

**Hasan Ferdi Turgutlu Teknoloji**  
 **Fakültesi**

**Yazılım Mühendisliği Bölümü**

**Online Etkinlik Rezervasyon Sistemi**  **Tasarımı ve Uygulaması**

**YZM 4101 – Profesyonel Yazılım**  **Geliştirme**

**Proje takım üyesi - Atakan KAYA**

**Proje takım üyesi - Efecan Şenol**

**Danışman - Dr. Öğr. Üyesi Fatih YÜCALAR**

**Manisa, 2024**

**Manisa Celal Bayar Üniversitesi**

**Hasan Ferdi Turgutlu Teknoloji Fakültesi**

**Yazılım Mühendisliği Bölümü**

Manisa Celal Bayar Üniversitesi Hasan Ferdi Turgutlu Teknoloji Fakültesi Yazılım Mühendisliği Bölümü öğrencileri Atakan kaya ve Efecan Şenol’un “Online Etkinlik Rezervasyon Sistemi Tasarımı ve Uygulaması” başlıklı mezuniyet projesi raporu   
……/……./……… tarihinde sunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur  
  
  
  
**Jüri Üyesi İmza**

**ÖNSÖZ**Bu tez çalışması, online etkinlik rezervasyon sistemleri üzerine yapılan bir araştırmanın ürünüdür. Bu süreçte, etkinlik düzenleme ve katılımcıların rezervasyon yapabilmesi gibi önemli bir sorunu ele almaya odaklandık. Online etkinlik rezervasyon sistemleri, günümüzde etkinlik sektöründe hızla yaygınlaşan ve büyük bir öneme sahip olan bir alan haline gelmiştir. Bu sistemin sağladığı avantajlar hem etkinlik düzenleyicileri hem de katılımcılar için birçok fırsat sunmaktadır.

Bu tez çalışmamızı tamamlamamızda bize yardımcı olan herkese teşekkür etmek istiyoruz. Özellikle danışmanınımız olan Dr.Öğr.Üyesi Fatih Yücalar Bey'e sonsuz teşekkürlerimizi sunuyoruz. Bize rehberlik ettiği, değerli önerileriyle çalışmamızı zenginleştirdiği ve motivasyonumuzu her zaman yüksek tuttukları için kendilerine minnettarız. Ayrıca Yazılım mühendisliği bölümümüzün değerli hocalarının bu zamana kadar bize kattıkları her şey için teşekkürü borç biliriz.

**İÇİNDEKİLER**

**ÖNSÖZ**3

**İÇİNDEKİLER** 4

**TABLO LİSTESİ** 5

**ŞEKİL LİSTESİ** 6

**ÖZET**7

1. GİRİŞ8

2.BÖLÜM 2: PROJE GEREKSİNİMLERİ

2.1 **Literatür Araştırması** 9

2.2 **İşlevsel Gereksinimler** 12

2.3 **İşlevsel Olmayan Gereksinimler** 15

3. PROJE ANALİZİ17

3.1. Veri Sözlüğü17

3.2. **İş Modeli** 20

3.2.1.**Use-Case’lerin kısa tanımları ve Aktörler** 20

3.2.2 **Use-Case Diyagramları** 22

3.2.3Use-Case Senaryoları ve İş Birliği Diyagramları 25

3.3. Yazılım Proje Yönetim Planı34

3.3.1 GanttÇizelgesi 34

3.3.2 Proje Risk Matrisi 35

3.3.3 Kullanılacak Altyapı ve Özellikler37

3.3.4 Sistem Gereksinimleri37

4. MİMARİ TASARIM 38

4.1 Mimari Tasarım41

4.1.1 Sistem Mimarisi41

4.1.2 Modüller43

4.1.3 Veri Tabanı Mimarisi 50

4.2 Detaylı Tasarım56

4.2.1 Sınıf Tasarımı59

5.GERÇEKLEŞTİRİM62

6. SONUÇ 84

7. REFERANSLAR85

# 

# **TABLO LİSTESİ**

[Tablo 1 Veri Sözlüğü-1 1](#_Toc29929954)8

[Tablo 2 Veri Sözlüğü-2 1](#_Toc29929954)9

[Tablo 3 Kayıt Olma Senaryosu](#_Toc29929955) 26

[Tablo 4 Giriş Yapma Senaryosu 2](#_Toc29929956)7

[Tablo 5 Etkinlik Oluşturma Senaryosu 2](#_Toc29929957)7

[Tablo 6 Etkinliğe katılma Senaryosu 2](#_Toc29929958)8

[Tablo 7 Mekan Ekleme Senaryosu 2](#_Toc29929959)8

[Tablo 8 Puan Verme Senaryosu 2](#_Toc29929960)9

[Tablo 9 Mekan Sahibi Profil Güncelleme Senaroyosu 2](#_Toc29929961)9

[Tablo 10 Etkinlik Güncelleme Senaryosu 30](#_Toc29929962)

[Tablo 11 Kullanıcı İstatistikleri Görüntüleme Senaryosu](#_Toc29929963) 30

[Tablo 12 Geri Bildirim Gönderme Senaryosu 3](#_Toc29929964)1

[Tablo 13 Ödeme İşlemi Senaryosu 3](#_Toc29929965)2

[Tablo 14 Yetkilendirme Ve Kullanıcı Yönetimi Senaryosu 32](#_Toc29929966)

[Tablo 15 Veritabanı Mekan Tablosu 55](#_Toc29929967)

[Tablo 16 Veritabanı Kullanıcı Tablosu 56](#_Toc29929968)

# 

# **ŞEKİL LİSTESİ**

[Şekil 1 Müşteri Aktörünün use-case diyagramı 23](#_Toc29929223)

[Şekil 2 Mekan Sahibi Aktörünün use-case diyagramı 24](#_Toc29929224)

[Şekil 3 Sistem Yöneticisi Aktörünün use-case diyagramı 25](#_Toc29929225)

[Şekil 4 İş Birliği Diyagramı 34](#_Toc29929226)

[Şekil 5 Gantt Çizelgesi 35](#_Toc29929227)

[Şekil 6 Proje Risk Matrisi 36](#_Toc29929228)

[Şekil 7 Sistem Mimarisi 42](#_Toc29929229)

[Şekil 8 Sistem Mimarisi 2 43](#_Toc29929230)

[Şekil 9 Kullanıcı Yönetim Modülü 44](#_Toc29929231)

[Şekil 10 Mekan Sahibi Yönetim Modülü 45](#_Toc29929232)

[Şekil 11 Etkinlik Yönetimi Modülü 46](#_Toc29929233)

[Şekil 12 Mekan Ve Değerlendirme Modülü 47](#_Toc29929234)

[Şekil 13 Başlangıç Sınıf Diyagramı 49](#_Toc29929235)

[Şekil 14 Veritabanı Mimarisi 51](#_Toc29929236)

[Şekil 15 Veritabanı Mimarisi 2 54](#_Toc29929237)

[Şekil 16 Veritabanında Kullanıcı Sınıfı 57](#_Toc29929238)

[Şekil 17 Veritabanında Etkinlik Sınıfı 58](#_Toc29929239)

[Şekil 18 Veritabanında Mekan Sahibi Sınıfı 59](#_Toc29929240)

[Şekil 19 Genel Sınıf Tasarımı 61](#_Toc29929241)

[Şekil 20 Giriş Yapma Ekranı 66](#_Toc29929242)

[Şekil 21 Kayıt Olma Ekranı 67](#_Toc29929243)

[Şekil 22 Şifremi unuttum Ekranı 67](#_Toc29929244)

[Şekil 23 Mekan Sahibi Ana Ekranı 68](#_Toc29929245)

[Şekil 24 Organizatör Etkinlik Görüntüleme Ekranı 69](#_Toc29929246)

[Şekil 25 Organizatör Etkinlik Güncelleme Ekranı 70](#_Toc29929247)

[Şekil 26 Organizatör Yeni Etkinlik Ekleme 71](#_Toc29929248)

[Şekil 27 Organizatör Kullanıcı Listesi Görüntüleme Ekranı 73](#_Toc29929249)

[Şekil 28 Organizatör Kullaıcı Detay Ekranı 74](#_Toc29929250)

[Şekil 29 Organizatör Etkinlik Yorumları Görüntüleme Ekranı 75](#_Toc29929251)

[Şekil 30 Kullanıcı Profil Ekranı 76](#_Toc29929252)

[Şekil 31 Kullanıcı Profil Güncelleme Ekranı 77](#_Toc29929253)

[Şekil 32 Kullanıcı Şifre Güncelleme Ekranı 78](#_Toc29929254)

[Şekil 33 Kullanıcı Etkinlik Görüntüleme Ekranı 79](#_Toc29929255)

[Şekil 34 Kullanıcı Etkinlik Filtreleme Ekranı 80](#_Toc29929256)

[Şekil 35 Kullanıcı Etkinlik Detay Ekranı 81](#_Toc29929257)

[Şekil 36 Kullanıcı Sipariş Oluşturma Ekranı 82](#_Toc29929258)

[Şekil 37 Kullanıcı Kargo Bilgileri Ekranı 82](#_Toc29929259)

[Şekil 38 Kullanıcı Sipariş Onaylama Ekranı 83](#_Toc29929260)

[Şekil 39 Kullanıcı Ödeme Ekranı 84](#_Toc29929261)

[Şekil 40 Kullanıcı Sipariş Geçmişi Ekranı 84](#_Toc29929262)

**ÖZET**

Bu tez çalışması, web ve mobil platformlar için online etkinlik rezervasyon sistemi tasarımı ve uygulamasını ele almaktadır. Geliştirilecek sistem, etkinlik düzenleyicileri ve sanatçıların işlemlerini takip etmek, etkinlik organizasyonlarını daha verimli hale getirmek ve kullanıcıların rezervasyon sürecini kolaylaştırmak amacıyla geliştirilmektedir.

Bu projenin temel hedefi, etkinlik organizatörlerinin kendisine ait uygun mekanlar da etkinlik oluşturabilmelerini ve bu mekanlarda varsa aşçı, garson gibi detayları belirterek ayarlayabilmelerini sağlamaktır. Ayrıca, etkinlik organizatörleri etkinliğe katılacak müşterilerine özel etkinlikler düzenleyebilir, müşterilerde oluşturulan bu etkinliklere katılabilirler . Bu sayede, süreç hem mekân sahipleri hem de müşteriler için daha şeffaf ve anlaşılır olacaktır. Aynı zamanda, sanatçıların kendilerini sisteme ekleyebileceği ve davet edildikleri mekanlara katılabileceği ve etkinlik katılımcıları tarafından puanlanabileceği bir yapı da sisteme entegre edilecektir.

Bu tez çalışması, analiz, tasarım ve gerçekleştirme aşamalarını içermektedir. Analiz aşamasında her bir kullanım durumunun tanımı, normal ve istisnai sonuçları detaylı bir şekilde incelenecektir. Ayrıca, her kullanım durumu senaryosu için UML diyagramları çizilecektir. Üst seviye tasarım aşamasında ise genel sistem mimarisi çizilecek ve gereken modüller belirlenecektir. Gerçekleştirme aşamasında ise danışman tarafından belirlenen modüller için birim testleri gerçekleştirilecektir.

**BÖLÜM 1: GİRİŞ**

Günümüzde internetin hızla yaygınlaşması ve mobil cihazların kullanımının artmasıyla birlikte, çeşitli etkinlikler ve etkinliklere katılım talepleri de önemli bir ivme kazanmıştır. Konserler, tiyatro oyunları, spor etkinlikleri ve diğer benzer etkinlikler gibi çeşitli kültürel ve eğlence aktiviteleri, insanların zamanlarını keyifli bir şekilde geçirmelerini sağlar. Ancak, bu etkinliklere katılmak veya rezervasyon yapmak bazen zorlu bir süreç haline gelebilir.

Geleneksel rezervasyon sistemleri sınırlı saatlerde çalışabilir ve fiziksel bir mekanın önünde uzun kuyruklara neden olabilir. Ayrıca, kullanıcıların telefonla rezervasyon yapması gereken durumlarda, yoğun talep nedeniyle telefon hatlarının meşgul olması gibi sorunlar ortaya çıkabilir. Tüm bu sorunlar, etkinliklere katılmak isteyen kullanıcıların zamanlarını ve enerjilerini harcamalarına yol açar ve etkinlik düzenleyicileri için de verimsiz bir süreç oluşturur.  
  
Günümüzde dijitalleşmenin de etkisi ile rezervasyon sistemlerine olan ilgi de gitgide artmıştır. Rezervem adlı internet sitesinin bir yazısın da (Rezervem, 2021)[1]denilerek bu konunun önemini vurgulamıştır.

Web ve mobil platformlarının hızla gelişmesiyle birlikte, online etkinlik rezervasyon sistemleri bu sorunları ortadan kaldırmak ve kullanıcı deneyimini iyileştirmek için önemli bir araç haline gelmiştir. Online etkinlik rezervasyon sistemleri, kullanıcılara istedikleri etkinliklere kolayca erişim sağlar, rezervasyon yapmalarına ve biletlerini hızlıca alıp ödemelerini gerçekleştirmelerine olanak tanır. Ayrıca, etkinlik düzenleyicileri için de yönetim süreçlerini kolaylaştırır ve verimliliklerini artırır.

Bu tez çalışması, online etkinlik rezervasyon sistemi tasarımı ve uygulaması konusunda birçok avantaj ve fayda sunmayı amaçlamaktadır. Günümüzde etkinlik düzenleyicileri, mekân sahipleri ve müşteriler arasında iletişim ve koordinasyon süreçleri genellikle karmaşık ve zaman alıcı olabilmektedir. Bu durum, etkinliklerin düzenlenmesi ve planlanmasında bazı sorunlara yol açabilmektedir.

Bu çalışma ile, online etkinlik rezervasyon sistemi ile bu sorunlara çözüm sağlamayı hedeflemektedir. Geliştirilen sistem, etkinlik düzenleyicilerinin mekanlarında istedikleri organizasyonu bir sanatçı davetiyle oluşturmalarını ve müşterilerine sunmayı amaç edinmektedir. Ayrıca, etkinlik açıklama kısmında detaylı iletişim numaraları bırakarak müşteri taleplerine göre yemek ve içecek gibi ek hizmetlerin kolaylıkla önceden bilgilendirmesi yapılacaktır. Sanatçılar da sisteme dahil olarak, davet edildikleri mekanlarda performans sergileyebilme imkânı bulmaktadır.

