.io, WebSockets protokolü üzerine inşa edilmiştir. Ancak, WebSockets protokolü yalnızca bazı modern tarayıcılar tarafından desteklenir. Socket.io, WebSockets kullanarak mümkünse gerçek zamanlı iletişim kurmaya çalışırken, desteklenmeyen tarayıcılarda da alternatif teknolojiler kullanarak geriye düşük seviyeli bir "polling" mekanizması sağlar. Bu sayede, Socket.io tüm tarayıcılarda çalışabilir.

6.0 KULLANICI ARAYÜZ TASARIMI

6.1 Giriş/Login Sayfası



- Kullanıcı, chat kısmına erişmek için öncelikle mail adresi ve parolasını girerek "giriş yap" butonuna tıklamalıdır. Ancak eğer girdiği bilgiler hatalı ise giriş işlemi başarısız olacaktır. Kullanıcı daha önce bir hesap oluşturmadıysa, kullanıcı "kayıt ol" butonuna tıklayarak kayıt olma menüsüne yönlendirilecektir.

6.2 Kayıt/Register Sayfası

Return home

Buluş & Tanış'a Katılın

İsim

Soy İsim

E-mail

Şifre

Ülke

Şehir

İlçe

Mahalle

Kayıt Ol

Buluş & Tanış
Hayatına Anlam Kat

Kullanıcı, isim, soyisim, e-posta, şifre, ülke, şehir, ilçe ve mahalle bilgilerini girerek, "Kayıt Ol" butonuna tıklamalıdır. Eğer girdiği bilgilerde herhangi bir hata varsa, sistem bir uyarı göstererek kullanıcının bilgileri düzeltmesini isteyecektir. Ancak, girilen bilgiler uygun ise, kayıt işlemi başarıyla tamamlanmış olacaktır. Ayrıca, "Return home" butonu ile kullanıcı "Giriş Sayfası"na geri dönebilir.

