

Mahalle

v1.0

13.12.2024

Kerim Ayberk Çıtak  
Onur Alp Gündüz  
Eylül Kanatlı  
Alperen Kemal Türk  
Boran Biçer

## Revizyon Geçmiři

Tarih	Açıklama	Yazar	Yorumlar
13.12.2024	v1.0	Ayberk-Onur	Demo

## İçindekiler Tablosu

<b>REVIZYON GEÇMİŞİ</b>	<b>ii</b>
<b>1. GİRİŞ 1</b>	
1.1 AMAÇ	1
1.2 KAPSAM	1
1.3 GENEL BAKIŞ	1
1.4 TANIMLAR VE KISALTMALAR	
<b>2. SİSTEME GENEL BAKIŞ</b>	<b>2</b>
<b>3. SİSTEM MİMARISI</b>	<b>2</b>
3.1 MİMARİ TASARIM	2
3.2 MİMARİ BİLEŞENLERİN AÇIKLANMASI	4
<b>4. VERİ TASARIMI (DATA DESIGN)</b>	<b>4</b>
4.1. VERİ BELİRTİMİ	4
4.2 VERİ TABANI TASARIMI	5
<b>5. KULLANICI ARAYÜZLERİ</b>	<b>5</b>
5.1 KULLANICI ARAYÜZLERİNE GENEL BAKIŞ	5
5.2 EKRAN RESİMLERİ	8
<b>6. KULLANILAN KÜTÜPHANELER VE ARAÇLAR</b>	<b>12</b>

## 1. Giriş

### 1.1 Amaç

Bu yazılım tasarım dokümantasyonu, **Mahalle Apartman Yönetim Sistemi**'nin mimari ve teknik detaylarını açıklamak amacıyla hazırlanmıştır. Hedef kitlesi; apartman yöneticileri, apartman sakinleri ve yazılım geliştiricilerdir. Bu doküman, sistemin tüm işlevselliklerini, bileşenlerini ve tasarım detaylarını aktarmayı, aynı zamanda projede yer alan yazılımcılar ve son kullanıcılar için bir rehber oluşturmayı hedeflemektedir.

### 1.2 Kapsam

Mahalle, apartman sakinlerinin ve yöneticilerinin apartman içi iletişim ve yönetim ihtiyaçlarını karşılayan bir dijital platformdur. Yazılım, aşağıdaki işlevsellikleri içerir:

- **Apartman sakinleri için:**
  - Yakındaki teknik servislerin listesini görüntüleme.
  - Yöneticiye anonim veya kimlik belirtilerek mesaj gönderme.
  - Etkinliklere katılım durumu belirleme ve görüntüleme.
  - Şikayetlerini yönetime bildirme.
- **Apartman yöneticileri için:**
  - Yeni etkinlikler oluşturma ve silebilme.
  - Apartman sakinlerinin listesini görüntüleme ve düzenleme.
  - Daire numaralarını düzenleme, sakin ekleme ve çıkarma.
  - Teknik servis bilgilerini yönetme.
  - Gelen şikayetleri inceleme ve çözüm sağlama.
  - Genel konuşma alanında mesaj paylaşma ve istenmeyen mesajları silme.

Sistem, apartman yönetim süreçlerini dijitalleştirerek zaman tasarrufu, daha etkin iletişim ve kullanıcı dostu bir deneyim sağlamayı amaçlamaktadır.

### 1.3 Genel Bakış

Bu doküman aşağıdaki bölümlerden oluşmaktadır:

- **Giriş:** Projenin amacı, kapsamı ve genel tanımı.
- **Sisteme Genel Bakış:** Sistemin genel bağlamı, işlevsellikleri ve özellikleri.
- **Sistem Mimarisi:** Sistem tasarımının genel yapısı ve modüler ilişkileri.
- **Veri Tasarımı:** Veritabanı yapıları ve veri akışları.
- **Kullanıcı Arayüzleri:** Sistem ile kullanıcı etkileşimleri ve arayüz detayları.
- **Kullanılan Kütüphaneler ve Araçlar:** Projede kullanılan yazılım araçları ve teknolojiler.