Bu proje kapsamında gerçekleştirilen analiz, tasarım ve uygulama aşamaları, online etkinlik rezervasyon sisteminin başarılı bir şekilde geliştirilmesini sağlamaktadır. Her bir use-case senaryosu, UML diyagramları ve dokümantasyonlar dayesinde sistemin işleyişi ayrıntılı bir şekilde belgelenmiştir. Bu dokümantasyonlar, sistemin gelecekteki bakım ve geliştirme çalışmaları için de önemli bir referans kaynağı oluşturmaktadır.

Bu tez çalışması, web ve mobil platformları için online etkinlik rezervasyon sisteminin tasarımı ve geliştirilmesi üzerine odaklanmaktadır. Geliştirilen sistem, etkinlik düzenleyicileri, mekân sahipleri ve sanatçılar arasındaki iletişim ve koordinasyonu kolaylaştırmayı amaçlamaktadır. Ayrıca, etkinlik düzenleme sürecini daha hızlı, şeffaf ve kullanıcı dostu hale getirmektedir. Amacı, kullanıcıların etkinliklere erişimini kolaylaştıran, rezervasyon sürecini optimize eden ve kullanıcı deneyimini geliştiren bir sistem oluşturmaktır. Bu sistem, kullanıcıların istedikleri etkinlikleri kolayca bulmalarını, rezervasyon yapmalarını ve biletlerini güvenli bir şekilde alıp ödemelerini gerçekleştirmelerini sağlayacaktır.

Bu çalışmanın sonucunda, online etkinlik rezervasyon sistemi sayesinde birçok avantaj elde edileceği görülmektedir. Etkinlik düzenleyicileri mekanlarında organizasyonlarını daha iyi planlayabileceklerdir. Müşterilerin talepleri doğrultusunda ek hizmetlerin kolaylıkla eklenmesi, müşteri memnuniyetini artıracaktır. Sanatçılar da sisteme dahil olarak, daha fazla mekânda performans sergileyebileceklerdir.

**BÖLÜM 2: PROJE GEREKSİNİMLERİ**

Bu bölüm de Online Etkinlik Rezervasyon Sistemi Tasarımı ve Uygulaması için proje gereksinimleri kısmını oluşturacağız. Bu doğrultuda Literatür ve Pazar Araştırmasına yönelik sonuçları inceleyeceğimiz “Literatür Araştırması” bölümü, yazılımın hangi işlemleri yapacağını, hangi girdileri alıp hangi çıktıları üreteceğini belirlediğimiz “İşlevsel Gereksinimler” bölümü ve yazılımın yanıtlama zamanı (response time), güvenilirlik (reliability) ve performans gibi sahip olması gereken özelliklerini belirlediğimiz “İşlevsel Olmayan Gereksinimler” bölümünü inceleyeceğiz.

* **BÖLÜM 2.1: Literatür Araştırması**

Bu kısımda "Online Etkinlik Rezervasyon Sistemi Tasarımı ve Uygulaması" konusunda yapılan araştırmaları ve mevcut literatürü derlemeyi amaçlamaktadır. Araştırma kapsamında, etkinlik yönetimi bilgi sistemleri, teknolojik çözümler ve pazar analizi gibi konular incelenmiştir.

Gerçekleştireceğimiz proje için literatür araştırmamıza ilk olarak 2019 Winter Simulation Conference (WSC) de yayınlananan “ Modeling and Simulation of the Bottlenecks of an Online Reservation System” den yapacağız.  
  
incelenen bu çalışma, bir çevrimiçi rezervasyon sistemi için bir analitik ve simülasyon modeli sunmayı ve darboğazlarıyla ilgili performansını incelemeyi amaçlamaktadır.

Rezervasyon sistemlerinin modellenmesi ve değerlendirilmesi, bu modeller mevcut olduğunda analitik modeller aracılığıyla yapılmalıdır. Bununla birlikte, genel olarak, bu tür modellerin gerçek dünyanın çeşitli ayrıntılarını dikkate almayan yaklaşımlara sahip olduğu bilinmektedir.

Söz konusu çoklu işlem sistemi, veritabanı sorguları kullandığından (ve ayrıca ön alım sürdürme önceliğini kullandığından) ve belirli bir karmaşıklık derecesine sahip olduğundan, yaklaşımımız, ayrık olay simülasyon modeli (ve tersi) tarafından doğrulanan yaklaşık bir analitik model kullanır.

Her iki model de gerçek dünyadaki bir sistemin ölçümlerinden elde edilen verilerle desteklenir. Doğrulandıktan sonra model, simülasyon yoluyla diğer girdi verileri ve dağılımlar için analiz edilebilir.

Bilgilerini bize sunmaktadır.

Sıradaki araştırmamızı 2022 1st International Conference on Computational Science and Technology (ICCST) de yayınlanan “Online College Event-Hall Booking Reservation System” den yapacağız.

İncelenen bu çalışma sistem salon yönetim yazılımı için bir projedir. Bir mekânı etkili bir şekilde yönetmek için gereken tüm araçlara ve yeteneklere sahiptir.

Tüm sistem işlemlerini yönetmek ve denetlemek için kullanılan bir yönetici hesabına sahiptir. Teknoloji hem ön rezervasyonları hem de salonun durumunu izler. Sistem, kullanıcı iletişim bilgileri, ilgili etkinlik bilgileri ve salon rezervasyonları ile ilgili bilgileri içeren bir veri tabanı tutar. Sistem GUI'si, yöneticinin salonlar için rezervasyonları ve zamanlamaları hızlı bir şekilde görüntülemesine olanak tanır. Teknoloji, yeni bir etkinlik zamanı yaklaştığında bildirimleri de sağlar.

Her departman, salonu kendi ihtiyaçları için rezerve etmek için kullanabilecekleri bir kullanıcı adı ve şifre alır. Bu sayede müşteriler, belirli bir salon için uygun tarihleri ​​ve orada sunulan kapasite, mikrofon, projektör, beyaz tahta vb. Kapasite ve müsaitlik durumuna göre salon seçilecek. İstek listesi ilgili bölüm başkanı tarafından görüntülenebilir. Kullanıcı, bu oturum açma bilgilerini ve hatta e-posta adresini kullanarak kabul ve ret işlemlerine erişebilir.

Alttaki bölümde de pazar araştırması yapılacaktır.

Online etkinlik rezervasyon sistemleri, etkinlikleri düzenleyen kişilerin ve müşterilerin işlemlerini takip etmelerine olanak tanıyan web ve mobil tabanlı yazılımlardır. Bu pazar araştırması, online etkinlik rezervasyon sistemleri pazarının mevcut durumunu ve gelecekteki eğilimlerini incelemektedir.

* Pazar Büyüklüğü ve Büyüme Trendleri:
* Online etkinlik rezervasyon sistemleri pazarı, son yıllarda hızla büyümüştür. Küresel olarak, bu pazarın büyüklüğü sürekli artmaktadır ve öngörülen büyüme trendi önümüzdeki yıllarda da devam etmektedir. Özellikle partiler ve konserler gibi etkinlikler için online rezervasyon sistemleri büyük talep görmektedir.
* Rekabetçi Faktörler:
* Bu pazarda birçok rekabetçi firma bulunmaktadır. Eventbrite, Cvent, Ticketmaster, Eventzilla ve Bookitbee gibi firmalar, online etkinlik rezervasyon sistemleri alanında önde gelen oyunculardır. Rekabetçi faktörler arasında kullanıcı deneyimi, özelleştirme seçenekleri, güvenlik ve kullanıcı verilerinin korunması gibi etkenler yer almaktadır.
* Etkinlik Türleri ve Talep:
* Online etkinlik rezervasyon sistemleri, çeşitli etkinlikler için kullanılmaktadır. Partiler, konferanslar, festivaller, tiyatro gösterileri ve spor etkinlikleri gibi birçok farklı etkinlik türü için bu sistemlerin kullanımı yaygındır. Talep, etkinlik türüne ve coğrafi konuma bağlı olarak değişiklik göstermektedir.
* Müşteri Beklentileri:
* Müşteriler, online etkinlik rezervasyon sistemlerinden kullanıcı dostu arayüzler, kolay kullanılabilirlik, hızlı ve güvenilir işlem süreçleri, ödeme seçenekleri, bilet teslimat yöntemleri ve müşteri destek hizmetleri gibi faktörler beklemektedir. Ayrıca, kişiselleştirme seçenekleri ve özel indirimler gibi avantajlar da müşteri memnuniyetini artırabilmektedir.
* Müşteri Veri Güvenliği ve Gizlilik:
* Online etkinlik rezervasyon sistemlerinde müşteri verilerinin güvenliği ve gizliliği büyük önem taşımaktadır. GDPR (Genel Veri Koruma Yönetmeliği) gibi yasal düzenlemeler, müşteri verilerinin korunmasına ve izinsiz kullanımının engellenmesine yönelik yönergeler sunmaktadır. Bu nedenle, pazarın lider oyuncuları müşteri verilerini etkili bir şekilde korumak ve gizlilik politikalarını şeffaf bir şekilde sunmak zorundadır.
* Teknolojik Trendler:
* Online etkinlik rezervasyon sistemleri alanında teknolojik gelişmelerin etkisi büyük olmuştur. Yapay zekâ ve makine öğrenme gibi teknolojiler, sistemlerin daha akıllı ve özelleştirilebilir hale gelmesini sağlamıştır. Chat botlar, müşteri hizmetlerindeki etkileşimleri kolaylaştırarak kullanıcı deneyimini iyileştirmektedir.

Sonuç:

Online etkinlik rezervasyon sistemleri pazarı hızla büyüyen bir sektördür. Müşterilerin etkinliklerini kolayca rezerve etmelerini sağlayan bu sistemler, organizatörlere ve sanatçılara da işlemlerini daha etkin bir şekilde takip etme imkanı sunmaktadır. Teknolojik gelişmeler ve müşteri talepleri doğrultusunda, bu pazarda sürekli yenilikler ve iyileştirmeler gözlenmektedir. Güvenli veri yönetimi ve müşteri memnuniyeti, rekabetçi bir avantaj sağlamaktadır.

* **BÖLÜM 2.2: İşlevsel Gereksinimler**

Bu bölümde projemizin işlevsel gereksinimler kısmını inceleyeceğiz. Visure solutions sitesinde yapılan tanıma göre (VisureSolutions, 2019) [2]şeklinde verilmiştir.

Aşağıda sistemimizin işlevsel gereksinimleri belirtilmiştir.

* Kullanıcı Kaydı ve Oturum Açma:
* Kullanıcılar, sisteme kaydolabilmeli ve bir hesap oluşturabilmelidir.
* Kayıtlı kullanıcılar, kullanıcı adı ve parolalarıyla oturum açabilmelidir.
* Etkinlik Arama ve Filtreleme:
* Kullanıcılar, etkinlikleri etkinlik türü, etkinlik puanı, etkinlik ismi gibi kriterlere göre arayabilmelidir.
* Arama sonuçları, kullanıcıların tercihlerine göre filtrelenmelidir.
* Etkinlik Rezervasyonu:
* Kullanıcılar, seçtikleri etkinliği rezerve edebilmelidir.
* Rezervasyon işlemi sırasında kullanıcılar katılımcı sayısını ve diğer tercihleri belirtebilmelidir.
* Rezervasyon bileti sepete eklendikten sonra, kullanıcılar tarafından ödeme yapılabilmesi sağlanmalıdır.
* Etkinlik ekleme ve Özelleştirme:
* Etkinliği düzenleyecek organizatörler (mekan sahipleri), etkinliği katılacak sanatçıyı belirtebilecek ve mekan fotoğraflarını ekleyebilecektir.
* Mekân sahipleri, etkinlik ekleme sayfasında açıklama kısmından detaylarını paylaşabilecek ve hangi kategori de etkinlik düzenlediğini belirtebilecektir.
* Kullanıcılar, seçtikleri etkinlikleri isteklerine göre filtreleyip arayabilecektir.
* Sanatçı Seçimi:
* Organizatörler düzenleyecek olduğu etkinliklerine belirtebilecekler.
* Kullanıcılar etkinliklere baktığı sayfada hangi sanatçıya ait etkinlik olduğunu görebilecek.
* Sanatçı Profili ve Değerlendirme:
* Sanatçılar, kendilerini sisteme ekleyebilmeli ve profil bilgilerini girebilmelidir.
* Kullanıcılar, etkinlik sonrası sanatçıları değerlendirebilmeli ve yorum yapabilmelidir.
* Yönetim Arayüzü:
* Sistem yöneticileri ve admin yetkisi olan organizatörler, etkinlikleri yönetebilmeli, yapılan rezervasyonları görebilmeli ve mekan/sanatçı bilgilerini güncelleyebilmelidir.
* Yöneticiler, etkinliklerin ve rezervasyonların istatistiksel verilerine erişebilmelidir. Ayrıca, sisteme kayıtlı kullanıcılar üzerinde güncelleme ve değişiklikler yapabilmelidir.
* Kullanıcı Geri Bildirimi:
* Kullanıcıların katıldığı etkinlik ile ilgili görüşlerini yorum ve puan sistemi ile değerlendirip paylaşabilmesi.
* Organizatörlerin kullanıcılardan gelen geri bildirimleri etkinlik numarası sayesinde aratıp görüntüleyebilmesi.
* Otomatik Bildirimler:
* Kullanıcılara rezervasyon onayı, iptali veya güncellemeleri hakkında otomatik bildirimler gönderilmesi.
* Kullanıcılara etkinlik tarihleri, saatleri veya hatırlatmaları için bildirimler gönderilmesi.
* Bilet Yönetimi:
* Kullanıcıların rezervasyon yapılan etkinlikler için elektronik bilet alabilmesi.
* Kullanıcıların biletlerini görüntüleyebilmesi, iptal edebilmesi veya değiştirebilmesi.
* İşlem Geçmişi ve Raporlama:
* Kullanıcıların rezervasyon ve ödeme geçmişlerini görüntüleyebilmesi.
* Yöneticilerin sistem üzerindeki tüm işlemleri takip edebilmesi sağlanacak.
* Etkinlik Değişiklikleri ve İptalleri:
* Organizatörlerin oluşturdukları etkinliklerde değişiklik yapabilmesi
* Etkinlik iptalleri için kullanıcıların süreci başlatabilmesi ve geri ödeme taleplerinde bulunabilmesi.
* Ödeme Yöntemleri ve Güvenliği:
* Kullanıcılara arayüzü anlaşılır bir ödeme yönteminin sunulması.
* Ödeme işlemlerinin güvenliği için gerekli önlemlerin alınması
* Gerçek Zamanlı Etkinlik Rezervasyon Kontrolü:
* Mekân sahiplerinin veya organizatörlerin gerçek zamanlı olarak mevcut rezervasyonları kontrol edebilmesi.
* Mobil Uygulama Desteği:
* Online etkinlik rezervasyon sistemine erişim için mobil uygulama geliştirilmesi.
* Kullanıcıların etkinlikleri mobil uygulama üzerinden arama, rezervasyon yapma ve yönetme imkanına sahip olması.
* Etkinlik Değerlendirmeleri:
* Kullanıcıların etkinliklere ve hizmetlere yönelik puanlama ve değerlendirme yapabilmesi.
* Diğer kullanıcıların etkinlik değerlendirmelerini görebilmesi ve bu değerlendirmelere dayanarak tercih yapabilmesi
* Etkinlik İçerik Yönetimi:
* Etkinliği düzenleyen organizatörlerin etkinlik içeriklerini (açıklama, fotoğraf vb.) yönetebilmesi.
* Organizatörlerin oluşturdukları etkinlikleri sonradan güncelleyebilmeleri.
* Sanatçı Puan Sistemi:
* Sanatçıların davet edildiği etkinliklere katılıp katılmaması durumuna göre bir puan sistemi oluşturulacaktır.
* **BÖLÜM 2.3: İşlevsel Olmayan Gereksinimler**

Bu bölümde projemizin işlevsel olmayan gereksinimler kısmını inceleyeceğiz.”İşlevsel Olmayan Gereksinimler, ölçeklenebilirlik, sürdürülebilirlik, performans, taşınabilirlik, güvenlik, güvenilirlik ve daha pek çok sorunla ilgilenir. İşlevsel Olmayan Gereksinimler, yazılım sistemleri için hayati önem taşıyan kalite sorunlarını ele alır.” (VisureSolutions, 2019) [3]şeklinde verilmiştir.

