

JourneyVista – İş Analizi Raporu

Ad Soyad: Berfin Aleyna Sukuti

Ünvan: İş Analisti & IT Uzmanı

Tarih: Mart 2025



İçindekiler

1. Proje Tanımı.....	3
1.1. Proje Adı	3
1.2. Projenin Amacı.....	3
1.3. Proje Kapsamı	3
2. Paydaşlar	4
2.1. Ana Paydaşlar.....	4
2.2. Paydaş Beklentileri.....	4
3. Mevcut Durum Analizi ve Gereksinimler.....	5
3.1. Mevcut Durum Analizi.....	5
3.2. Fonksiyonel Gereksinimler	5
3.3. Fonksiyonel Olmayan Gereksinimler.....	5
4. İş Akışı ve Kullanım Senaryoları.....	6
4.1. İş Akışı	6
4.2. Kullanım Senaryoları	6
5. Teknolojik Gereksinimler	9
6. Risk Analizi	9
7. Sonuç ve Yol Haritası	10

1. Proje Tanımı

1.1. Proje Adı

JourneyVista (Seyahat Manzarası)

1.2. Projenin Amacı

JourneyVista, tur operatörleri ve seyahat acenteleri için geliştirilen bir platformdur. Platform, turların dünya haritası üzerinde konum bazlı olarak görüntülenmesini ve tur süresince konum güncellenmesini sağlar. Kullanıcılar turların başlangıç ve bitiş tarihlerini takip edebilir ve aktif turların harita üzerinde görüntülenmesini sağlayabilir. Ayrıca tüm bu bilgiler de API ile kolayca kullanılabilir.

1.3. Proje Kapsamı

- Tur operatörleri için turları dijital olarak yönetme imkânı.
- Harita tabanlı tur görüntüleme.
- Kullanıcı girişi ve yetkilendirme.
- Tur rehberi ve tur bilgileri entegrasyonu.
- API ile otomatik veri senkronizasyonu.

2. Paydaşlar

2.1. Ana Paydaşlar

- Tur operatörleri
- Seyahat acenteleri
- Tur rehberleri
- Müşteriler (tur katılımcıları)

2.2. Paydaş Beklentileri

- Tur bilgilerine kolay erişim.
- Anlık konum güncellemesi. (Tur bilgileri panele manuel olarak girilmektedir.)
- Kullanıcı dostu arayüz.
- Hata toleransı düşük bir sistem.

3. Mevcut Durum Analizi ve Gereksinimler

3.1. Mevcut Durum Analizi

JourneyVista, seyahat acenteleri için büyük bir kolaylık sağlamak amacıyla geliştirilmektedir. Halihazırda tur yönetiminde manuel süreçler bulunmakta ve bu da operasyonel verimsizliğe neden olmaktadır.

3.2. Fonksiyonel Gereksinimler

- Kullanıcı girişi ve yetkilendirme.
- Tur ekleme, güncelleme ve silme.
- Turların haritada ve listede gösterilmesi.
- Harita üzerinde güncel konum belirleme.
- Tur rehberi bilgileri entegrasyonu.
- Turların başlangıç ve bitiş tarihine göre otomatik olarak haritadan kaldırılması.
(Panel üzerinden silinmez.)

3.3. Fonksiyonel Olmayan Gereksinimler

Sistem performansı: Hızlı ve kesintisiz çalışmalı.

Kullanılabilirlik: Basit ve etkili bir arayüz sunulmalı.

Güvenlik: Kullanıcı bilgilerinin korunması.

Entegrasyon: API entegrasyonu ile harita verisi senkronizasyonu.

4. İş Akışı ve Kullanım Senaryoları

4.1. JourneyVista İş Akışı

- Admin Sisteme Giriş Yapar
 - Sisteme yalnızca yetkili admin kullanıcı giriş yapabilir.
 - Güvenli kimlik doğrulama sağlanır (kullanıcı adı + şifre).
 - Başarılı giriş sonrası yönetim paneline yönlendirilir.
- Yeni Tur Bilgileri Eklenir
 - Admin, “Yeni Tur Ekle” ekranına ulaşır.
 - Tur bilgileri adım adım girilir:
 - Tur Adı
 - Tur Başlangıç ve Bitiş Tarihi
 - Gün Bazlı Lokasyon Listesi (1. Gün: Tokyo, 2. Gün: Kyoto vb.)
 - Tur Rehberi Adı
 - Tur Süresi
 - Tur Görseli

 Her günün lokasyonu ayrı ayrı tanımlanır. Bu bilgi daha sonra harita üzerinde hangi gün nerede olduğunu göstermek için kullanılır.