## 1.4 Tanımlar ve Kısaltmalar

**API:** Application Programming Interface, istemci-sunucu arasındaki veri alışverişini sağlar.

**UI:** User Interface, kullanıcıların sistemle etkileşim kurduğu grafiksel arayüz.

**DB:** Database, verilerin saklandığı yapı.

## 2. Sisteme Genel Bakış

Mahalle, apartman sakinleri ve yöneticileri arasındaki iletişimi modernize eden ve yönetim süreçlerini kolaylaştıran bir dijital platformdur. Sistem, sakinlerin günlük ihtiyaçlarını hızla karşılamalarını sağlarken, yöneticilere operasyonel verimlilik sunar.

### Temel Özellikler:

- Dijital apartman panosu: Günlük duyuru ve etkinliklerin paylaşımı.
- Anonim mesajlaşma: Sakinlerin kimlik gizliliği ile mesaj iletmesi.
- Merkezi kontrol paneli: Yöneticilere sakin listesi, daire atama ve şikayet yönetimi gibi araçlar sunar.
- Kullanıcı profilleri: Sakinler kişisel bilgilerini görüntüleyebilir.
- Etkinlik ve servis yönetimi: Yönetici ve sakinlerin etkinliklere ve hizmetlere erişimini sağlar.

**Avantajlar:** Kullanıcı dostu arayüzüyle sistem, güvenlik ve gizliliği korurken iletişim süreçlerini hızlandırır. Yöneticilere zaman tasarrufu sunar ve şeffaf bir yönetim deneyimi sağlar. Mahalle, yaşam kalitesini artıran yenilikçi bir çözümdür.

## 3. Sistem Mimarisi

### 3.1 Mimari Tasarım

Sistem istemci-sunucu tabanlı bir mimariye sahiptir ve üç ana bileşenden oluşmaktadır:

#### A. Frontend

- Kullanıcıların sistemle etkileşim kurduğu arayüzdür.
- HTML, CSS, JavaScript ve Bootstrap kullanılarak geliştirilmiştir.
- **Öne Çıkan Sayfalar:**
  - index.html: Ana sayfa ve genel bilgilendirme.
  - signup.html ve login.html: Kayıt ve giriş işlemleri.
  - profil.html: Kullanıcı profil görüntüleme.
  - control.html: Yönetici kontrol paneli.
  - eventPage.html ve technicService.html: Etkinlik ve hizmet listeleri.
  - Actuel.html: Günlük duyurular ve mesajlar.

## B. Backend

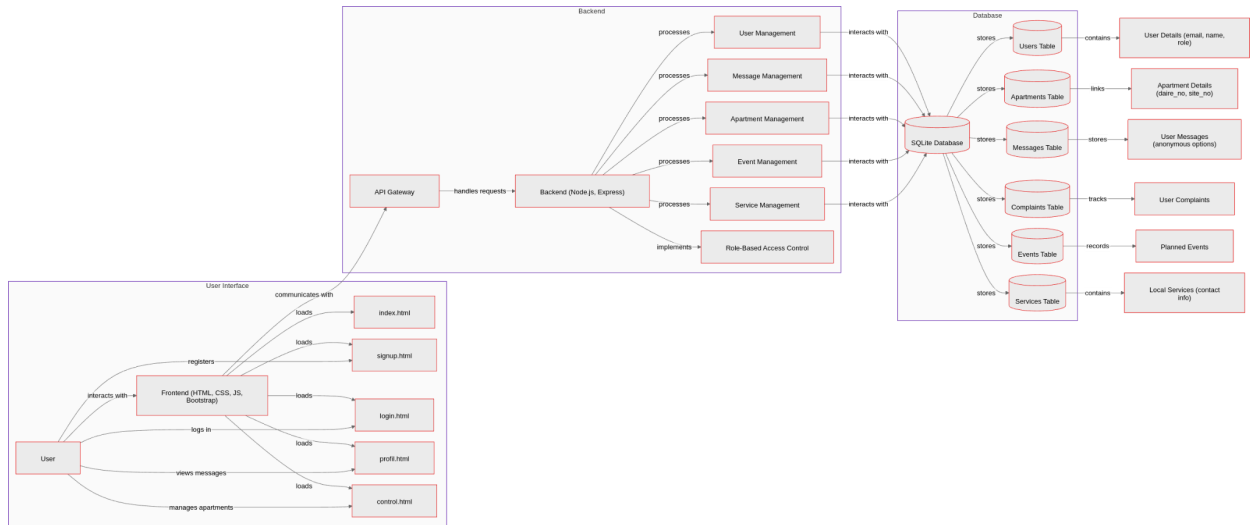
- Node.js ve Express framework ile geliştirilmiştir.
- İş mantığını ve veri işleme süreçlerini yönetir.
- **Temel İşlevler:**
  - **API Gateway:** Kullanıcı isteklerini alır ve ilgili işleme yönlendirir.
  - **Rol Tabanlı Yetkilendirme:** Kullanıcı rollerine göre erişim sağlar.
  - **Veri İşleme Modülleri:**
    - Kullanıcı, mesaj, daire, etkinlik ve hizmet yönetimi.