Aşağıda sistemimizin işlevsel gereksinimleri belirtilmiştir.

* Performans Gereksinimleri:
* Sistem, kullanıcılara hızlı yanıt vermelidir.
* Yüksek trafikli durumlarda bile sistemin performansı etkilenmemelidir.
* Rezervasyon işlemleri ve diğer işlemler hızlı ve verimli bir şekilde gerçekleştirilmelidir.
* Kullanılabilirlik Gereksinimleri:
* Kullanıcı arayüzü basit, anlaşılır ve kullanıcı dostu olmalıdır.
* Kullanıcılar etkinlikleri kolaylıkla arayabilme ve filtreleme seçeneklerine sahip olmalıdır.
* Rezervasyon süreci, adım adım ve kullanıcıya rehberlik eden bir şekilde tasarlanmalıdır.
* Güvenlik Gereksinimleri:
* Kullanıcıların kişisel verileri güvenli bir şekilde saklanmalı ve korunmalıdır.
* Ödeme işlemleri ve finansal bilgiler güvenli bir şekilde işlenmelidir.
* Sisteme yetkisiz erişimler engellenmelidir.
* Erişilebilirlik Gereksinimleri:
* Tüm kullanıcılar için erişilebilir bir kullanıcı arayüzü sağlanmalıdır.
* Özel ihtiyaçlara sahip kullanıcılar için uygun destek ve kolaylık sunulmalıdır.
* Kullanıcı Deneyimi Gereksinimleri:
* Kullanıcıların etkinlikleri kolaylıkla bulabilmesi için arama ve filtreleme seçenekleri sunulmalıdır.
* Rezervasyon süreci sade ve hızlı olmalı, gereksiz adımlardan kaçınılmalıdır.
* Kullanıcılara geri bildirim ve bildirimlerle ilgili düzenlemeler yapılmalıdır.
* Sistemin kullanıcı trafiği gereksinimleri:
* Birden fazla rezervasyon aynı anda gerçekleştiğinde sistem performansı etkilenmemelidir.
* Yeni kullanıcılar ve etkinlikler sisteme sorunsuz bir şekilde eklenebilmelidir.
* Veri Yedekleme ve Kurtarma Gereksinimleri:
* Sistem, düzenli aralıklarla kullanıcı ve etkinlik verilerini yedeklemelidir.
* Veri kaybı durumunda sistemin kurtarma mekanizması bulunmalıdır.
* Yedekleme ve kurtarma süreçleri otomatik olarak gerçekleştirilmelidir.
* Entegrasyon Gereksinimleri:
* Ödeme işlemleri için güvenilir ve uyumlu bir ödeme ağ geçidi entegrasyonu sağlanmalıdır.
* Uyumluluk Gereksinimleri:
* Sistem, farklı web tarayıcılarında ve mobil platformlarda sorunsuz çalışmalıdır.
* Farklı ekran boyutları ve çözünürlüklerine uyumlu olmalıdır.
* Sürdürülebilirlik Gereksinimleri:
* Sistem, uzun vadeli kullanım için sürdürülebilir bir altyapıya sahip olmalıdır.
* Güncelleme ve bakım işlemleri kolaylıkla gerçekleştirilebilmelidir.
* Sistem, gelecekteki teknolojik gelişmelere ve değişikliklere uyum sağlayabilecek esneklikte olmalıdır.
* Yasal uyumluluk gereksinimleri:
* Sistem, ilgili yasal düzenlemelere uygun olmalıdır. Kullanıcıların kişisel verilerin korunması konusunda yasal haklarına saygı gösterilmeli ve veri gizliliği politikalarına uymalıdır.

* **BÖLÜM 3: PROJE ANALİZİ**

Bu bölümde Online Etkinlik Rezervasyon Sistemi Tasarımı ve Uygulaması için proje analizi kısmını oluşturacağız. Bu analiz bölümünde uygulama sahasına yönelik teknik terimleri ve açıklamalarını içeren veri sözlüğü bölümü, projenin Use-case’leri, Aktörleri ve bu use-case’lere ilişkin senaryolar ayrıntılı olarak açıklandığı iş modeli bölümü ve proje risk matrisi, gantt çizelgesi, kullanılacak teknolojik altyapı , sistem gereksinimleri gibi kısımların olduğu yazılım proje yönetim planı kısmı ele alınacaktır.

* **BÖLÜM 3.1: Veri Sözlüğü**

Veri sözlüğü, bir veri tabanının meta verilerini içeren bir dosya veya dosya kümesidir. Veri sözlüğü, veri sahipliği, diğer nesnelere veri ilişkileri ve diğer veriler gibi veri tabanındaki diğer nesnelerle ilgili kayıtları içerir. Veri sözlüğü, herhangi bir ilişkisel veri tabanının önemli bir bileşenidir. (Anonim, 2018)[4]  
Aşağıda uygulamamızın veri sözlüğü bölümü maddeler halinde verilmiştir.

* Tablonun Adı: users
* müsteri\_id [PK]: Müşteriye ait benzersiz tanımlayıcı bir kimlik numarası.
* name: Müşterinin adı.
* password: Müşterinin şifresi.
* email: Müşterinin e-posta adresi.
* avatar: Müşterinin fotoğrafı.
* Role:Müşterinin Rolü
* createdAt:Müşterinin hesabını oluşturma tarihi

Tablo 1: Veri Sözlüğü-1users tablosu

* Tablonun Adı: products
* Mekan\_id [PK]: Sanatçıya ait benzersiz tanımlayıcı bir kimlik numarası.
* price: Mekanın fiyatı.
* ratings: Mekanın puanı.
* stock: Toplam bilet sayısı.
* numOfReviews: Mekanın toplam değerlendirme sayısı
* name:Mekanın adı
* description:Mekan hakkında açıklama
* images:Mekanın fotoğrafları
* category:Etkinliğin türü
* seller:Organizatör ismi
* reviews:Yapılan yorumların içeriği
* createdAt:Mekanın oluşturulma tarihi

Tablo 2: Veri Sözlüğü-2 products tablosu

* **BÖLÜM 3.2: İş Modeli**

Bu bölümde projenin Use-case’leri, Aktörleri ve bu use-case’lere ilişkin senaryolar ilgili bölümlerde ayrıntılı olarak açıklanacaktır.

* **BÖLÜM 3.2.1: Use-Case’lerin kısa tanımları ve Aktörler**

Use-case, bir sistem aracılığı ile sunulan veya sunulacak tek bir fonksiyonu tanımlamaktadır. Diğer bir deyişle, kullanıcıların bir fonksiyon aracılığı ile sistem üzerinde gerçekleştirmek istedikleri işlemleri tanımlamaktadır.

Aktörler, use-case analizinde, belli bir amacı gerçekleştirmeye yönelik olarak sistemle etkileşime giren kullanıcılar aktör olarak adlandırılır.

Bu bölümde aşağıda use-case’ler ve aktörler belirlenecektir.

Aktörler:

* Müşteri: Etkinliğe katılacak kişi.
* Organizatör/Mekan sahibi: Mekan ve etkinliğin sahibi kişi.
* Sistem yöneticisi: Sistem üzerindeki yetkilendirme, kullanıcı yönetimi ve genel sistem yönetimi işlemlerinden sorumlu kişi.

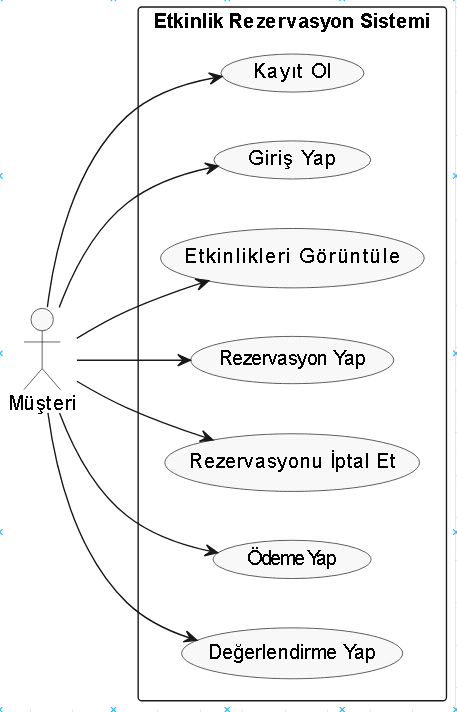
Use-case’ler:

* Kayıt olma: Sisteme müşteriler ve mekan sahipleri kayıt olabilecektir
* Giriş yapma: Sisteme müşteriler ve mekân sahipleri kaydoldukları bilgiler ile giriş yapabilecektir.
* Etkinlik oluşturma: Mekân sahibi, online platform üzerinden yeni bir etkinlik oluşturabilecektir.
* Etkinliğe katılma: Müşteriler oluşturulan etkinliklere rezervasyon yaparak katılabilecektir.
* Mekân ekleme: Mekân sahipleri, etkinlik için bir mekân ekleyebilir ve mekânın uygunluğunu kontrol edebilir.
* Personel Bilgilendirme: Mekân sahibi, mekanın içerdiği personel sayısını hakkında kısmında belirtebilir ve müşteriler de bunu görebilir.
* Yemek Seçimi: Mekan sahibi mekanın içerdiği yemek gibi özellikleri mekanın hakkında kısmında belirtebilir.Müşteriler de bunu görüntüleyebilir.
* Puan verme: Mekân sahipleri davet ettikleri sanatçının etkinliğe katılımına ve müşteriler de sanatçıdan memnun olma durumuna göre puan verecekler.
* Mekân ve Müşteri Profil Sayfaları: Mekân sahipleri ve Müşteriler, kendilerini tanıtan profil sayfaları oluşturabilir ve güncel bilgilerini paylaşabilir
* Etkinlik yeri görüntüleme: Müşteriler, sistemdeki katılmak istedikleri etkinliğin lokasyon bilgisini görüntüleyebilecek.
* Kullanıcı İstatistikleri: Sistem yöneticisi, müşterilerin etkinlik geçmişleri, tercihleri ve istatistikleri gibi bilgilere erişebilir.
* Geri Bildirim ve İnceleme: Müşteri, gerçekleşen etkinlik sonrasında mekânı veya hizmeti değerlendirebilir ve incelemeler yapabilir.
* Ödeme İşlemleri: Müşteri, etkinlik için ödeme yapabilir ve ödeme yöntemlerini seçebilir.
* Rezervasyon İptali: Müşteri, katılacağı etkinliği iptal edebilir.
* Organizasyon iptali: Mekân sahibi düzenleyeceği etkinliği iptal edebilir.
* Yetkilendirme ve Kullanıcı Yönetimi: Sistem yöneticisi, kullanıcı hesaplarını yönetebilir, yetkilendirme işlemleri yapabilir ve güvenlik önlemleri alabilir.
* **BÖLÜM 3.2.2: Use-Case Diyagramları**

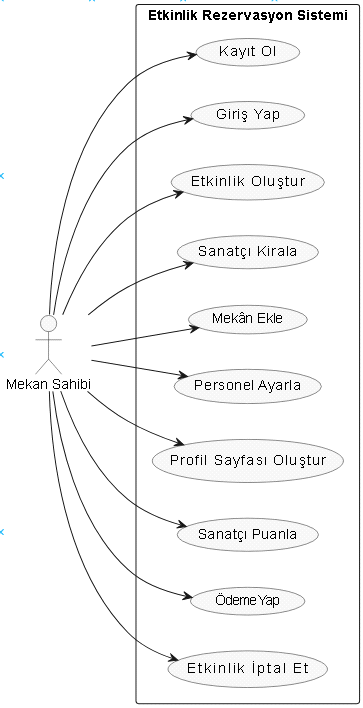
Medium adlı sitedeki tanımına göre Sistemler, kullanıcıların iş süreçlerini desteklemek üzere geliştirilmektedir. Bu nedenle iş süreçlerinin sistemler aracılığı ile yürütülmesi aşamasında kullanıcılar sistemdebirden fazla fonksiyona ihtiyaç duymaktadır. (Güneş, 2020) [5]Use Case Diyagramları, iş süreçlerinin yönetilmesi aşamasında ihtiyaç duyulan tüm fonksiyonları, bu fonksiyonları tetikleyecekaktörleri**,** fonksiyonlardan etkilenecek aktörleri ve fonksiyonlar arasındaki ilişkileri göstermek amacıyla kullanılmaktadır.