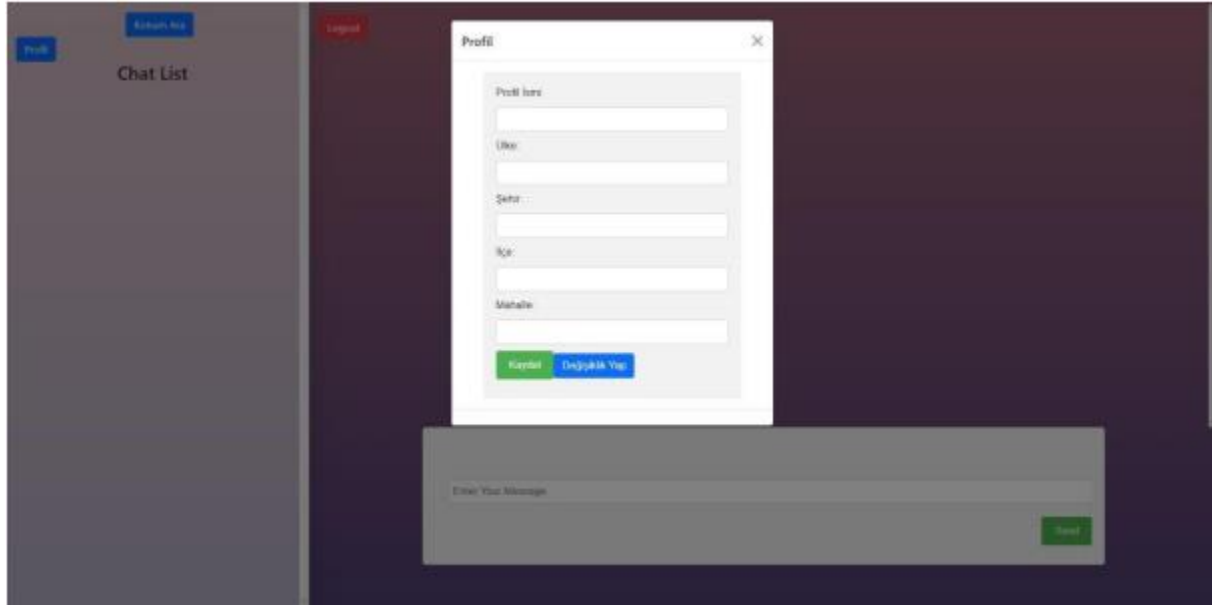
6.3 Kullanıcı/Userpage Sayfası

The screenshot displays a web interface for a chat application. On the left, there is a 'Chat List' section. In the top right corner, there are two buttons: 'Konum Ara' (Location Search) and 'Çıkış' (Logout). A modal window titled 'Konum Ara' is open in the center, featuring four input fields for location data: 'Ülke:' (Country), 'Şehir:' (City), 'İlçe:' (District), and 'Mahalle:' (Neighborhood). Below these fields is a green 'Ara' (Search) button. In the background, a message input area is visible with the placeholder text 'Enter Your Message' and a green 'Gönder' (Send) button.

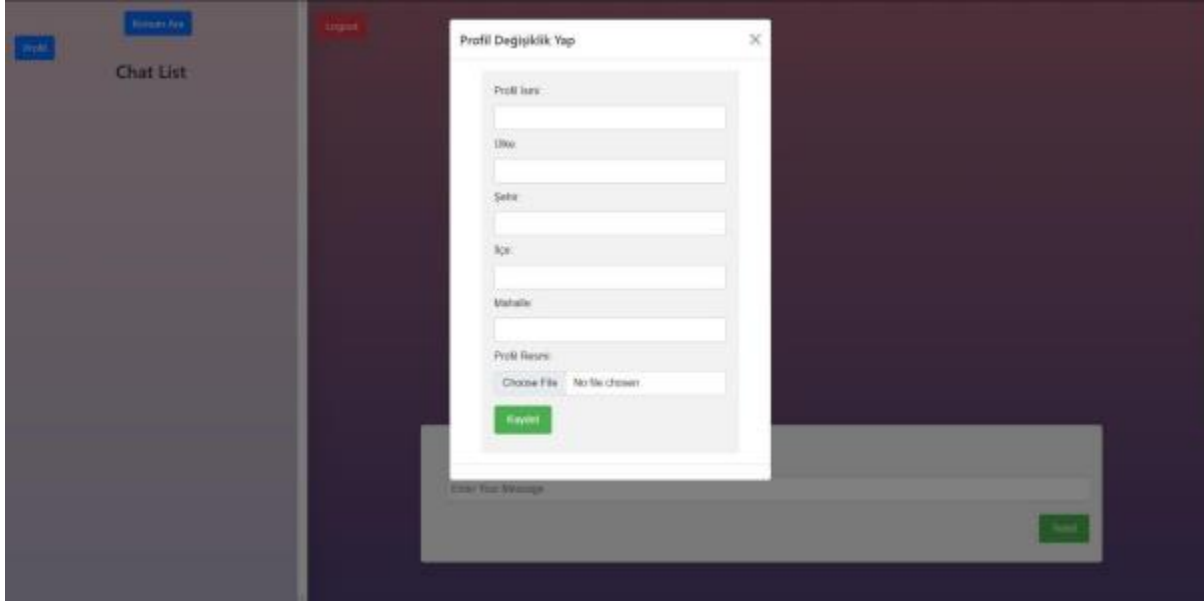
Ekranın sol üstüne eklenen "Konum ara" butonu ile ülke, şehir, ilçe, mahalle verilerini yazabileceğimiz forum bulunmaktadır. Formun altında "ara" butonu ile ilgili özellikler hakkında arama yapılabilmektedir.

Bu butona basınca girmiş olduğumuz adres filtresine sahip olan kullanıcıların profillerini görüyor ve onlara mesaj atabiliyoruz.

6.4 Profil/Profile Sayfası



Sitenin sol üst köşesinde bulunan profil butonuna tıklayarak kullanıcı kendi profil bilgilerini görüntüleyebilir. Profil kısmını açtığımızda Profil İsmi , Şehir , Ülke , İlçe ve Mahalle bilgileriniz gözükmemekte.Profilinizde bulunan Kaydet ve Değişiklik Yap butonları ile yaptığınız değişiklikleri kaydedebilirsiniz. Değişiklik Yap butonuna tıkladığınız durumda karşınıza aynı şekilde Profil İsmi, Ülke , Şehir , İlçe , Mahalle ve Profil resminiz için Resim yükleme kısmı çıkmakta Burada yapacağınız değişiklikleri Kaydet butonu ile kalıcı hale getirebilirsiniz.