## C. Database

- SQLite kullanılarak kalıcı veri saklama sağlanır.
- **Temel Tablolar:**
  - **Users:** Kullanıcı bilgileri.
  - **Apartments:** Daire ve kullanıcı eşleştirmeleri.
  - **Messages:** Kullanıcı mesajları.
  - **Complaints:** Şikayet ve öneriler.
  - **Events:** Etkinlik detayları.
  - **Services:** Hizmet sağlayıcı bilgileri.

## Veri Akışı

- Kullanıcı, Frontend arayüzünden bir talepte bulunur.
- Backend, bu talebi işler ve Database'den gerekli veriyi alır.
- İşlenen veri, kullanıcıya Frontend üzerinden gösterilir.



### İşleyiş:

1. Kullanıcılar, frontend aracılığıyla API istekleri gönderir.
2. Backend, bu istekleri işleyerek gerekli veriyi veritabanından çeker.
3. Frontend, backend'den gelen veriyi kullanıcıya sunar.

### 3.2 Mimari Bileşenlerin Açıklanması

**Node.js:** Backend geliştirme için kullanılan framework.

**SQLite3:** Yerel ve hafif bir veritabanı çözümü.

**Bootstrap:** Modern, duyarlı ve kullanıcı dostu tasarımlar için kullanılan frontend kütüphanesi.

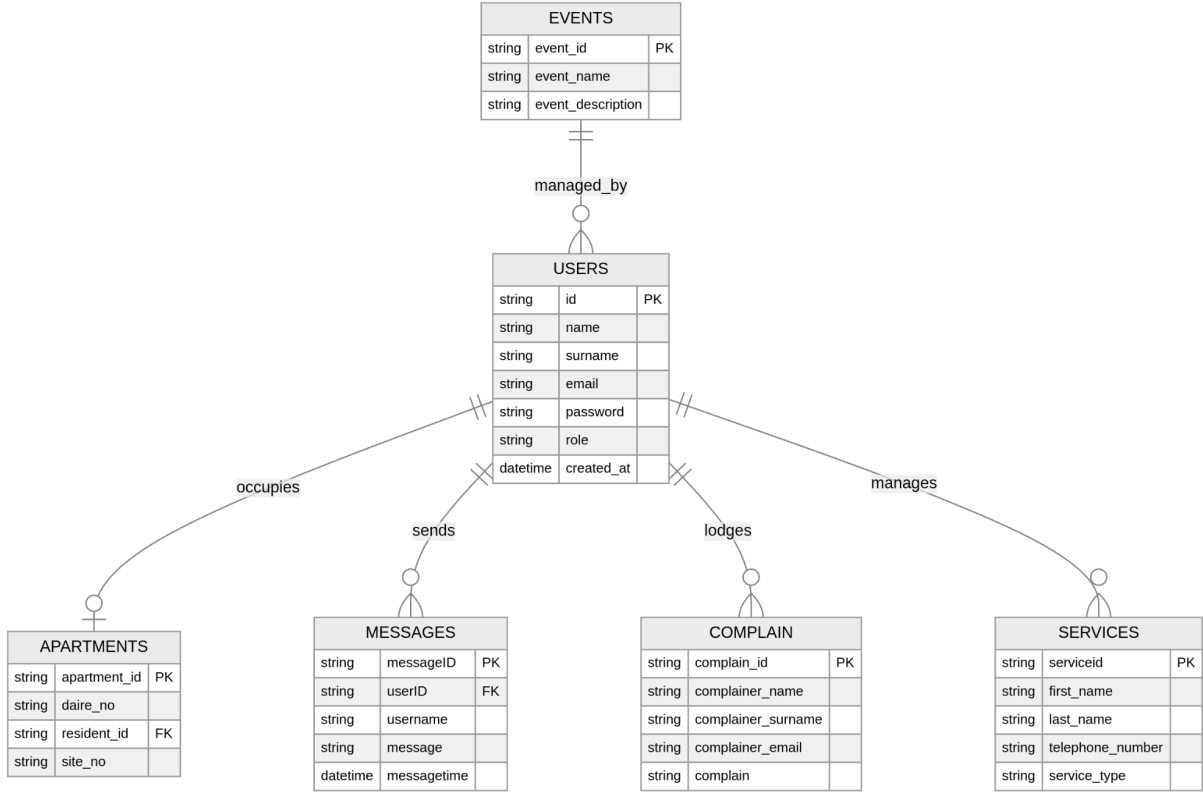
## 4. Veri Tasarımı (Data Design)