Yukarıda bölüm 3.2.1 de bahsedilen use-case’lerin kısa tanımları ve aktörlere göre bu bölümde use-case diyagramları verilecektir.

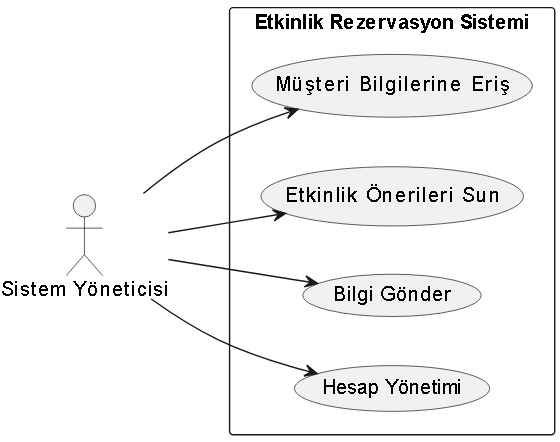
Aşağıda müşteri , mekan sahibi ve sistem yöneticisi aktörlerinin ilgili use-case diyagramları maddeler halinde belirtilecektir.



Şekil 1 : Müşteri aktörünün use-case diyagramı

* 

Şekil 2: Mekan Sahibi aktörünün use-case diyagramı



Şekil 3:Sistem Yöneticisi aktörünün use-case diyagramı

* **BÖLÜM 3.2.3: Use-Case Senaryoları ve İş Birliği Diyagramları**

Bu bölümde kısa tanımlar verilen ve diyagramlar ile gösterilen use-case’lerin senaryoları yazılacaktır.Use-case senaryolarını yazdıktan sonra ise iş birliği diyagramları verilecektir.

* Use-case senaryoları
* Kayıt Olma:
* Mekan sahibi veya müşteri kendi bölümlerinden kayıt formunu doldurarak sisteme kayıt olmaya başlar.
* Sistem, kullanıcıdan adı, soyadı, e-posta adresini ve şifre gibi gerekli bilgileri talep eder.
* Mekân sahibi veya müşteri, tüm gerekli bilgileri doğru ve eksiksiz bir şekilde girer.
* Sistem, Mekân sahibi veya müşteri girdiği bilgileri doğrulamak için e-posta adresine doğrulama bağlantısı gönderir.
* Mekân sahibi veya müşteri, e-posta hesabını kontrol eder ve doğrulama bağlantısına tıklar.
* Sistem Mekân sahibi veya müşteri kim kaydoluyorsa doğrular ve kaydın tamamlandığına dair bir onay mesajı gösterir.
* Kayıt yapan kişi, artık sisteme giriş yapabilir.

Tablo 3: Kayıt Olma Senaryosu

* Giriş Yapma:
* Mekân sahibi müşteri, giriş formunu doldurarak sisteme giriş yapmaya çalışır.
* Sistem, ilgili kişiden e-posta adresi ve şifre bilgilerini talep eder.
* İlgili kişi, kaydolduğu e-posta adresini ve şifresini girer.
* Sistem, İlgili kişinin girdiği bilgileri doğrular ve ilgili kişi rolüne göre ilgili kullanıcı arayüzüne yönlendirir.
* İlgili kişi, başarılı bir şekilde giriş yapar ve etkinlik rezervasyon sistemine erişim sağlar.

Tablo 4: Giriş Yapma Senaryosu

* Etkinlik Oluşturma:
* Mekân sahibi, sistemde etkinlik oluşturma işlemine başlar.
* Sistem, mekân sahibi/organizatör mekanıyla alakalı bilgileri doldurur.Mekan sistemdeki “products” tablosuna kayıt edilir.
* Mekân sahibi, gerekli bilgileri doğru ve eksiksiz bir şekilde girer.
* Sistem, girilen bilgileri doğrular ve yeni bir etkinlik oluşturulduğuna dair bir onay mesajı gösterir.
* Etkinlik oluşturulduktan sonra, sistem etkinliği platformda yayınlar ve müşterilerin görüntülemesini sağlar.

Tablo 5: Etkinlik Oluşturma Senaryosu

* Etkinliğe Katılma:
* Müşteri, sistemdeki etkinlikler arasından filtreleme ve arama özelliğini kullanarak istediği etkinliğe gider.
* Müşteri beğendiği etkinliğe devam ederek hakkında kısmında ücret,konum,etkinlik puanı ve tarih hakkında bilgileri görüntüleyebilir.
* Müşteriye, etkinlikle ilgili rezervasyon bilgileri ve ödeme detayları gösterilir.
* Müşteri, etkinliğe başarılı bir şekilde rezervasyon yapar.

Tablo 6: Etkinliğe Katılma Senaryosu

* Mekân Ekleme:
* Mekân sahibi, sisteme yeni bir mekân eklemek için işlem başlatır.
* Sistem, mekân sahibinden mekânla ilgili ayrıntıları talep eder, bu ayrıntılar arasında mekân adı,etkinlik türü ücret ve konum gibi özellikler yer alır.
* Mekân sahibi, gerekli bilgileri doğru ve eksiksiz bir şekilde girer.
* Sistem, girilen bilgileri doğrular ve yeni bir mekânın sisteme eklendiğini onaylar.
* Mekân sahibi, eklenen mekânın uygunluğunu kontrol eder ve mekânın rezervasyon durumunu takip edebilir.

Tablo 7: Mekan Ekleme Senaryosu

* Puan Verme:
* Sistem, mekân sahibine puan verme arayüzünü sunar.
* Müşteri mekân sahibinin etkinliğe katılımını ve performansını değerlendiren bir puan verir.
* Sistem, verilen puanı kaydeder ve Mekan sahibinin puanını günceller.

Tablo 8: Puan Verme Senaryosu

* Mekân Sahibi Profil Sayfaları:
* Mekân sahibi kendisini tanıtan ve güncel bilgilerini paylaşan bir profil sayfası oluşturmak istediğini belirtir.
* Sistem, mekân sahibine profil sayfası oluşturma arayüzünü sunar.
* Mekân sahibi profil sayfasında yer almasını istediği bilgileri (ad, fotoğraf, iletişim bilgileri, referanslar vb.) girer.
* Sistem, girilen bilgileri kaydeder ve mekân profil sayfasını oluşturur.
* Profil sayfası, sistem üzerinde görüntülenebilir ve güncellenen bilgileri içerir.
* Müşteriler, mekân sahibi hakkında daha fazla bilgi edinmek için ilgili profil sayfasını ziyaret edebilirler.

Tablo 9: Mekan Sahibi Profil Güncelleme Senaryosu

* Etkinlik Yeri Görüntüleme:
* Müşteri etkinliğin konumuna mekan bilgileri kısmından ulaşabilir.
* Sistem, müşteriye etkinliklerin listelendiği bir sayfa gösterir ve müşteri seçtiği etkinliği bulur.
* Müşteri, seçtiği etkinliğin detaylarını görüntülemek için ilgili etkinliğin sayfasına gider.
* Sistem, etkinlik sayfasında etkinliğin lokasyon bilgisini, adresini ve harita üzerindeki konumunu gösterir.
* Müşteri, etkinliğin yerini görüntüler ve gerekli bilgilere erişir.

Tablo 10 :Etkinlik Yeri Görüntüleme Senaryosu

* Kullanıcı İstatistikleri:
* Sistem yöneticisi, müşterilerin etkinlik geçmişleri, tercihleri ve istatistikleri gibi bilgilere erişmek istediğini belirtir.
* Sistem, yöneticiye kullanıcı istatistikleri arayüzünü sunar ve istatistikleri görüntülemek için seçenekler sağlar.
* Yönetici, istediği istatistiklere göre filtrelemeler yaparak belirli bir kullanıcı grubunu veya tüm kullanıcıları seçebilir.
* Sistem, seçilen kriterlere göre kullanıcıların etkinlik geçmişleri, katılım oranları, tercihleri, puanları gibi istatistikleri görüntüler.
* Yönetici, elde edilen bilgilere dayanarak çeşitli analizler yapabilir ve sistem performansını değerlendirebilir

Tablo 11 : Kullanıcı İstatiskileri Görüntülüme Senaryosu

* Geri Bildirim ve İnceleme:
* Müşteri, gerçekleşen etkinlik sonrasında mekânı veya hizmeti değerlendirmek ve incelemeler yapmak istediğini belirtir.
* Sistem, müşteriye geri bildirim ve inceleme arayüzünü sunar ve ilgili alanları doldurması için gerekli bilgileri sağlar.
* Müşteri, etkinlik mekân hakkında derecelendirme yapar, yorumlar ekler ve deneyimini paylaşır.
* Sistem, müşterinin geri bildirimini kaydeder ve ilgili profil veya etkinlik sayfasında görüntüler.
* Diğer kullanıcılar, geri bildirim ve incelemeleri okuyabilir ve bu bilgileri etkinlik veya hizmet seçimlerinde dikkate alabilirler.

Tablo 12 : Geri bildirim İnceleme Senaryosu

* Ödeme İşlemleri:
* Müşteri, etkinlik için ödeme yapmak ve ödeme yöntemlerini seçmek istediğini belirtir.
* Sistem, müşteriye ödeme arayüzünü sunar ve desteklenen ödeme yöntemlerini gösterir.
* Müşteri, tercih ettiği ödeme yöntemini seçer ve gerekli ödeme bilgilerini girer.
* Sistem, müşterinin ödeme bilgilerini doğrular ve işlemi gerçekleştirir.
* Müşteri, ödeme işlemi sonucunda bir onay veya ödeme bilgisi alır.
* Mekân sahibi, organizasyonuna kiraladığı sanatçıya ödeme yapmak istediğini belirtir.
* Sistem, mekân sahibine ödeme arayüzünü sunar ve gerekli bilgileri sağlar.
* Sistem, mekân sahibinin ödeme bilgilerini doğrular ve işlemi gerçekleştirir.
* Mekân sahibi, ödeme işlemi sonucunda bir onay veya ödeme bilgisi alır.

Tablo 13 : Ödeme İşlemleri Senaryosu

Yetkilendirme ve Kullanıcı Yönetimi:

Sistem yöneticisi, kullanıcı hesaplarını yönetmek, yetkilendirme işlemleri yapmak ve güvenlik önlemleri almak istediğini belirtir.

Sistem, yöneticiye kullanıcı yönetimi arayüzünü sunar ve gerekli yetkilendirme ve güvenlik seçeneklerini sağlar.

Yönetici, kullanıcıların erişim kontrolünü yönetebilir ve belirli işlevlere veya verilere erişim izinlerini ayarlayabilir.

Yönetici, sistemdeki kullanıcıların aktivitelerini izleyebilir, güvenlik ihlallerini tespit edebilir ve gerektiğinde önlemler alabilir.

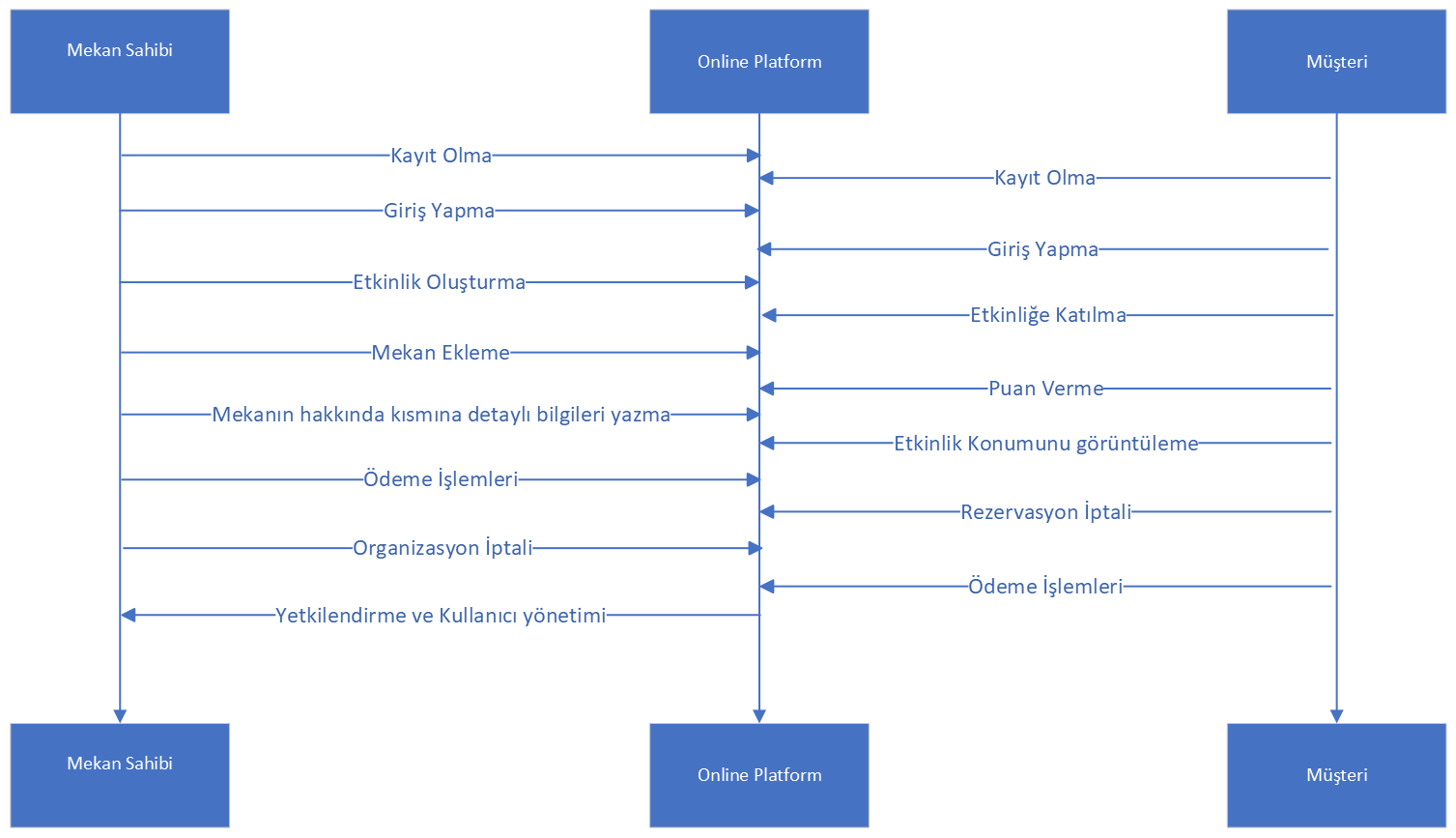
Tablo 14 : Yetkilendirme Ve Kullanıcı Yönetimi Senaryosu

İş birliği diyagramı:

İş birliği diyagramı, sistem içindeki aktörlerin (kullanıcılar, sistem bileşenleri, dış sistemler vb.) birbirleriyle nasıl iletişim kurduklarını ve iş birliği yaptıklarını gösteren bir UML diyagramıdır. (Güneş, medium, 2017)[6]  
İş birliği diyagramı, belirli bir kullanım durumu veya senaryo için aktörlerin rollerini, bu aktörler arasındaki mesaj alışverişini ve işbirliği yollarını görsel olarak temsil eder. Bu diyagram, bir sistemdeki etkileşimleri anlamak, aktörlerin sorumluluklarını ve rollerini belirlemek, iletişim düzenini göstermek ve iş birliği yapılacak süreçleri tasarlamak için kullanılabilir.  
İş birliği diyagramı, sistemin içsel yapısını ve işleyişini ayrıntılı olarak açıklamaz, bunun yerine aktörler arasındaki iletişimi ve iş birliğini odaklar. Genellikle sistem analizi ve tasarım süreçlerinde kullanılan bir araçtır ve gereksinim analizinden başlayarak sistemin uygulanmasına kadar olan süreçte kullanıcıların, bileşenlerin ve dış sistemlerin etkileşimini modellemek için kullanılır.

