

DESIGN DOCUMENT (TASARIM BELGESİ)

Proje Adı: TripGuide

Takım: TrueGuiders

Katkıda Bulunanlar (List of Contributors)

- Saadet Cansu Baktıroğlu
- İrem Keser
- Efe Selim Sürekli
- Zeynep Ay
- Elif Sema Küçük

İçindekiler (Table of Contents)

- Görev Matrisi (Task Matrix)
- System Overview
- Implementation Details
- Use Case Support in Design
- Design Decisions
- GitHub Commit Requirement

Görev Matrisi (Task Matrix)

Bölüm	Sorumlu Kişi	Açıklama
System Overview	Saadet Cansu Baktıroğlu (PM)	Proje kapsamı ve mimari tanımı
Implementation Details	İrem Keser, Elif Sema Küçük, Saadet Cansu Baktıroğlu	Backend yapısı, API tasarımı ve kod detayları
Visual Interfaces	Efe Selim Sürekli	Arayüz tasarımları ve wireframe açıklamaları
Use Case Support	Saadet Cansu Baktıroğlu, Zeynep Ay, İrem Keser	Senaryoların seçimi, test edilebilirliği ve akış tasarımı
Design Decision & Compilaiton	Elif Sema Küçük, Efe Selim Sürekli	Teknoloji karşılaştırmaları ve belgenin derlenmesi

1. System Overview (Sistem Genel Bakış)

1.1 Brief Project Description

TripGuide; kullanıcıların seyahat etmek istedikleri şehri, gezi süresini (kısa/uzun) ve ilgi alanlarını (müze, eğlence, tarih vb.) belirterek kendilerine özel rota ve plan oluşturmasını sağlayan akıllı bir tur rehberi platformudur. Sistem, kullanıcı tercihlerini analiz ederek mekan önerileri sunar ve bunları bir zaman çizelgesi üzerinde planlar.

1.2 System Architecture

Proje, sürdürülebilirlik ve modülerlik sağlamak amacıyla **Katmanlı Mimari (Layered Architecture)** yapısında tasarlanmıştır:

- **Presentation Layer (Frontend):** Kullanıcı etkileşimini sağlayan web arayüzü.
- **Business Layer (Backend):** Rota hesaplama, öneri motoru ve iş mantığının çalıştığı Spring Boot servisi.
- **Data Access Layer (Database):** Şehir, mekan ve kullanıcı verilerinin tutulduğu PostgreSQL veritabanı katmanı.

1.3 Technology Stack

- **Backend:** Java (Spring Boot)
- **Frontend:** HTML, CSS, JavaScript
- **Database:** PostgreSQL
- **Build Tool:** Maven
- **Version Control:** GitHub

2. Implementation Details

2.1 Codebase Structure

- **backend/:** Controller, Service, Repository, Model
- **frontend/:** HTML/CSS/JS sayfaları
- **database/:** schema.sql ve seed.sql
- **resources/:** application.properties (DB ayarları)

2.2 Key Implementations

- **Öneri Motoru:** Kullanıcı kategorilerine göre mekanları filtreler ve basit puanlama ile sıralar.
- **Rota Planlayıcı:** Seçili mekanları kısa/uzun gezi süresine göre gün–saat bloklarına yerleştirir.
- **Gezi Puanlama:** Kullanıcı tamamladığı geziler için 1–5 arası puan verebilir.

2.3 Component Interfaces (API Endpoints)

- POST /api/auth/register → Kullanıcı kaydı
- POST /api/trips/create → Tercihlerden rota oluşturma
- GET /api/trips/{userId} → Kayıtlı planları listeleme
- POST /api/trips/{tripId}/rate → Gezi puanlama

2.4 Visual Interfaces

TripGuide arayüzü basit ve tek sayfalık bir yapıdadır:

- **Üst Menü:** Logo ve temel gezinme linkleri.
- **Ana Sayfa Formu:** Şehir ve gün seçimi + "Rota Oluştur" butonu.
- **Özellik Kartları:** Rota planlama ve öneri özelliklerini kısaca açıklar.
- **Popüler Şehirler:** Şehir kartları üzerinden hızlı gezi fikirleri sağlar.
- **Örnek Gezi Planı:** Popüler şehirler için birkaç günlük örnek zaman çizelgesi gösterilir.
- **Hakkımızda & İletişim:** Kısa ekip tanıtımı ve iletişim formu.