### 4.1. Veri Belirtimi

Sistem verileri şu şekilde organize edilmiştir:

- **users:** Kullanıcı bilgileri.
- **apartments:** Daire numaraları ve sakinlerin bağlantı bilgileri.
- **messages:** Kullanıcıların mesaj geçmişi (anonim mesajlaşmayı da içerir).
- **services:** Teknik servis bilgileri.
- **complain:** Kullanıcı şikayetleri.
- **event:** Etkinlik detayları.

## 4.2 Veri Tabanı Tasarımı



## 5. Kullanıcı Arayüzleri

### 5.1 Kullanıcı Arayüzlerine Genel Bakış

#### Sistem Kullanıcıları (Actors)

##### 1. Resident (Sakin):

Sistemin genel kullanıcılarıdır. Sistemi kullanarak:

- Etkinlikleri görüntüleyebilir,
- Şikayet gönderebilir (anonim veya isimle),
- Teknik servis bilgilerine erişebilir,
- Profil bilgilerini görüntüleyebilir ve güncelleyebilir,
- Mesajlaşma sistemi üzerinden duyuruları takip edebilir.

##### 2. Manager (Yönetici):

Sistemin yöneticisidir ve sakinlere ek olarak şu yetkilere sahiptir:

- Apartman sakinlerini yönetme,
- Apartman bilgilerini düzenleme ve daire atama,
- Teknik servis bilgilerini ekleme/silme,
- Şikayetleri ve etkinlikleri düzenleme veya silme.



## Kullanım Senaryoları (Use Cases)

### 1. Giriş ve Kayıt:

- **Actors:** Residents, Managers.
- **Açıklama:** Kullanıcılar giriş yapabilir veya yeni hesap oluşturabilir. Hatalı girişlerde sistem uygun hata mesajı verir.

### 2. Profil Görüntüleme:

- **Actors:** Residents, Managers.
- **Açıklama:** Kullanıcılar, profil.html üzerinden isim, soyisim, e-posta ve apartman bilgilerini görüntüler.

### 3. Şikayet Gönderme ve Görüntüleme:

- **Actors:** Residents (Gönderme), Managers (Görüntüleme ve Silme).
- **Açıklama:** Resident'lar isimli veya anonim şikayet gönderebilir. Yöneticiler ise şikayetleri görüntüleyebilir ve silebilir.

### 4. Etkinlik Yönetimi:

- **Actors:** Residents, Managers.
- **Açıklama:** Tüm kullanıcılar etkinlikleri görüntüleyebilir. Yöneticiler etkinlikleri oluşturabilir veya silebilir.

### 5. Teknik Servis Yönetimi:

- **Actors:** Managers.
- **Açıklama:** Yöneticiler teknik servis bilgilerini ekleyebilir ve güncelleyebilir. Tüm kullanıcılar bu bilgileri görüntüleyebilir.

### 6. Apartman Yönetimi:

- **Actors:** Managers.
- **Açıklama:** Yöneticiler, sakinlere daire atayabilir ve apartman bilgilerini düzenleyebilir.

### 7. Mesajlaşma Sistemi:

- **Actors:** Residents, Managers.
- **Açıklama:** Tüm kullanıcılar mesaj panosunda duyuruları takip edebilir. Bütün kullanıcılar mesaj gönderebilir, sakinler anonim olabilir.