Bu diyagram, karmaşık bir sistemi daha anlaşılır ve görsel bir şekilde temsil etmeyi sağlar. Ayrıca, ekip üyeleri arasında iletişimi kolaylaştırır, gereksinimleri belirlemeye yardımcı olur ve tasarım sürecinde karar verme sürecini destekler.

Oluşturulacak sistemin iş birliği diyagramı aşağıdaki şekilde özetlenmiştir.



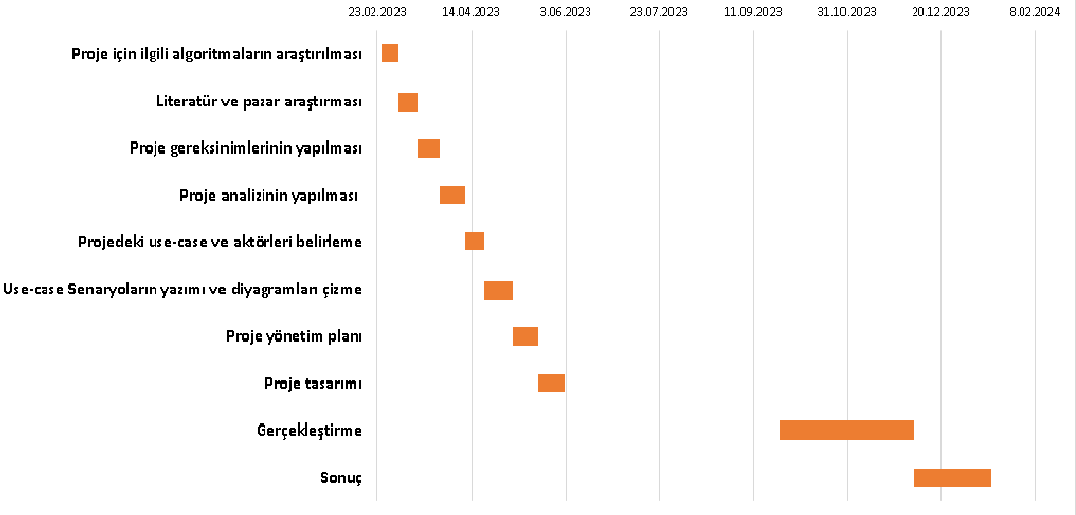
Şekil 4: İş Birliği Diyagramı

* **BÖLÜM 3.3: Yazılım Proje Yönetim Planı**

Bu bölümde gantt çizelgesinin çizilmesi, proje risk Matrisinin oluşturulması, kullanılacak teknolojik altyapı ve özelliklerin belirlenmesi ve sistem gereksinimleri kısımları ele alınarak aşağıda bölüm bölüm incelenecektir.

**BÖLÜM3.3.1:GanttÇizelgesi**  
blog.cubid.io sitesine göre Gantt çizelgesi, Henry Gantt tarafından tasarlanan, iş yönetiminde zaman planlamasını sağlamaya yönelik oluşturulmuş bir diyagramdır. 1910 yılında ortaya çıkarılmış bu çizelge, yaygın olarak proje yönetiminde kullanılarak bir zaman ölçeğine göre görevlerin planlanması için kullanılan bir araçtır (Arslan, 2019)[7]. Gantt çizelgesi, iş planlaması gerektiren birçok sektörde kendine yaşam alanı bulabildiği için günümüzde de üretim ve proje planlamalarında yaygın bir şekilde kullanılan görsel iş planlama aracıdır.

Aşağıda projenin gannt çizelgesi verilmiştir.

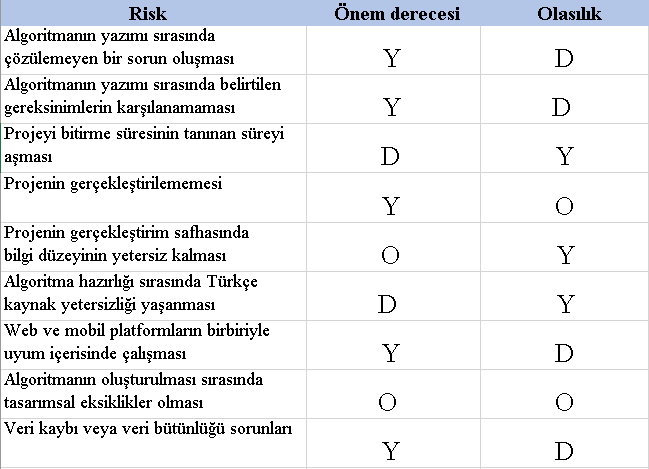


Şekil 5: Gantt Çizelgesi

* **BÖLÜM 3.3.2: Proje Risk Matrisi**

Risk matrisi, risk değerlendirmesi için kullanılan bir proje yönetim aracıdır. Risklerin olasılığı ve riskin ciddiyeti açısından değerlendirilmesine yardımcı olur. Aynı zamanda olasılık ve etki matrisi olarak da bilinir (Anonim, creately, 2022)[8].

Aşağıda verilen grafik gerçekleştirilecek olan projenin risklerini vermiştir. Bu risklerin önem derecelerini ve gerçekleşme olasılıklarını yüksek(Y) , orta(O) ve düşük(D) şeklinde belirtmiştir.



Şekil 6: Proje Risk Matrisi

**BÖLÜM 3.3.3: Kullanılacak Teknolojik Alt Yapı ve Özellikler**

Program Geliştirme Araçları (IDE) olarak Web tarafında, React.js ve Node.js kullanılacağından, popüler bir IDE olan Visual Studio Code tercih edilecektir. Bu IDE, React.js ve Node.js projelerini geliştirmek, hata ayıklamak ve test etmek için kullanışlıdır. Ayrıca, projenin yönetimini kolaylaştırmak adına Redux Toolkit entegrasyonu da sağlamaktadır.Mobil tarafında React Native kullanılacağından, React Native projelerini geliştirmek için yine Visual Studio Code kullanılacaktır.

Veri tabanı kısmında MongoDB kullanılacağından, MongoDB için geliştirilmiş olan MongoDB Compass adlı araç kullanılacaktır. Bu araç, MongoDB veritabanını oluşturma, yönetme, sorguları çalıştırma ve verileri görsel olarak inceleme gibi işlemleri kolaylaştırır.

Web tarafında sunucu tarafı için Express.js kullanılacaktır. Bu hafif ve esnek web uygulama çerçevesi, Node.js üzerinde çalışarak server-side işlemleri kolaylaştırır.Bu teknolojik alt yapı ve araçlar, proje ekibinin verimliliğini artırmak, hızlı bir geliştirme süreci sağlamak ve kaliteli bir ürün ortaya çıkarmak adına seçilmiştir.

**BÖLÜM 3.3.4: Sistem Gereksinimleri**

* 1. Web Tarafı Gereksinimleri:
* İşletim Sistemi: Projenin desteklediği işletim sistemleri arasında Windows, macOS ve Linux bulunmaktadır.
* Web Sunucusu: Node.js ve Express.js kullanıldığı için, Node.js'in en son sürümü tercih edilecektir.

Tarayıcı: Sistemi kullanacak olan kullanıcılar için en son sürümde Chrome, Firefox, Safari veya Edge gibi modern tarayıcılar önerilmektedir.

Mobil Tarafı Gereksinimleri:

* İşletim Sistemi: Uygulama, Android işletim sistemini desteklediği için kullanıcıların Android işletim sistemine sahip cihazları kullanmaları beklenir.
* Minimum Android Sürümü: Uygulamanın çalışması için gereken minimum Android sürümü Android 4.1 veya daha üstüdür.
* Entegrasyonlar ve veri tabanı: MongoDB veri tabanı kullanıldığından, projenin veri tabanı bağlantıları ve entegrasyonları buna uygun şekilde yapılandırılmalıdır.
* **BÖLÜM 4: PROJE TASARIMI**

Bu çalışmada, yazılım projesinin tasarım aşamasında Draw.io adlı görsel çizim aracının tercih edilmesinin arkasındaki temel nedenlerden biri, kullanıcı dostu arayüzü ve geniş diyagram seçenekleridir. Draw.io'nun arayüzü, kullanıcılara hızlı ve etkili bir şekilde çizim yapma imkanı tanırken, sunduğu geniş diyagram seçenekleri sayesinde farklı tasarım gereksinimlerine uyum sağlamak mümkün olmaktadır.

Özellikle, proje ekibi olarak Draw.io'nun sağladığı araçlarla, projenin temel bileşenlerini, veri akışlarını, genel yapısını ve kullanıcı arayüzünü görsel olarak ifade etme şansına sahip olduk. Bu, proje ekibinin birlikte çalışma sürecini kolaylaştırmış ve ortak bir vizyon oluşturulmasına yardımcı olmuştur. Draw.io'nun sunduğu çıktıları kolayca paylaşabilme ve iş birliği yapabilme özellikleri de projenin paylaşılabilirliğini artırdı.

Projeyi takip eden diğer ekip üyeleriyle veya paydaşlarla, çizimlerin veya diyagramların paylaşılması ve üzerinde işbirliği yapılması, iletişimi güçlendirdi. Bu sayede, herkesin projenin ilerlemesini daha iyi anlaması ve gerektiğinde düzenlemeler yapabilmesi mümkün hale geldi.

Proje tasarımında, draw.io'nun sunduğu çeşitli diyagram türlerinden faydalanıldı. Veri akış diyagramları, yazılımın işleyişini anlatmak ve veri akışlarını görselleştirmek için kullanıldı. Yapısal diyagramlar ise yazılımın bileşenlerini, sınıflarını ve ilişkilerini göstermek amacıyla oluşturuldu

Draw.io'nun kullanımı, projenin tasarım sürecinde büyük bir kolaylık sağladı. Görsel çizimler, proje ekibi arasında ortak bir anlayış oluşturmayı ve iletişimi geliştirmeyi sağladı. Ayrıca, draw.io'nun kullanımıyla proje tasarımı daha anlaşılır hale getirildi ve gelecekteki geliştirme aşamaları için bir temel oluşturuldu.

Projenin mimari tasarım aşamasında, Sistem Mimarisi Oluşturulması başlığı sağlanacak olup, projenin genel yapı ve organizasyonu belirlenir. Bu adım, projenin büyük resmini çıkarmak ve sistemin temel bileşenlerini tanımlamak için önemlidir. Sistem mimarisi oluşturulurken, projenin gereksinimlerini karşılamak adına kullanılacak teknolojiler, iletişim protokolleri ve anahtar bileşenler belirlenir.

Proje Modüllerinin Tanımlanması ve Başlangıç Sınıf Diyagramının Oluşturulması ile, projeyi oluşturan ana modüllerin belirlenmesini içerir. Her bir modül, projenin farklı işlevselliğini veya bileşenini temsil eder. Bu modüller arasındaki ilişkiler ve etkileşimler analiz edilerek, başlangıç sınıf diyagramı oluşturulur. Bu diyagram, projenin temel yapı taşlarını ve modüller arasındaki ilişkileri görsel olarak ifade eder.

Veri Tabanı Mimarisi Oluşturulması ile, projenin veri yönetimi ve depolama gereksinimleri belirlenir. Veri tabanı mimarisi, projenin veritabanı yapısını ve ilişkilerini içerir. Veri tabanı tasarımı, projenin veri entegrasyonu, güvenliği ve performansını etkileyen kritik bir adımdır.

Detaylı tasarım aşamasında ise Projeyi Oluşturan Ana Modüllerin Detaylandırılması ve Sınıf Seviyesinde Parçalara Bölünerek İlişkilerin Tespit Edilmesi başlığı sağlanarak, her ana modül detaylı bir şekilde incelenir. Modüller sınıf seviyesinde parçalara bölünür ve bu parçalar arasındaki ilişkiler tespit edilir. Bu adım, projenin daha derinlemesine analizini sağlar ve sınıflar arasındaki etkileşimleri netleştirir.

Bu detaylı analizin sonucunda, İlgili Diyagramlar Kullanılarak Sınıf Tasarımlarının Yapılması adımına geçilir. UML sınıf diyagramları veya benzeri görsel araçlar kullanılarak, her bir sınıfın özellikleri, metotları ve ilişkileri belirlenir. Bu sınıf tasarımları, yazılım geliştirme sürecinde temel bir rehber sağlar ve kodlama aşamasında kullanılır.

Bu aşamaların detaylı bir şekilde ele alınması, projenin daha sağlam bir temel üzerine inşa edilmesine ve geliştirilmesine olanak tanır. Ayrıca, ilgili ekibin daha iyi bir anlayış geliştirmesine ve proje süreçlerinin daha etkili bir şekilde yönetilmesine katkı sağlar.

* **BÖLÜM 4.1: Mimari Tasarım**

Bu mimari tasarım aşamasında sistem mimarisi, modüller ve veri tabanı mimarisi başlıkları altında incelenecektir.