The screenshot shows the homepage of the TripGuide website. At the top, there is a purple header bar with the logo 'TrueGuiders' and navigation links for 'Ana Sayfa', 'Şehirler', 'Hakkımızda', and 'İletişim'. Below the header, the main title 'Mükemmel Seyahatin Planlayıcısı' is displayed in a large, bold, purple font. Underneath the title, a subtitle reads 'Şehir seçin, gün sayısını belirleyin, size özel hazırlanmış gezi rotanızı alın!'. A central search form contains two dropdown menus: 'Şehir Seçin' (City Select) and 'Kalış Süresi (Gün)' (Workout Duration (Day)), both with placeholder text 'Bir şehir seçin' and 'Gün sayısını seçin'. Below these is a large red button labeled 'Rota Oluştur' (Create Route). The background of the page is light blue.

The screenshot shows the 'Neden TrueGuiders?' (Why TrueGuiders?) section of the website. It features three cards with icons and descriptions:

- Kişiselleştirilmiş Rotalar**: An icon of a map with a route line. Description: 'İlgili alanlarınız ve zamanınızla özelleştirilmiş günlük gezi planları oluşturuyoruz.'
- Saat Bazlı Planlama**: An icon of a clock. Description: 'Her aktivite için ideal zaman aralıklarını belirleyerek zamanınızı en verimli şekilde kullanın.'
- Yerel Öneriler**: An icon of a heart. Description: 'Sadece turistik yerler değil, yerel halkın da tavsiye ettiği gizli hazineleri keşfedin.'

Popüler Şehirler

Antalya

Turkuaz sahilleri, tarihi kaleleri ve doğal güzellikleriyle Türkiye'nin incisi.

[Keşfet →](#)

İstanbul

Asya ve Avrupa'nın buluştuğu, tarih ve modernliğin iç içे geçtiği eşsiz şehir.

[Keşfet →](#)

İzmir

Ege'nin incisi, antik tarih ve modern yaşamın mükemmel uyumu.

[Keşfet →](#)

Kapadokya

Peri bacaları, balon turları ve yeraltı şehirlerle büyüleyici bir deneyim.

[Keşfet →](#)

Örnek Gezi Planı: Antalya (3 Gün)

İletişim

1. Gün

2. Gün

3. Gün

09:00 - 11:00 Kaleiçi'ni Keşfedin

Tarihi Kaleiçi bölgesinde dar sokaklarda yürüyüş, Hadrian Kapısı ve tarihi evleri görün.

11:30 - 13:00 Antalya Müzesi

Türkiye'nin en önemli müzelerinden birinde antik eserleri keşfedin.

13:30 - 14:30 Öğle Yemeği

Liman bölgesinde taze balık ve meze keyfi yapın.

15:00 - 17:00 Düden Şelalesi

Şehir merkezine yakın bu doğal güzellikleri keşfedin.

19:30 - 21:30 Akşam Yemeği

Kaleiçi'nde geleneksel Türk mutfağının lezzetlerini deneyin.

Adınız

E-posta

Konu

Mesajımız

[Mesaj Gönder](#)

Adres

Çankaya
Ankara, Türkiye

Telefon

+90 507 156 05 62

E-posta

info@TrueGuiders.com

3. Use Case Support in Design (Tasarımın Senaryo Desteği)

3.1 Use Case Selection

Proje Planı (Requirements Document) içerisindeki Fonksiyonel Gereksinimlerden seçilen 4 ana kullanım senaryosu:

- Kullanıcı Rota Oluşturur**
- Kullanıcı Planı Kaydeder**
- Kullanıcı Önerileri Filtreler**
- Kullanıcı Planı Değerlendirir (Puanlar)**
- Popüler Şehirler İçin Gezi Planı Oluşturur**