## Sistem İlişkileri ve Hiyerarşisi

### 1. Kullanıcı Hiyerarşisi:

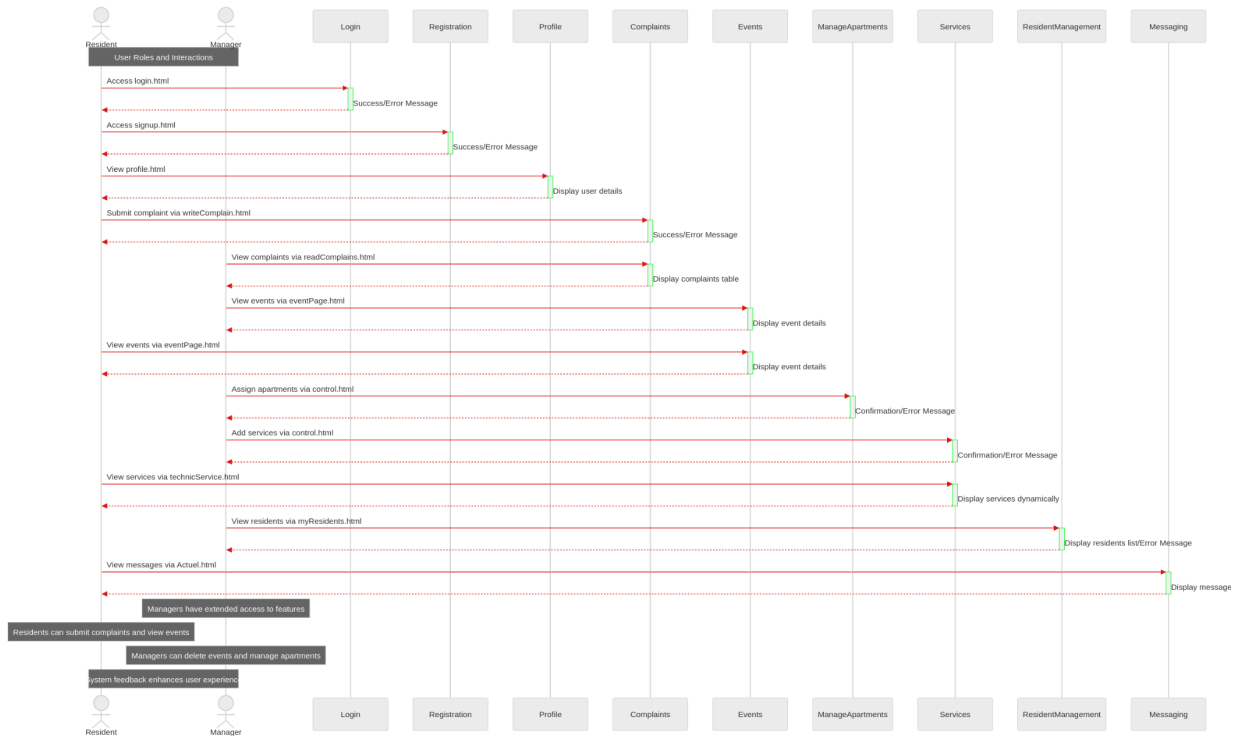
- Manager, Resident'ın özelliklerini devralır ve ek yetkilere sahiptir.

### 2. Sistem Geri Bildirimleri:

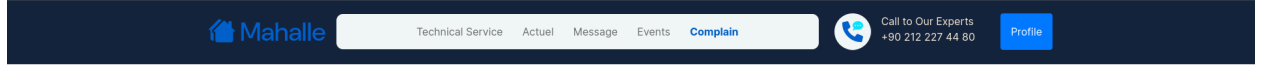
- Tüm işlemler (şikayet gönderme, etkinlik ekleme, mesajlaşma) başarı veya hata durumunda kullanıcılara bildirilir.

### 3. Kapsam ve Modüler Yapı:

- Sistemin modüler tasarımı, tüm kullanıcıların farklı ihtiyaçlarını karşılamaya odaklanmıştır.