- Tur API Üzerinden Sisteme Kaydedilir
 - Bilgiler backend'e (Django REST API) gönderilir.
 - Veritabanına güvenli şekilde kaydedilir.
 - Başarılı kayıt sonrası kullanıcı bilgilendirilir.
- Tur Haritada ve Listede Gösterilir

Harita ekranında:

 - Kırmızı lokasyon pin'i ile tur işaretlenir.
 - anki gün hangi şehirde ise, o şehir vurgulanır.
 - Liste görünümünde:
 - Tüm aktif turlar görüntülenir (isim, süre, gün, rehber vb.).
- Her Gün Konum Otomatik Güncellenir
 - Sistem, sunucu saati üzerinden gün kontrolü yapar.
 - Mevcut tarihe göre o günün şehir bilgisi gösterilir.
 - Haritada otomatik olarak o günün konumu aktif olur.
 - Kullanıcının manuel müdahalesine gerek kalmaz.
- Tur Süresi Dolduğunda Sistemden Otomatik Kaldırılır
 - Sistem, tur bitiş tarihine ulaştığında:
 - Harita üzerindeki pin kaldırılır.
 - Liste görünümünden silinir.
 - Tüm bilgiler sistemde arşivlenir (isteğe bağlı).

Opsiyonel:

Admin tur bilgilerini dilerse güncelleyebilir veya silebilir (süre bitmemişse).

API üzerinden gelen tüm veriler, haritayla senkronize çalışır.

Ekstra Notlar:

Bu yapı günlük şehir takibi, rehber kontrolü ve otomatik gösterim gibi kritik operasyonel süreçleri kolaylaştırır.

İş akışı hem kullanıcı deneyimini artırır, hem de hataları minimuma indirir.

JourneyVista İş Akış Şeması

ADMIN SİSTEME GİRİŞ YAPAR

- Sisteme yalnızca yetkili admin kullanıcı giriş yapabilir.
- Güvenli kimlik doğrulama sağlanır (kullanıcı adı + şifre).
- Başarılı giriş sonrası yönetim paneline yönlendirilir.

YENİ TUR BİLGİLERİ EKLENİR

Tur bilgileri adım adım girilir:

Tur Adı, Tur Başlangıç ve Bitiş Tarihi, Gün Bazlı Lokasyon Listesi (1. Gün: Tokyo vb.), Tur, Rehberi Adı, Tur Süresi, Tur Görseli

TUR API ÜZERİNDEN SİSTEME KAYDEDİLİR

- Bilgiler backend'e (Django REST API) gönderilir.
- Veritabanına güvenli şekilde kaydedilir.
- Başarılı kayıt sonrası kullanıcı bilgilendirilir.

TUR HARITADA VE LİSTEDE GÖSTERİLİR

- Kırmızı lokasyon pin'i ile tur işaretlenir. (Haritada)
- Hanki gün hangi şehirde ise, o şehir vurgulanır.
- Tüm aktif turlar görüntülenir (isim, süre, gün, rehber vb.).

HER GÜN KONUM OTOMATİK GÜNCELLENİR

- Sistem, sunucu saati üzerinden gün kontrolü yapar.
- Mevcut tarihe göre o günün şehir bilgisi gösterilir.
- Haritada otomatik olarak o günün konumu aktif olur.
- Kullanıcının manuel müdahalesine gerek kalmaz.

TUR SÜRESİ DOLDUĞUNDA SİSTEMDEN OTOMATİK KALDIRILIR

- Sistem, tur bitiş tarihine ulaştığında:
- Harita üzerindeki pin kaldırılır.
- Liste görünümünden silinir.
- Tüm bilgiler sistemde arşivlenir (isteğe bağlı).

4.2. Kullanım Senaryoları

Senaryo 1: Admin Kullanıcısı Tur Ekliyor

- Admin panele giriş yapar.
- "Yeni Tur Ekle" butonuna tıklar.
- Tur bilgilerini doldurur.
- "Kaydet" butonuna basarak turu ekler.
- Tur bilgileri haritada görüntülenir.

Senaryo 2: Kullanıcı Haritada Tur Bilgilerini Görüyor

- Kullanıcı web sitesini açar.
- Harita üzerinde aktif turları görür.
- Bir tura tıkladığında tur bilgileri açılır.
- Listede turlar bilgi akışında görünür.

5. Teknolojik Gereksinimler

Frontend: React.js

Backend: Django REST Framework

Database: SQLite (veya PostgreSQL)

API: Google Maps API veya OpenStreetMap API

Hosting: AWS (EC2, S3, RDS gibi servisler) veya Render

6. Risk Analizi

Aşağıda potansiyel riskler ve çözüm önerileri listelenmiştir:

RISK	ÇÖZÜM
HARITA API HATALARI	API anahtarını doğru konfigüre etmek, alternatif API düşünmek
KULLANICI VERİSİ KAYBI	Veri tabanı yedekleme stratejisi oluşturmak
TRAFİK YOĞUNLUĞU	Sunucu kapasitesini esnek hale getirmek

7. Sonuç ve Yol Haritası

- ✓ **MVP Aşaması:** Temel tur ekleme ve harita entegrasyonu.
- ✓ **Geliştirme Aşaması:** Kullanıcı dostu arayüz iyileştirmeleri.
- ✓ **Lansman:** JourneyVista beta sürüm yayınlanması.