* **BÖLÜM 4.1.1: Sistem Mimarisi**

Web ve mobil için online etkinlik rezervasyon sistemimiz için istemci-sunucu mimarisi kullanılacaktır.

İstemci tarafı:

Web istemcisi için kullanılan teknoloji: react.js

Mobil istemcisi için kullanılan teknoloji: react native

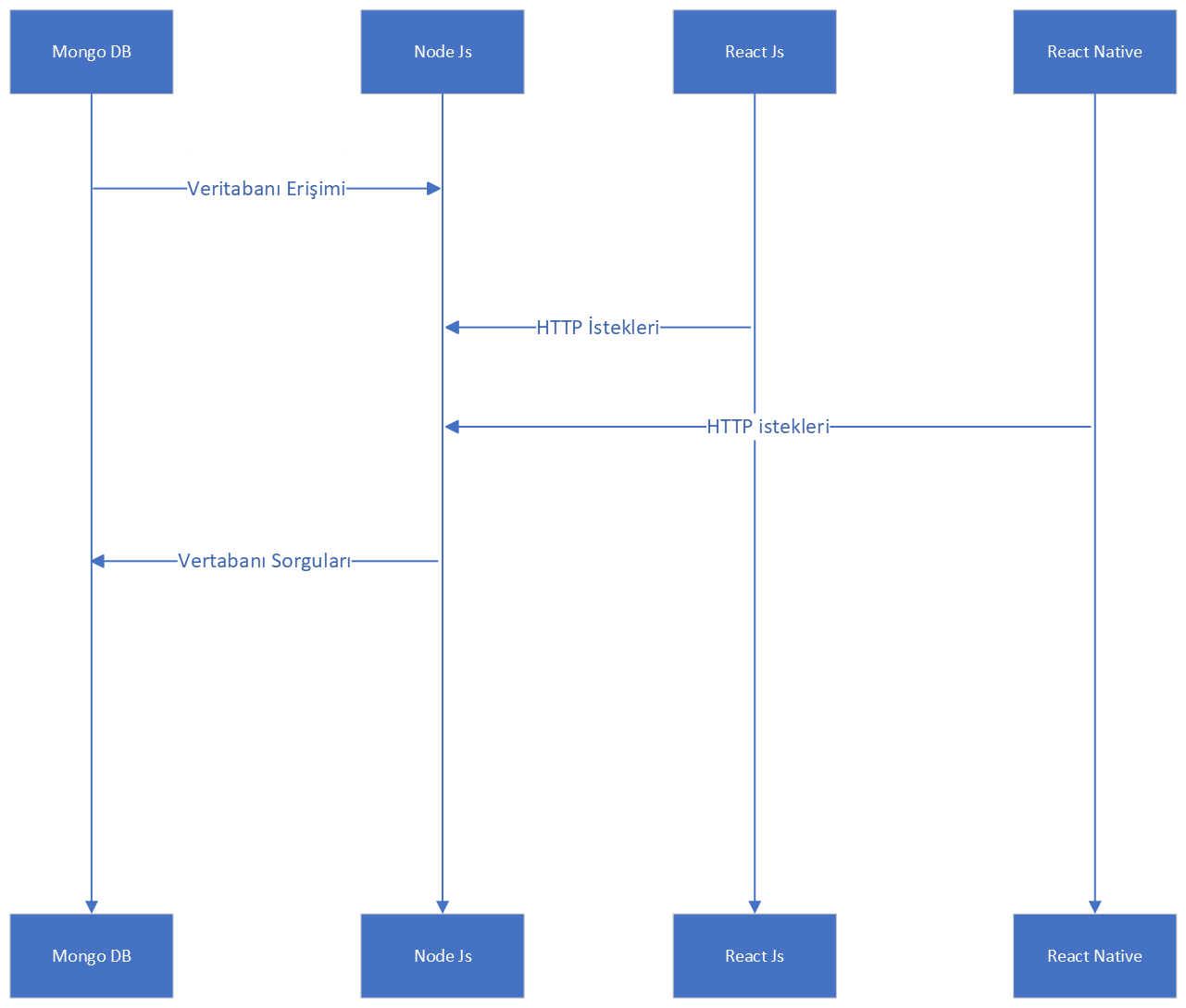
Sunucu tarafı:

Sunucu tarafında kullanılan teknoloji : Node.js , Express.js

Veritabanı yönetim sistemi : MongoDB

İstemci-sunucu mimarisi kullanılarak geliştirilecek sistemim yukarıdaki yapıları kullanacaktır.

Aşağıdaki verilecek olan tasarım sistemin genel sistem mimarisini basit ve anlaşılır bir şekilde görselleştirir., Sistemdeki bileşenleri temsil eden kutular ve bu bileşenler arasındaki ilişkileri gösterir.   
MongoDB veritabanı, Node.js ve Express.js sunucu, React.js web istemcisi ve react native mobil istemcisi olarak temel bileşenler vardır. İstemci ve sunucu arasındaki iletişim HTTP istekleri üzerinden gerçekleşir ve sunucu, MongoDB veritabanıyla etkileşime geçer.



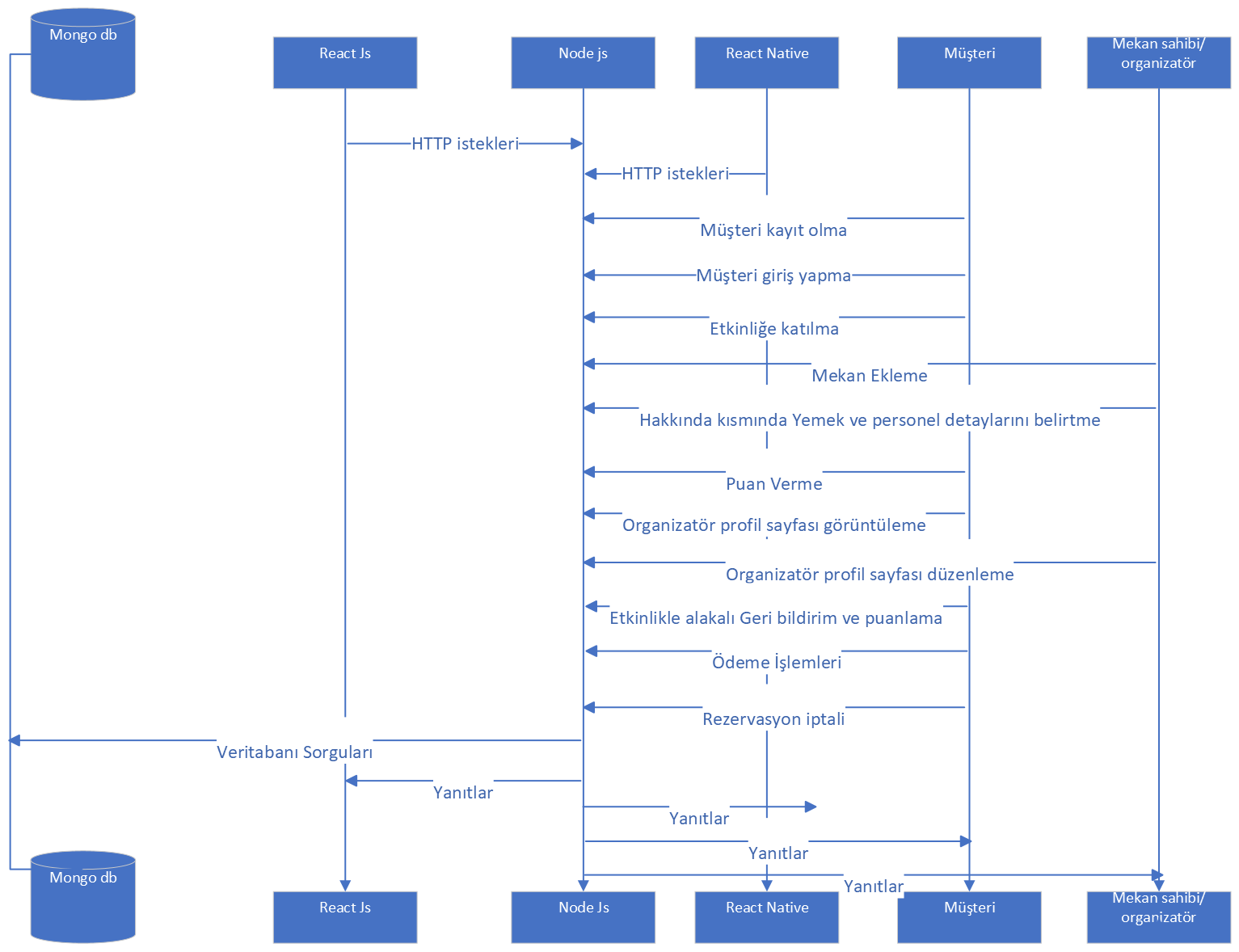
Şekil 7: Sistem Mimarisi

Sisteme giriş yapacak kullanıcıların İstemci tarafından sunucuya gönderilen istekleri ve sunucunun buna karşılık oluşturduğu yanıtları özetleyen daha detaylı olan sistem tasarımı da aşağıda verilecektir. Mimari, üç farklı aktörü içerir: Müşteri (Customer) ve Mekân Sahibi (Owner).

İstemci tarafında React.js ve React Native kullanılmaktadır. Angular.js, web istemcisi için kullanılırken, React Native mobil istemci tarafı için tercih edilmektedir. İstemciler, sunucuya HTTP istekleri gönderirler. Bu istekler aracılığıyla kaydolma, giriş yapma, etkinlik oluşturma, etkinliğe katılma, rezervasyon yapma, puan verme gibi işlemleri gerçekleştirebilirler.

Sunucu tarafında ise Node.js kullanılmaktadır. Sunucu, istemcilerden gelen HTTP isteklerini alır ve MongoDB veri tabanı ile iletişim kurar. Veri tabanı sorguları aracılığıyla kullanıcı kayıtları, etkinlik bilgileri, rezervasyonlar ve diğer veriler yönetilir.

Sunucu, aldığı isteklere yanıtlar gönderir. Yanıtlar, işlemin başarılı olup olmadığını veya gerektiğinde hata mesajlarını içerir. Bu yanıtlar, istemcilerin kullanıcı arayüzlerinde gerekli geri bildirimleri görmelerini sağlar.



Şekil 8: Sistem Mimarisi 2

* **BÖLÜM 4.1.2: Modüller**

Bu bölümde proje modüllerinin tanımlanması ve başlangıç sınıf diyagramının oluşturulması sağlanacaktır.İlk olarak proje modülleri ele alınacak ardından  
başlangıç sınıf diyagramları ele alımacaktır.

* **Proje modülleri**

Proje modülleri, bir projenin farklı işlevsel veya bölgesel parçalarını temsil eden bağımsız bileşenlerdir. Genellikle projenin farklı işlevselliğini yönetmek, karmaşıklığı azaltmak ve modüler bir yapı oluşturmak amacıyla kullanılırlar.

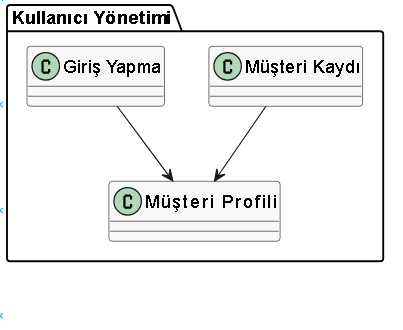
Proje modülleri, projenin belirli bir alanında bir araya getirilen ilgili sınıfların ve bileşenlerin bir koleksiyonudur. Her modül, benzer işlevlere, sorumluluklara veya bileşenlere sahip olan sınıfları ve alt modülleri içerir. Bu modüler yapı, projenin daha iyi anlaşılmasını, sürdürülebilirliğini ve ölçeklenebilirliğini artırır.

Her modül, belirli bir işlevi yerine getirmek için bir araya gelen sınıfların bir grubudur.

Bu bağlamda, proje modülleri, projenin parçalarını organize etmek, bağımsızlık sağlamak ve karmaşıklığı azaltmak için kullanılan kavramsal yapıları ifade eder. Her modül, belirli bir işlevi yerine getirmek için gerekli olan sınıfları, bileşenleri ve ilişkileri içerir.

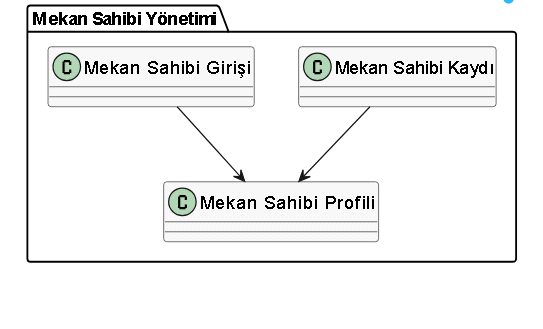
Aşağıda projenin modülleri başlıklar altında verilecektir.

* 1.Kullanıcı yönetimi modülü



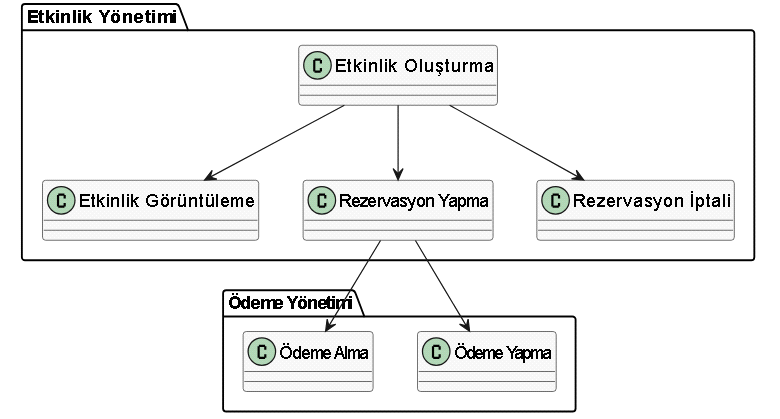
Şekil 9:Kullanıcı Yönetimi Modülü

2. Mekân sahibi yönetimi modülü:



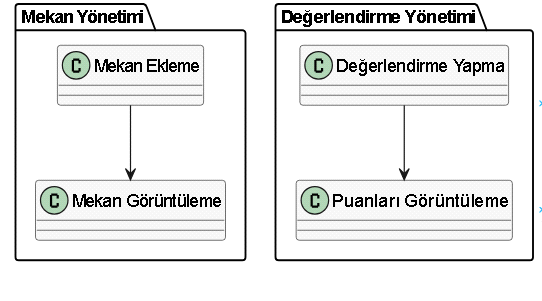
Şekil 10: Mekan Sahibi yönetimi Modülü

3.Etkinlik Yönetimi Modülü



Şekil 11: Etkinlik Yönetimi Modülü

4. Mekân ve değerlendirme yönetimi



Şekil 12:Mekan ve Değerlendirme Yönetimi

* **Başlangıç sınıf diyagramı**

Başlangıç sınıfı diyagramı, bir sistem veya yazılım uygulamasının başlangıç durumunu ve başlangıç noktasını gösteren bir UML) diyagramıdır.