## 5.2 Ekran Resimleri



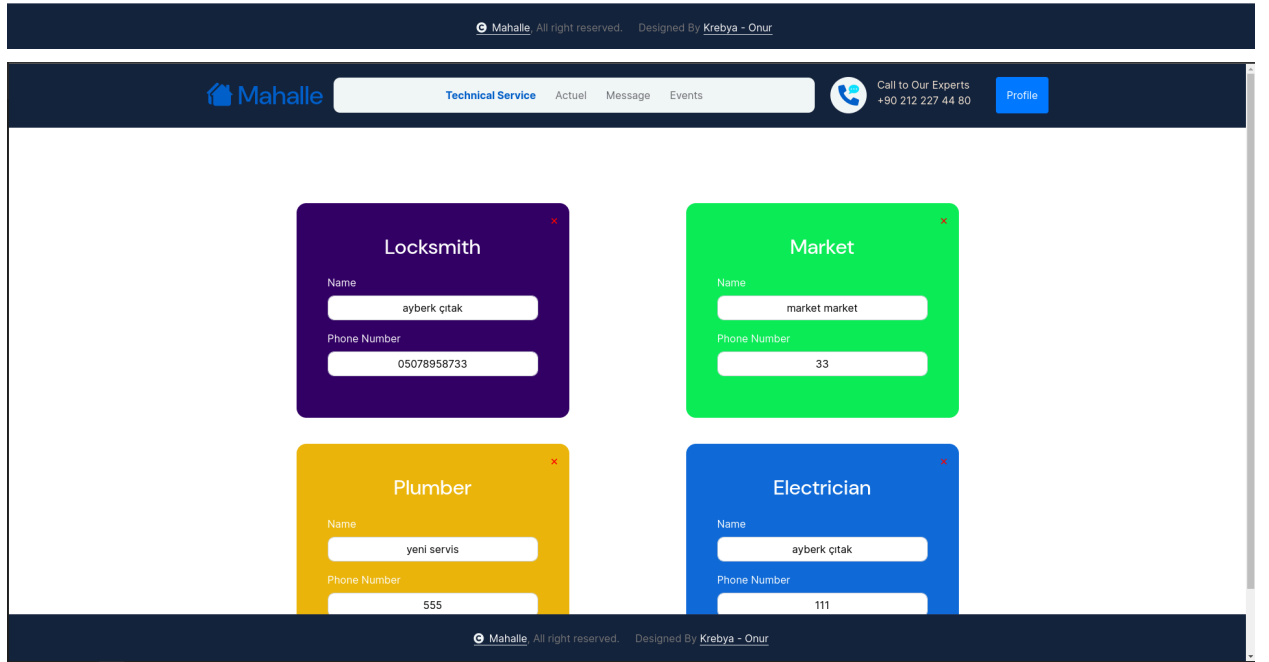
Write Your Complain or Suggestion

Post As:  
☒ Named ☐ Anonymous

Your Complain or Suggestion

Type your message here...

Send



Create Your Account

Email

Name

Surname


Password

Sign Up


Login

My Residents

Name	Surname	Email	Daire No
manager	manager	manager@manager.c	8
Jane	Smith	jane.smith@example.com	10
Michael	Johnson	michael.johnson@example.com	11
Emily	Davis	emily.davis@example.com	12
Chris	Brown	chris.brown@example.com	13
Jessica	Miller	jessica.miller@example.com	14
David	Wilson	david.wilson@example.com	15
Sarah	Moore	sarah.moore@example.com	16
Daniel	Taylor	daniel.taylor@example.com	17




Technical ServiceActuelMessageEvents

Call to Our Experts  
+90 212 227 44 80


Profile

All Complaints and Suggestions

Name	Surname	Email	Complaint	Action
Anonymous	Anonymous	Anonymous	oppa	X
ayberk	çitak	1234.ayberk@gmail.com	ayberk denme yap	X



Technical ServiceActuelMessageEventsComplain

Call to Our Experts  
+90 212 227 44 80

Profile

Post Your Message


Post As:

☒ Named ☐ Anonymous


Your Message

Type your message here...

