

# SEYAHAT DOSTU

## GEREKSİNİM DOKÜMANI

### Takım 4

Yunus Efe Yılmaz  
Ramazan Aslan  
Eren Altay  
Mahmut Bahadır Çalışkan  
Mustafa Erkin Dal

# İÇİNDEKİLER

<b>1. Önsöz</b>	<b>3</b>
<b>2. Giriş</b>	<b>4</b>
<b>3. Kullanıcı Gereksinimleri</b>	<b>4</b>
<b>4. Sistem Mimarisi</b>	<b>6</b>
<b>5. Yazılım Geliştirme Süreci</b>	<b>7</b>
<b>6. Sistem Gereksinimleri</b>	<b>8</b>
<b>7. Kaynakça</b>	<b>9</b>

# 1. Önsöz

Bu rapor, Yazılım Mühendisliği dersi kapsamında hazırlanan gereksinim belgesi raporudur. Bu belge, Yunus Efe Yılmaz, Ramazan Aslan, Eren Altay, Mahmut Bahadır Çalışkan ve Mustafa Erkin Dal tarafından 13.05.2024 tarihinde tamamlanan bir çalışmanın ürünüdür.

Bu belge, projenin gereksinimlerini ve hedeflerini tanımlamak amacıyla hazırlanmıştır. Projenin başarılı bir şekilde tamamlanması ve hedeflerine ulaşılması için, ortak bir anlayış ve yönergelerin belirlenmesi gereklidir. Bu belge, bu ortak anlayışın oluşturulması ve proje sürecinin yönetilmesi için bir temel oluşturmayı amaçlamaktadır.

Bu belge, projenin ilerleyen aşamalarında bir referans kaynağı olarak kullanılabilir ve proje sürecinin yönetimi ve izlenmesine yardımcı olabilir.

Bu belgenin hazırlanmasında emeği geçen herkese teşekkür ederiz.

Saygılarımızla,

## 2. Giriş

"Seyahat Dostu" web sitesi, seyahat etmeyi seven insanları bir araya getirerek ortak yolculuk deneyimlerini kolaylaştıran bir platformdur. Amacı, kullanıcıların seyahat planlarını oluşturmak, seyahat rotalarını paylaşmak ve birlikte seyahat etmek isteyen yolcuları bulmak için etkili bir çözüm sunmaktır.

Projede 3 temel parça 3 ekip tarafından geliştirilecektir:

- Backend: Kullanıcı ön tarafta yaptığı işlemlerin veri tabanı bağlantısı ile fonksiyonellik kazanması sağlanacaktır
- Frontend: Bu kısımda sistem tarafından belirlenen kullanıcıya hizmet verilmesi için gerekli işlemlerin arayüz üzerinden yürütülmesi sağlanacaktır.
- Veritabanı: Bu kısımda sitedeki nesne bilgilerinin nasıl tutulması gerektiğinin bulunması gibi detaylar veritabanı üzerinden sağlanacaktır.

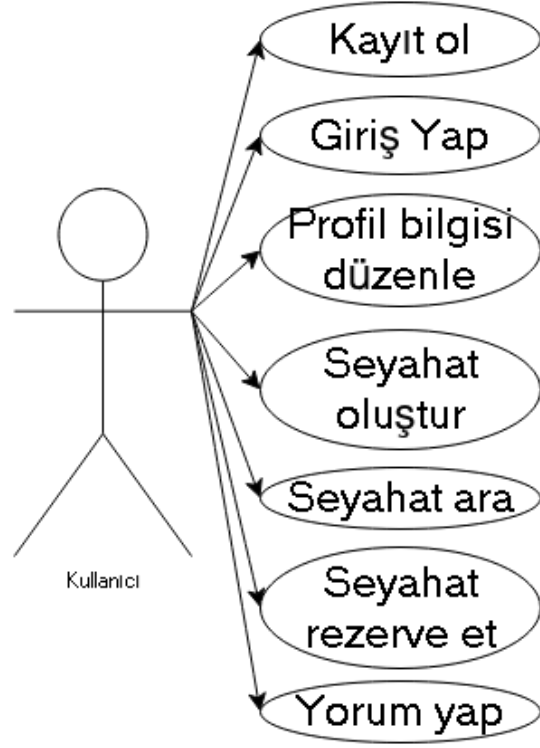
## 3. Kullanıcı Gereksinimleri

Kullanıcı gereksinimleri, müşteriye sunulacak özelliklerin ne olduğunu doğal bir dille açıklayan gereksinimlerdir. Fonksiyonel ve fonksiyonel olmayan olarak ikiye ayrılabilir. Fonksiyonel gereksinimler bir sistemin neyi başarması gerektiğini belirler[1]. Aşağıda projenin fonksiyonel gereksinimleri verilmiştir:

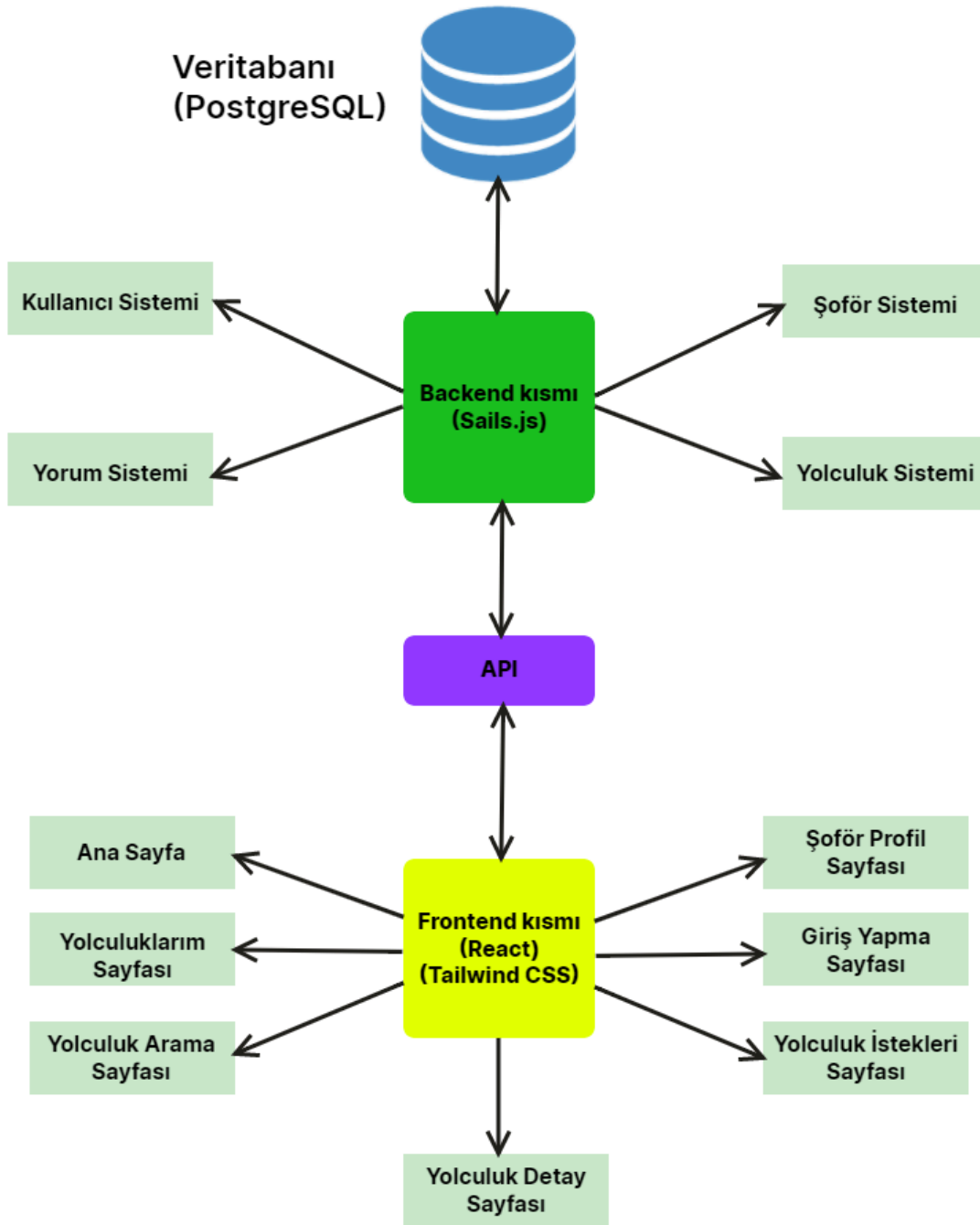
- Kullanıcılar kayıt olabilir ve oturum açabilir.
- Kullanıcılar profil bilgilerini düzenleyebilir.
- Kullanıcılar seyahat oluşturabilir.
- Kullanıcılar seyahatler arasında arama yapabilir ve rezervasyon yapabilir.
- Kullanıcılar hakkında yorum ve değerlendirme yapabilir.