3.2 Requirement Mapping

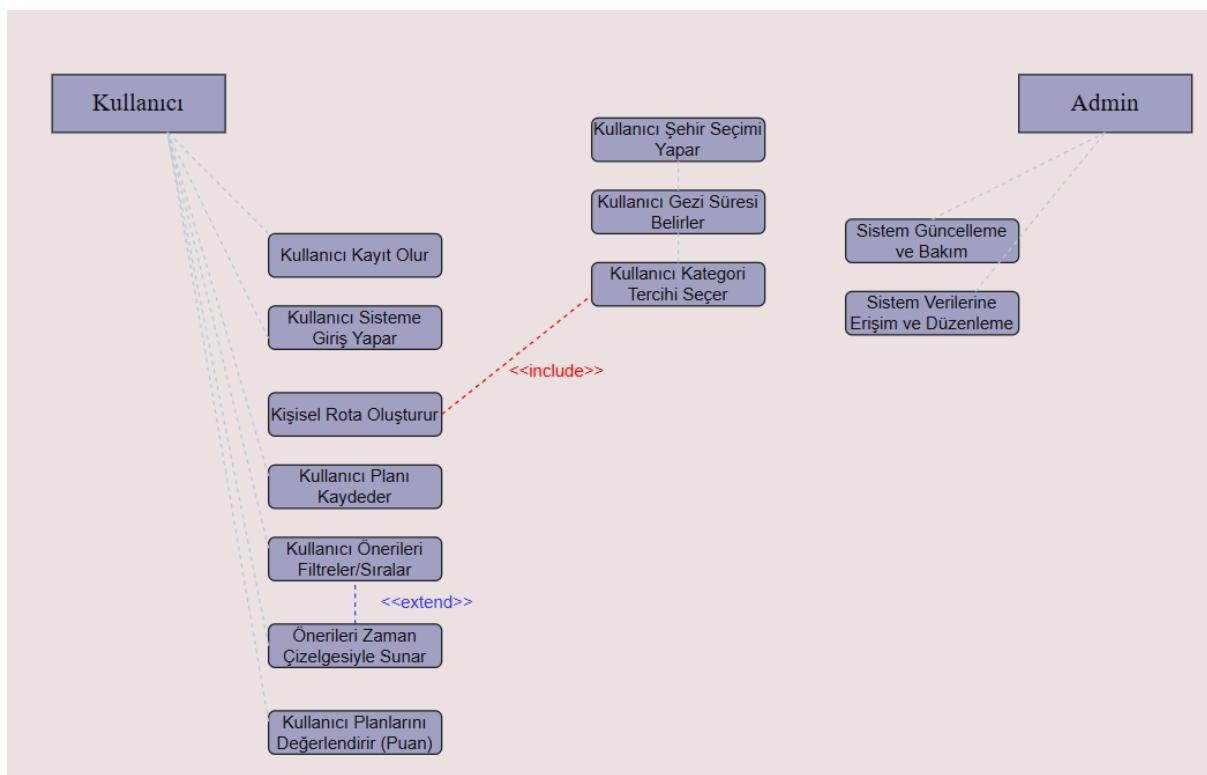
Seçilen senaryoların Gereksinim Belgesi ile eşleşmesi:

- UC-1 (Rota Oluşturma):** Kullanıcı şehir seçimi yapabilir(1), Kullanıcı gezi süresini (kısa/uzun) belirtebilir (2), Kullanıcı kategori tercihi seçebilir (şehir merkezi, müze, eğlence parkı vb.) (3), Sistem SQL veri tabanından öneriler üretir (4), Sistem önerileri zaman çizelgesiyle sunar (5) gereksinimlerini kapsar. (Şehir, süre, kategori seçimi ve öneri üretimi).

- **UC-2 (Plan Kaydetme):** Kullanıcı planı kaydedebilir.(6) gereksinimlerini kapsar.
- **UC-3 (Filtreleme):** Kullanıcı önerileri filtreleyebilir/sıralayabilir.(8) gereksinimlerini kapsar.
- **UC-4 (Değerlendirme):** Kullanıcı planlarını değerlendirebilir (puan).(9) gereksinimlerini kapsar.
- **UC-5(Gezi Planı Oluşturma):** Kullanıcı şehir seçimi yapabilir.(1), Sistem önerileri zaman çizelgesiyle sunar.(5) gereksinimlerini kapsar.