7.0 VERİ YÖNETİMİ

7.1 Veri Tabanı Mimarisi

- Friends Koleksiyonu:

Bu koleksiyon, bir kullanıcının diğer bir kullanıcıyı arkadaşı olarak eklediği durumları saklar. Koleksiyon, iki sütun içerir: userMail ve friendMail. Bu sütunlar arasında iki adet anahtar (key) bulunmaktadır: userMail ve friendMail. userMail, arkadaş ekleyen kullanıcının e-posta adresini içerir ve friendMail arkadaş olarak eklenen kullanıcının e-posta adresini içerir.

- Message Koleksiyonu:

Bu koleksiyon, kullanıcılar arasındaki mesajlaşma işlemleri için kullanılır. Koleksiyon, dört sütun içerir: userMailSendMessage, userMailGetMessage, message ve messageTime. Bu sütunlar arasında iki adet anahtar (key) bulunmaktadır: userMailSendMessage, userMailGetMessage. userMailSendMessage, mesajı gönderen kullanıcının e-posta adresini içerir. userMailGetMessage, mesajı alan kullanıcının e-posta adresini içerir. message, gönderilen mesajı içerir. messageTime, mesajın gönderildiği tarihi ve saatini tutar.

- User Koleksiyonu:

Bu koleksiyon, kullanıcıların bilgilerini saklar. Koleksiyon, yedi sütun içerir: userName, userMail, userPassword, userCountry, userCity, userDistrict, userNeighbourhood ve userCreated. Bu sütunlar arasında bir adet anahtar (key) bulunmaktadır: userMail bulunmaktadır. userName, kullanıcının adını içerir. userMail, kullanıcının e-posta adresini içerir. userPassword, kullanıcının şifresini içerir. userCountry, kullanıcının ülkesini içerir. userCity, kullanıcının şehrini içerir. userDistrict, kullanıcının ilçesini içerir. userNeighbourhood, kullanıcının mahallesini içerir. userCreated, kullanıcının hesap oluşturma tarihini ve saatini tutar.

Bu koleksiyonlar arasındaki ilişkiler şu şekildedir:

- Bir kullanıcının birden fazla arkadaşı olabilir. Friends koleksiyonu, bu arkadaşlıkların listesini saklar. Friends koleksiyonundaki belgelerin userMail alanı, kullanıcının kimliğini belirtir ve friendMail alanı, kullanıcının arkadaşının kimliğini belirtir. Bu nedenle, Friends koleksiyonu, User koleksiyonu ile bir ilişki içerir.
- Kullanıcılar birbirleriyle mesajlaşabilir. Message koleksiyonu, kullanıcılar arasındaki mesajları saklar. Message koleksiyonundaki belgelerin userMailSendMessage ve userMailGetMessage alanları, mesajlaşan kullanıcıların kimliklerini belirtir. Bu nedenle, Message koleksiyonu, User koleksiyonu ile bir ilişki içerir.

8.0 PROJE ZAMAN ÇİZELGESİ

12.03.2023	Proje Seçimi
15.03.2023	Grup Görev Dağılımı
17.03.2023	1. Grup Toplantısı
18.03.2023	Proje Başlangıç Tarihi
20.03.2023	2. Grup Toplantısı
21.03.2023	Ürün Tanıtım Kataloğu
23.03.2023	Hikaye Kartları
03.04.2023	Kullanım Kılavuzu v1.0
07.04.2023	3. Grup Toplantısı
17.04.2023	4. Grup Toplantısı
01.05.2023	Community v1.0
05.05.2023	5. Grup Toplantısı
07.05.2023	Proje Gereksinim Raporu
10.05.2023	Proje İş Planı
12.05.2023	İlk Demo Prototip v1.0