Fonksiyonel olmayan gereksinimler ise, sistem kısıtları veya olması beklenen durumlar olarak tarif edilebilir[1]. Aşağıda fonksiyonel olmayan gereksinimler verilmiştir:

- Sistem 7/24 kullanılabilir olmalı.
- Kullanıcılar istedikleri konumdan sisteme girebilmeli.
- Sistem hızı kullanıcıları çok bekletmemeli.
- Sistem hataları geliştiriciler tarafından kolayca düzeltilebilmeli.



## 4. Sistem Mimarisi



Backend kısmı verilerin saklandığı veritabanı ile iletişim kurup kullanıcı isteklerine göre veri iletimi sağlayan kısımdır[2]. SailsJS, NodeJS için hazırlanmış gerçek zamanlı bir MVC(Model-View-Controller)[3] kütüphanesidir. Veritabanının da ise ilişkisel bir veritabanı[4] olan postgresql kullanılmıştır.

Frontend tarafından gelen istekler API aracılığıyla Backend kısmına gelir. İstenen istek gerekli sistem üzerinde işlem görür. Bu istekler veri ekleme, çıkarma, silme veya güncelleme olabilir[5]. Bu istekler sistemde işlenir ve gerekli veriler veritabanında güncellenir. Veriler Frontend kısmına yine API üzerinden JSON formatında aktarılır.

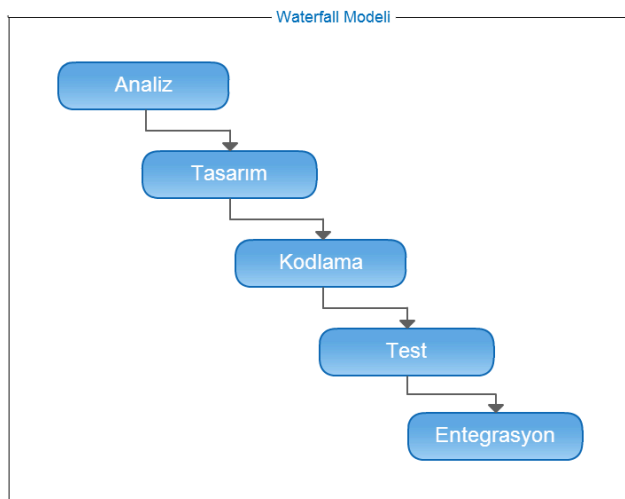
Frontend kısmı kullanıcının etkileşime geçtiği kısımdır[6]. Kullanıcı gitmek istediği sayfayı Frontend den ister. Frontend sayfayı oluşturur. Oluşan sayfa içinde gerekli alanlar Backend kısmından istenir. Gelen veriler alanlara doldurulup kullanıcıya iletilir.

## 5. Yazılım Geliştirme Süreci

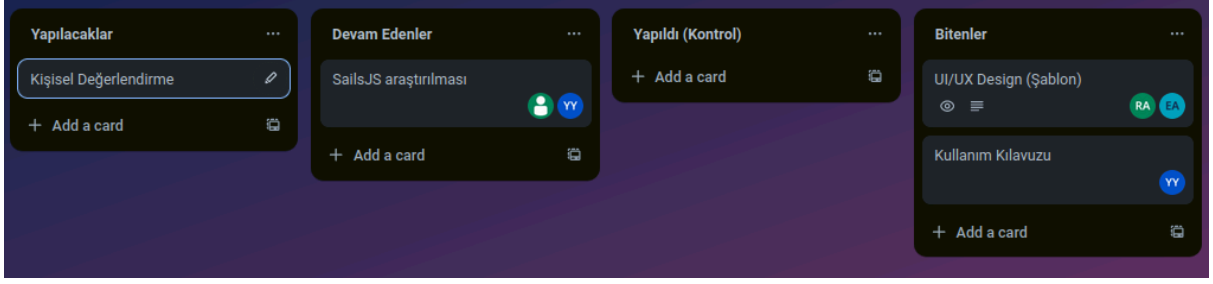
Projemizde Şelale Modeli benimsenmiştir, bu karar geleneksel olarak kullanılan Şelale Modeli'nin sağladığı adımlı ve belirli bir süreç olması, ayrıca proje gereksinimlerinin başlangıçta net bir şekilde tanımlanması gerektiği gerçeği yer almıştır. Şelale Modeli, projenin başından sonuna kadar belirli bir sırayla ilerlenmesini sağlar[7].