3.3 Use Case Design & Architecture Support

- **UC-1 Tasarımı:** Kullanıcı istek attığında **TripController** isteği karşılar → **RecommendationService** SQL veritabanından (**PlaceRepository**) kategoriye uygun mekanları çeker → Algoritma bunları sıralar ve **Trip** nesnesi olarak döner.
- **UC-2 Tasarımı:** Oluşturulan rota nesnesi, kullanıcı onayı ile **SavedTrips** tablosuna, kullanıcının ID'si ile ilişkilendirilerek (Many-to-One) kaydedilir.
- **UC-3 Tasarımı:** Kullanıcı filtreleme/sıralama isteği gönderir → **TripController** isteği alır → **RecommendationService** mevcut öneri listesini filtreleme kriterlerine göre düzenler → Güncellenmiş liste kullanıcıya döner.
- **UC-4 Tasarımı:** Kullanıcı bir planı puanlar → **TripController** puanı alır → **RatingService** ilgili **Trip** kaydına puanı işler → Güncellenmiş değerlendirme bilgisi veritabanına kaydedilir.
- **UC-5 Tasarımı:** Kullanıcı şehir seçimini gönderir → **TripController** isteği alır → **RecommendationService** seçilen şehre ait mekanları getirir → **TimelineGenerator** bu mekanlardan zaman çizelgesi oluşturur → Oluşturulan gezi planı kullanıcıya sunulur.



3.4 Demo Requirement

Bu 5 kullanım senaryosu (Rota oluşturma, kaydetme, filtreleme, puanlama, popüler şehirler için plan oluşturma) dönem sonu demosunda canlı olarak gösterilecek fonksiyonlardır.

4. Design Decisions (Tasarım Kararları)

4.1 Technology Comparisons

Backend: Spring Boot (Java) vs. Node.js

- Node.js esnek bir yapı ve yüksek I/O performansı sunarken, Spring Boot (Java) güçlü tip denetimi, olgun bir ekosistem ve kurumsal düzeyde yapı (DI, AOP) sağlar. Ekip yetkinliği ve projenin SQL ile olan güçlü bağı nedeniyle Spring Boot tercih edilmiştir.

Database: PostgreSQL vs. MySQL

- MySQL hızlı okuma işlemleriyle bilinir. Ancak PostgreSQL, daha karmaşık sorgu desteği, veri bütünlüğü (ACID uyumluluğu) ve ileride gerekebilecek coğrafi veri (PostGIS) desteği konularında avantajlıdır. Projemizdeki rota mantığı için PostgreSQL daha uygun görülmüştür.

4.2 Decision Justifications

- Spring Boot Tercihi:** Projenin backend kısmında, ekip üyelerinin aşina olduğu ve kurumsal standartlarda geliştirme imkanı sunan Spring Boot tercih edilmiştir. Özellikle Maven bağımlılık yönetimi ve REST API oluşturma kolaylığı kararımızda etkili olmuştur.
- SQL Veritabanı Tercihi:** İlişkisel verilerin (Kullanıcı -> Rota -> Mekan) tutarlılığı kritik olduğu için NoSQL yerine SQL (PostgreSQL) tercih edilmiştir.

5. GitHub Commit Requirement

5.1 Code Implementations & Interfaces

- Projenin kaynak kodları, tasarım diyagramları ve dökümantasyonu GitHub deposunda tutulacaktır.
- Commit Standardı:** Her commit, yapılan değişikliği net açıklayan mesajlar içerecektir.
- Her ekip üyesi kendi sorumluluğundaki modülü geliştirip pushlayacak, böylece katkılar şeffaf şekilde izlenebilecektir.

5.2 Technology Comparisons

- 4.1 bölümünde yapılan teknoloji karşılaştırmalarını desteklemek için kullanılan basit kod örnekleri veya analiz dökümanları da repository'ye commit edilecektir.