Post

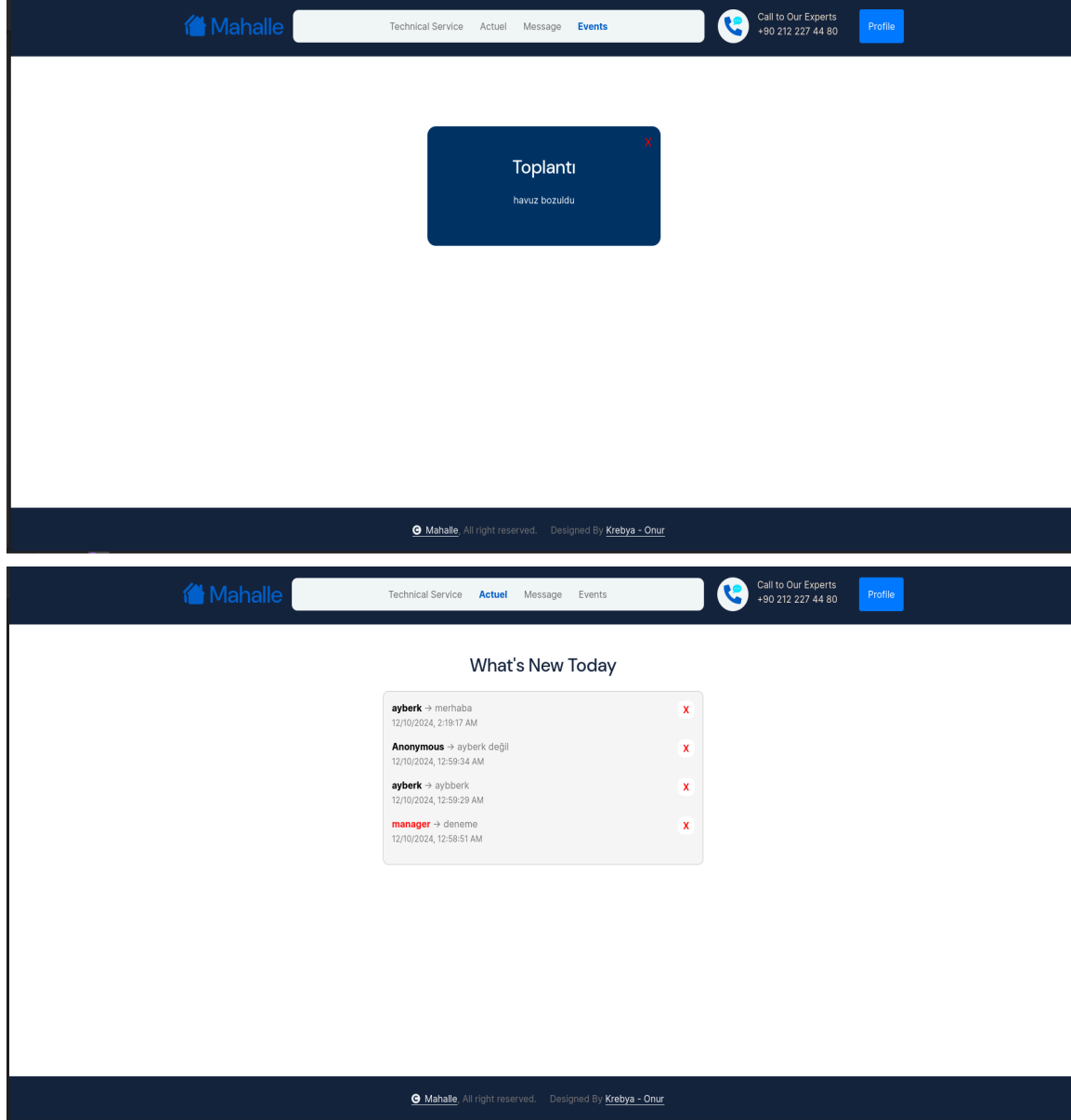


Technical ServiceActuelMessageEventsComplain

Call to Our Experts  
+90 212 227 44 80

Profile





## 6. Kullanılan Kütüphaneler ve Araçlar

**Node.js:** Backend geliştirme için.

**SQLite3:** Verilerin taşınabilir ve hafif bir şekilde saklanması için.

**Bootstrap:** Duyarlı tasarım standartlarını destekler.

**jQuery:** Frontend'deki dinamik işlemleri kolaylaştırır.

**WOW.js:** Kullanıcı deneyimini artırmak için animasyonlar sağlar.