Ayrıca, proje sürecinde paralel olarak Kanban yöntemi de takip edilmiştir. Kanban, iş akışını görsel olarak yönetilmesini sağlar ve iş öğelerinin tahtalar üzerindeki ilerlemesini izlenmesine olanak tanır. Bu yöntem, özellikle iş öğelerinin önceliklendirilmesi ve iş akışının optimize edilmesi konularında yardımcı olmaktadır[8].

Projede takip edilen Şelale modeli diyagramı şu şekildedir:



Projenin analiz aşamasında kullanılan kanban tahtasının bir örneği:



## 6. Sistem Gereksinimleri

Sistem gereksinimleri, bir sistemin yapması gerekenleri ve müşteriye sunulan veya sunulmayan özellikleri belirleyen önemli özelliklerdir[9]. Bu gereksinimler, genellikle fonksiyonel ve fonksiyonel olmayan olarak kategorize edilir. Fonksiyonel gereksinimler, sistemin ne yapması gerektiğini tanımlarken, fonksiyonel olmayan gereksinimler genellikle sistem performansı, güvenilirlik ve kullanılabilirlik gibi konulara odaklanır.

### Fonksiyonel Gereksinimler:

#### Ana Sayfa:

- Kullanıcıya filtrelenebilir özellikleri gösteren bir form elemanı sunmalı.
- Giriş yapmayan kullanıcılar yer ayırmaya çalıştıklarında kullanıcıları giriş yapma ekranına yönlendirmeli.
- Kullanıcıya sistemi kullanırken uyulması gereken güvenlik kurallarını göstermeli.
- Kullanıcıya sıkça sorulan soruların cevaplarını göstermeli.

#### Yolculuk Arama Sonucu Ekranı:

- Kullanıcıya filtrelediği özelliklere göre güncel yolculukları göstermeli.
- Kullanıcıya fiyat bilgisi, tahmini süre, araç modeli, saat ve şoförün ad ve soyad bilgilerini göstermeli.
- Kullanıcının uygun yolculuktan yer ayırmasını sağlamalı.

#### Yolculuk Detay Ekranı:

- Kullanıcıya yolculuk arama sonucu ekranında gözükmeyen şoför bilgilerini (sigara kullanımı, ilgi alanları, varsa şoförün özel mesajı) göstermeli.

#### Giriş Yap Ekranı:

- Kullanıcıların hesaplarına giriş yapmasını sağlamalı.
- Hesap oluşturma seçeneği sunmalı.

#### Kullanıcı Yolculuk Listesi Ekranı:

- Kullanıcının oluşturduğu veya katıldığı yolculukları detaylı bir şekilde (tarih, ücret vb.) listelemeli.

#### Şoför Profili Ekranı:



- Şoförün adı, kullandığı araç, puanı, geçmiş yolculukları ve yorumları görüntülemeli.

Yolculuk İstekleri Ekranı:

- Kullanıcıdan gelen yolculuk istekleri görüntülenmeli.
- Yolculuğu onaylama ve reddetme seçenekleri sunulmalı.

Fonksiyonel Olmayan Gereksinimler:

- Uygulama kesintisiz hizmet verebilmelidir.
- Kullanıcılar bütün platformlardan sisteme erişebilmelidir.
- Adminler sisteme her yerden ve her zaman erişebilmelidir.
- Sistem hızı kullanıcı isteklerini karşılayabilecek düzeyde olmalıdır.
- Kullanıcıların şifre ve diğer bilgileri güvenli şekilde saklanmalıdır.

## 7. Kaynakça

1. <https://www.cogen.com.tr/blog/functional-non-functional-requirements-nedir>
2. <https://www.techtarget.com/whatis/definition/front-end>
3. <https://developer.mozilla.org/en-US/docs/Glossary/MVC>
4. <https://www.oracle.com/database/what-is-a-relational-database/>
5. <https://blog.postman.com/what-are-http-methods/>
6. <https://roadmap.sh/frontend>
7. <https://business.adobe.com/blog/basics/waterfall>
8. <https://www.planview.com/resources/guide/introduction-to-kanban/>
9. <https://www.inflectra.com/Ideas/Topic/Requirements-Definition.aspx>