Bu diyagram, sistemin temel nesnelerini, bileşenlerini veya modüllerini temsil ederek, sistemin başlangıç noktasını belirlemek amacıyla kullanılır.

Başlangıç sınıfı diyagramı, sistemdeki ilk adımları yönlendiren ve diğer bileşenler arasında bir bağlantı noktası olan bir sınıf veya bileşeni temsil eder.

Söz konusu sınıf veya bileşen genellikle başlatma işlemlerini gerçekleştirir ve sistemdeki diğer bileşenlerle ilişkilendirilir. Bu sayede, başlangıç sınıfı diyagramı, sistemin genel yapısını anlamamızı sağlar ve diğer sınıf diyagramlarına referans olabilir.

Başlangıç sınıfı diyagramında, başlangıç sınıfının adı ve başlangıç işlemini gösteren bir ok kullanılır. Bu ok, başlangıç sınıfının başlatma işlemiyle diğer sınıflar veya bileşenler arasında bir ilişki olduğunu gösterir.

Sonuç olarak, başlangıç sınıfı diyagramı, bir sistem veya yazılım uygulamasının başlangıç durumunu belirleyen ve sistemin yapısal ilişkilerini gösteren bir UML diyagramıdır.

Yukarıdaki bir üstte başlıkta oluşturulan proje modüllerine göre aşağıda sistemin başlangıç sınıf diyagramı tasarımı oluşturulmuştur.

metin, diyagram, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 13:Başlangıç Sınıf Diyagramı

* **BÖLÜM 4.1.3: Veri tabanı mimarisi**
* Döküman Tabanlı Veritabanı Yapısı

Döküman tabanlı veritabanları, geleneksel ilişkisel veritabanlarına alternatif olarak kullanılan bir modeldir. Bu tasarım yaklaşımı, özellikle MongoDB gibi NoSQL veritabanlarında yaygın olarak kullanılır.

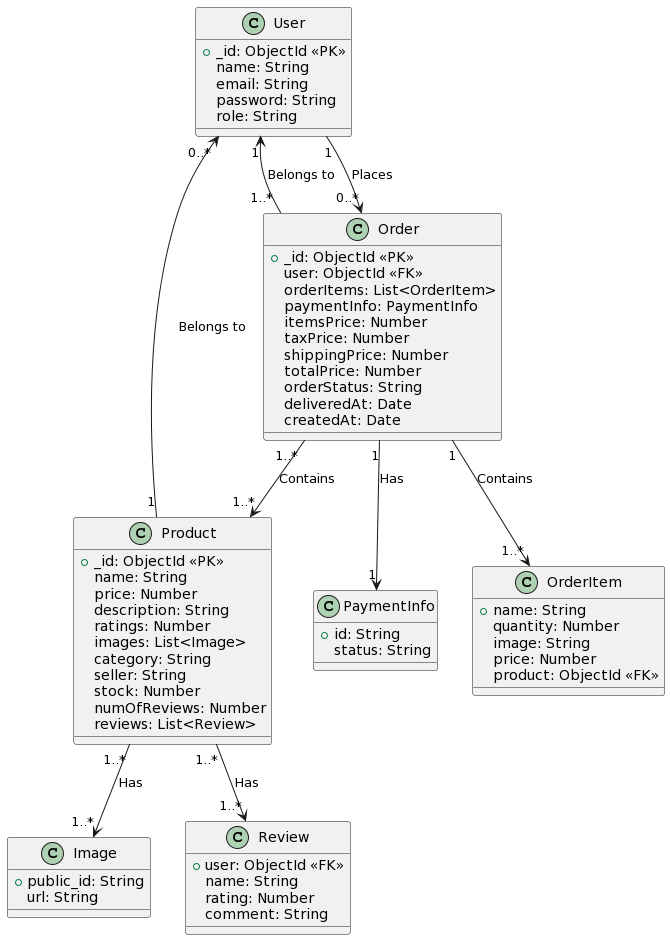
Entity/Relationship (E/R) diyagramları, veritabanı tasarımında döküman tabanlı yapıları modellendirmek için de kullanılabilir. "E" (entity/varlık) ve "R" (relationship/ilişki) kelimelerinin baş harflerinden oluşan E/R diyagramları, sistemdeki veri akışını ve ilişkileri görsel olarak temsil etmek için kullanılır.

Varlık (Entity), gerçek dünyadaki nesnelerin veya kavramların temsilidir. Örneğin, bir kütüphane sistemi için "Kitap" ve "Yazar" gibi varlıklar düşünülebilir. Her varlık, benzersiz bir kimlik alanına (primary key) ve verilerin niteliklerini (attribute) temsil eden özelliklere sahiptir. Örneğin, bir kitabın adı, yayın yılı ve yazarı gibi nitelikler olabilir.

İlişki (Relationship), varlıklar arasındaki bağlantıları temsil eder. Örneğin, bir kitap bir yazar tarafından yazılmış olabilir veya bir yazar birden fazla kitap yazmış olabilir. İlişkiler, varlıklar arasında bir çizgiyle gösterilir ve ilişkinin doğasını belirten bir etiketle tanımlanır. Örneğin, "Bir Yazarın Yazdığı Kitaplar" şeklinde bir ilişki etiketi olabilir.

E/R diyagramları, varlık ve ilişkileri görsel olarak temsil eden grafiksel bir gösterim sunar. Bu diyagramlar, bir sistem tasarımı sırasında gereksinimleri anlamak, veri tabanının yapısını görselleştirmek ve tasarım hatalarını tespit etmek için kullanılır.

Proje tasarımının bu aşamasında yukarıda tanımlanan E/R diyagramı, MongoDB gibi döküman tabanlı bir veritabanı yapısı için uyarlanmıştır.



Şekil 14:Veritabanı mimarisi

E-ticaret sistemini temsil eden E/R diyagramında bulunan varlıklar ve ilişkiler aşağıdaki gibi tanımlanmıştır:

User (Kullanıcı): Sisteme kayıt olan kullanıcıları temsil eder. Kullanıcılar birincil anahtarları \_id ile tanımlanır ve ad, e-posta, şifre ve rol gibi özelliklere sahiptir.

Order (Sipariş): Kullanıcıların oluşturduğu siparişleri temsil eder. Bir siparişin birincil anahtarı \_id'dir ve kullanıcıya aittir. Sipariş, sipariş kalemleri, ödeme bilgisi ve diğer detayları içerir.

OrderItem (Sipariş Kalemi): Bir sipariş içindeki ürünleri temsil eder. Sipariş kalemi, bir ürünün adı, miktarı, resmi, fiyatı ve ilişkili ürün bilgilerini içerir.

Product (Ürün): Satışa sunulan ürünleri temsil eder. Bir ürünün adı, fiyatı, açıklaması, derecelendirmesi, görüntüleri, kategorisi, satıcısı, stoku, incelemeleri ve diğer detayları içerebilir.

Image (Görüntü): Ürünlerin görüntülerini temsil eder. Bir ürün, birden fazla görüntü içerebilir.

Review (İnceleme): Ürünlere yapılan incelemeleri temsil eder. İnceleme, kullanıcı tarafından yapılan incelemenin adı, derecesi ve yorumunu içerir.

PaymentInfo (Ödeme Bilgisi): Bir siparişin ödeme bilgilerini temsil eder. Bir ödemenin benzersiz bir kimliği (id) ve durumu (status) bulunabilir.

İlişkiler aşağıdaki şekilde tanımlanmıştır:

Bir kullanıcı birden çok sipariş verebilir, ancak bir sipariş yalnızca bir kullanıcıya aittir (User "1" --> "0..\*" Order : Places).

Bir sipariş, bir kullanıcıya aittir (Order "1..\*" --> "1" User : Belongs to).

Bir sipariş birden çok ürün içerebilir ve bir ürün birçok sipariş içinde yer alabilir (Order "1.." --> "1.." Product : Contains).

Bir ürün, bir veya daha fazla incelemeye ve bir veya daha fazla görüntüye sahip olabilir (Product "1.." --> "1.." Review : Has, Product "1.." --> "1.." Image : Has).

Bir sipariş, bir veya daha fazla sipariş kalemini içerebilir (Order "1" --> "1..\*" OrderItem : Contains).

Bir sipariş, bir ödeme bilgisine sahip olabilir (Order "1" --> "1" PaymentInfo : Has).

* İlişkisel olmayan veri tabanının oluşturulması

İlişkisel veri tabanları, verileri tablolar ve ilişkiler aracılığıyla düzenler, ancak MongoDB gibi NoSQL veri tabanları, esnek veri modelleri ve ölçeklenebilirlik avantajları ile farklı bir yaklaşım sunar.

MongoDB, belge tabanlı bir NoSQL veri tabanıdır ve verileri JSON benzeri belgeler içinde depolar. Bu belgeler, ilişkisel tabloların yerine koleksiyonlar olarak adlandırılır ve her bir belge, ilgili veri öğelerini içeren bir JSON formatında temsil edilir. Bu yapı, veri modelinin esnek ve uyarlanabilir olmasını sağlar.

Projemizde, online rezervasyon sistemi tasarlanırken MongoDB kullanılarak veriler düzenlenir. Örneğin, kullanıcı bilgilerini içeren bir belge, ilgili rezervasyon bilgilerini içerir ve bu belgeler arasında benzersiz anahtarlar (IDs) kullanılarak ilişkiler kurulabilir.

NoSQL veri tabanları, özellikle büyük ve karmaşık veri setlerini hızlı ve etkili bir şekilde işlemek için idealdir. MongoDB'nin esnek veri modeli, projenin ihtiyaçlarına uygun özelleştirilmiş bir veri düzeni oluşturmayı mümkün kılar. Bu sayede, projenin gereksinimleri değiştikçe ve büyüdükçe, veri tabanı modelini kolayca güncellemek ve adapte etmek mümkün olur.

MongoDB'nin NoSQL yaklaşımı, ilişkisel veri tabanlarından farklı olarak, şema tasarımında daha fazla esneklik sağlar. Veri tabanı şemasının değişmesi gerektiğinde, yeni alanlar eklemek veya var olanları güncellemek daha kolaydır. Bu özellik, projenin gelişim sürecindeki dinamizme cevap verir.

Sonuç olarak, MongoDB ve NoSQL veri tabanları, özellikle büyük ve değişken veri setleri üzerinde çalışırken esneklik, ölçeklenebilirlik ve performans avantajları sağlar. Bu, online rezervasyon sistemimizin veri yönetimi açısından etkili ve güçlü bir temel oluşturmasına yardımcı olur.

Bu bilgilere göre mongoDB kullanılarak yapılacak olan projenin json schema formatından görselleştirilmiş hali aşağıda verilmiştir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 15:Veritabanı Mimarisi 2

Bu veri tabanı şeması, etkinlik yönetimi platformunda kullanılan temel yapıları temsil etmektedir. Veri tabanı şemasındaki varlıklar ve buna bağlı olarak aralarındaki ilişkilerin açıklamalar aşağıda verilmiştir.

Product (Ürün):

\_id: ObjectId, ürünün benzersiz kimliği.

price: double, ürünün fiyatı.

ratings: int, ürüne verilen puan.

stock: int, ürünün stok miktarı.

numOfReviews: int, ürün için yapılan yorumların sayısı.

name: string, ürünün adı.

description: string, ürünün açıklaması.

category: string, ürünün kategorisi.

seller: string, ürünü satan kişi veya firma.

images: Image sınıfından bir liste, ürünle ilişkilendirilmiş resimlerin listesi.

user: ObjectId, ürünü ekleyen kullanıcının kimliği.

reviews: Review sınıfından bir liste, ürüne yapılan yorumların listesi.

createdAt: Date, ürünün oluşturulma tarihi.

\_\_v: int, sürüm kontrolü için kullanılan alan.

Image (Resim):

\_id: ObjectId, resmin benzersiz kimliği.

public\_id: string, resmin genel kimliği.

url: string, resmin URL'si.

Review (Yorum):

\_id: ObjectId, yorumun benzersiz kimliği.

user: ObjectId, yorumu yapan kullanıcının kimliği.

name: string, yorumu yapan kullanıcının adı.

rating: int, yorumda verilen puan.

comment: string, yapılan yorum.

Tablo 15: Veri Tabanı Mekan Tablosu

User (Kullanıcı):

\_id: ObjectId, kullanıcının benzersiz kimliği.

role: string, kullanıcının rolü (örneğin, admin, normal kullanıcı).

name: string, kullanıcının adı.

email: string, kullanıcının e-posta adresi.

password: string, kullanıcının şifresi.

avatar: Image sınıfından bir nesne, kullanıcının profiline eklenmiş avatar resmi.

createdAt: Date, kullanıcının hesap oluşturulma tarihi.

Bu sınıflar arasındaki ilişkiler şu şekildedir:

Product sınıfı, Image ve Review sınıfları ile ilişkilidir.

Image sınıfı, Product ve User sınıfları ile ilişkilidir.

Review sınıfı, Product ve User sınıfları ile ilişkilidir.

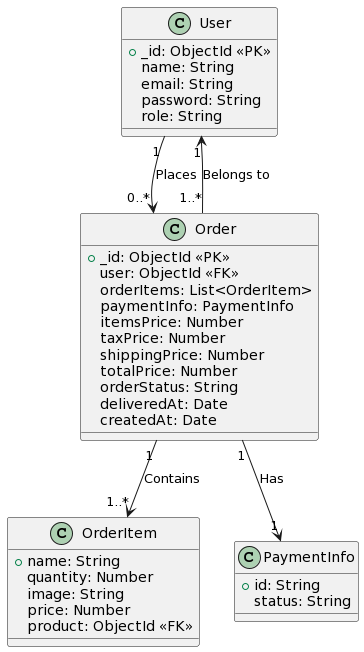
Bu model, ürünlerin resimleri ve yorumları ile ilişkilendirilmiş bir veritabanını temsil etmektedir.

Tablo 16: Veri Tabanı Kullanıcı Tablosu

* **BÖLÜM 4.2: Detaylı tasarım**

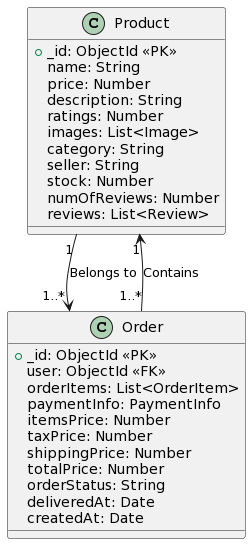
Bu bölümde Projeyi oluşturan ana modüllerin detaylandırılıp, sınıf seviyesinde parçalara bölünerek bunlar arasındaki ilişkiler tespit verilecektir.

Kullanıcı sınıfı



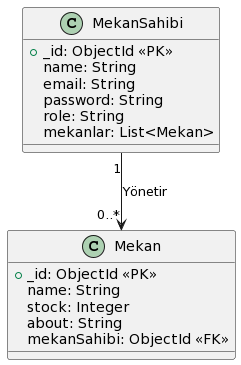
Şekil 16:Veritabanında Kullanıcı Sınıfı Mimarisi

* Etkinlik sınıfı



Şekil 17:Veritabanında Etkinlik Sınıfı Mimarisi

* Mekân sahibi sınıfı



Şekil 18:Veritabanında Mekan Sahibi Mimarisi

**BÖLÜM 4.2.1: Sınıf Tasarımı**

Sınıf tasarımı, yazılım geliştirme sürecinde kullanılan bir yaklaşımdır ve bir sistem veya uygulamanın yapısını, planlama, organize etme ve modelleme işlemidir. Sınıf tasarımı, nesne yönelimli programlamanın (OOP) temel prensiplerini uygular ve yazılımın modüler, esnek, sürdürülebilir ve yeniden kullanılabilir olmasını sağlar. Sınıf tasarımı, yazılımı bağımsız ve yeniden kullanılabilir parçalara ayırır. Bu sayede, her bir sınıf kendi sorumluluklarına odaklanır ve değişikliklerin diğer bileşenlere minimum etkisi olur. Bu modülerlik, kodun daha anlaşılır, sürdürülebilir ve test edilebilir olmasını sağlar.

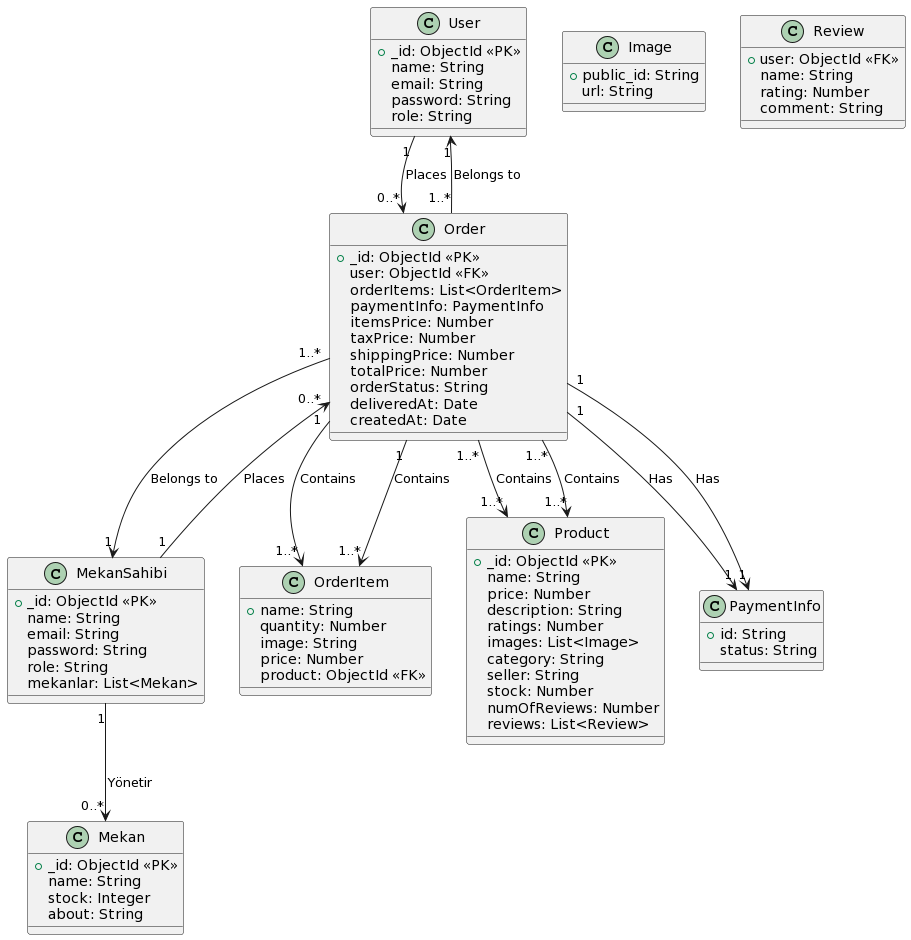
İyi bir sınıf tasarımı, yeni gereksinimlerin veya değişikliklerin kolaylıkla uygulanabilmesini sağlar. Sınıflar arasındaki ilişkiler, bağımlılıklar ve arayüzlerin doğru şekilde belirlenmesi, sistemdeki değişikliklerin yayılmamasını ve istenilen özelliklerin eklenmesini veya çıkarılmasını kolaylaştırır.

İyi bir sınıf tasarımı, hata ayıklama ve bakım süreçlerini kolaylaştırır. Sınıfların net sınırları ve sorumlulukları olduğunda, hataların kaynağı daha kolay belirlenebilir ve düzeltilebilir. Ayrıca, bir sınıfın içinde yapılan değişikliklerin diğer sınıflara minimum etkisi olur, bu da bakım süreçlerini daha düşük maliyetli ve risksiz hale getirir.

İyi tasarlanmış sınıflar, kodun daha anlaşılır ve okunabilir olmasını sağlar. Sınıflar arasındaki ilişkilerin doğru bir şekilde modellenmesi ve sınıf isimlerinin ve yöntemlerin anlamlı olması, kodun kolayca anlaşılmasını ve takip edilmesini sağlar.

Yukarıdaki detaylı tasarım başlığı altında sistemin modüllerinden yola çıkarak sınıfları parça parça ele alınmıştı. Bu sınıf tasarımı bölümünde parça parça oluşturulan bu sınıf parçalarını birleştirerek sistemin genel sınıf tasarımı verilecektir.

Sınıf diyagramları kullanılarak oluşturulan sistemin genel sınıf tasarımı aşağıda verilmiştir.



Şekil 19:Genel Sınıf Tasarımı

Sistemin genel sınıf yapısı, online etkinlik rezervasyon sistemi projesinin temel bileşenlerini ve bu bileşenler arasındaki ilişkileri anlamamıza yardımcı olur. Şimdi, bu sınıf diyagramının detaylarına daha derinlemesine bir bakış yapalım. Aşağıda ana sınıflar üzerinden kısaca sınıf tasarımı özetlenmiştir.

1. User Sınıfı:

"User" sınıfı, sisteme kayıt olan ve etkinliklere katılan müşterileri temsil eder. Bu sınıf, temel kullanıcı bilgilerini (ad, soyad, e-posta vb.) içeren özelliklere sahiptir. Ayrıca, müşterilerin etkinliklere kaydolma, rezervasyon yapma, ödeme yapma ve değerlendirme gibi işlemleri gerçekleştirmek için gerekli olan metotları içerir.

2. MekanSahibi Sınıfı:

"MekanSahibi" sınıfı, etkinlikleri düzenleyen ve mekanlara sahip olan kullanıcıları temsil eder. Bu sınıf, mekân sahiplerinin profil bilgilerini içeren özelliklere sahiptir. Aynı zamanda, etkinlikler oluşturma, mekanları ekleme ve detayları (örneğin yemek ve personel açıklamaları) belirtme gibi işlevleri gerçekleştirmek için metotları içerir.

3. Product Sınıfı:

"Product" sınıfı, etkinliklerde sunulan ürünleri temsil eder. Bu sınıf, ürünlerin özelliklerini (ad, fiyat, açıklama vb.) içeren alanlara sahiptir. Etkinliklere eklenen ürünlerle ilgili işlemleri gerçekleştirmek için gerekli metotları içerir.

4. Order Sınıfı:

"Order" sınıfı, müşterilerin gerçekleştirdiği rezervasyon ve ödeme işlemlerini temsil eder. Bu sınıf, rezervasyon bilgilerini, ödeme detaylarını ve diğer ilgili bilgileri içeren özelliklere sahiptir. Rezervasyon ve ödeme işlemleriyle ilgili metotları içerir.

Sınıflar arasındaki ilişkiler, oklarla gösterilmektedir. Örneğin, "User" sınıfının "Order" sınıfı ile ilişkisi, müşterilerin rezervasyon ve ödeme işlemlerini gerçekleştirdiği bağlantıyı gösterir.

Bu sınıf tasarımı, projede yer alan temel bileşenleri ve bu bileşenler arasındaki etkileşimleri anlamamıza yardımcı olur. Sistemdeki görevlerin ve ilişkilerin bu şekilde görsel olarak temsil edilmesi, geliştirme ekibine projenin daha bütünlük kazanmış bir anlayışını sunar. Ayrıca, sınıflar arasındaki ilişkilerin netleştirilmesi, yazılım geliştirme sürecinin daha düzenli ve anlaşılır olmasına katkı sağlar.

**BÖLÜM 5: GERÇEKLEŞTİRİM**

Bu kısım da kullanılan IDE’ler , programlama dilleri, ORM’ler, teknolojiler, script çerçeveleri , veritabanları gibi yapılar anlatılacak olup daha sonrasında geliştirilen son kullanıcı ara yüzleri ekran görüntüleri alınarak, anlatılacaktır.

Aşağıda maddeler projede kullanılan yapılar detaylı bir şekilde verilmiştir.

1. Kullanılan IDE'ler:

Visual Studio Code: Visual Studio Code (VS Code), yazılım geliştirmek için oldukça popüler bir entegre geliştirme ortamıdır. Bu IDE'nin tercih edilme sebeplerinden biri, açık kaynaklı olmasıdır. Bu özelliği sayesinde, geliştiriciler IDE'nin iç yapısını anlayabilir, özelleştirebilir ve ihtiyaçlarına göre uyarlayabilirler. Ayrıca, geniş bir eklenti ekosistemine sahip olması, kullanıcıların çeşitli programlama dilleri ve framework'ler için eklentiler ekleyerek VS Code'u kendi ihtiyaçlarına uygun hale getirmelerini sağlar.

React.js ve Node.js uygulamaları geliştirme sürecinde VS Code'un sunduğu özellikler büyük avantajlar sağlar. React.js projeleri için JSX (JavaScript XML) desteği, kod otomatik tamamlama, hata ayıklama ve derleme özellikleri gibi avantajlar, geliştiricilere etkili bir şekilde çalışma imkanı tanır. Aynı şekilde, Node.js uygulamaları içinse paket yönetimi, hata ayıklama, ve uygulama izleme gibi özellikler, geliştirme sürecini kolaylaştırır.

VS Code'un kullanıcı dostu arayüzü ve hafif tasarımı, geliştiricilerin karmaşık projeler üzerinde rahatça çalışmalarına olanak tanır. Bu IDE, çoklu platform desteği ile Windows, macOS ve Linux işletim sistemlerinde sorunsuz bir şekilde çalışabilir. Ayrıca, entegre terminal özelliği sayesinde kullanıcılar, kodlarını yazarken aynı pencerede terminal komutlarını da kolayca çalıştırabilirler.

Sonuç olarak, Visual Studio Code, açık kaynaklı yapısı, geniş eklenti ekosistemi, kullanıcı dostu arayüzü ve zengin özellik seti ile modern web geliştirme projelerinde tercih edilen bir IDE olması gerekçesiyle proje de kullanılmıştır.

2. Programlama Dilleri:

JavaScript ve TypeScript: React.js ve Redux Toolkit gibi modern web uygulama geliştirme araçları, temelde JavaScript ve TypeScript dillerini kullanır. Bu diller, web tarayıcılarında çalışan ve kullanıcı arayüzlerini etkileşimli hale getirmeye odaklanan bu tür uygulamaların geliştirilmesinde yaygın olarak tercih edilir. JavaScript, tarayıcı ortamında çalışan bir programlama dilidir ve React.js'in temel dilidir. Web tarayıcıları, JavaScript'i doğrudan anlayabilen bir dil olarak çalıştırabilirler. Bu, kullanıcıların tarayıcıda etkileşimli ve dinamik web sayfalarını deneyimlemelerini sağlar. JavaScript'in esnek yapısı, tarayıcıda çalışan React uygulamalarının hızlı ve etkili bir şekilde çalışmasına olanak tanır. Ayrıca, React.js'in kullanımıyla birlikte, JSX (JavaScript XML) adı verilen bir dil uzantısı kullanılarak bileşen tabanlı bir geliştirme modeli benimsenir.

3. ORM (Object-Relational Mapping):

MongoDB kullanıldığından ORM kullanımına gerek duyulmamıştır. MongoDB, JSON benzeri BSON belgelerini depolayarak, veri tabanı işlemlerini JavaScript nesneleriyle doğrudan gerçekleştirmeye imkân tanımıştır.

4. Kullanılan Teknolojiler:

React.js: Kullanıcı arayüzü geliştirmek için kullanılan bu kütüphane, bileşen tabanlı yapısı ve sanal DOM mekanizması ile performanslı ve modüler bir arayüz sağlamıştır.

• Kullanım Alanı: React.js, kullanıcı arayüzü geliştirmek için kullanılan bir JavaScript kütüphanesidir. Facebook tarafından geliştirilmiştir ve genellikle tek sayfa uygulamaları (SPA) oluşturmak için tercih edilir.

• Bileşen Tabanlı Yapı: React, bileşen tabanlı bir yapı sunar. Bu, uygulamayı küçük, bağımsız ve tekrar kullanılabilir bileşenlere bölmeyi sağlar. Bu da kodun daha okunabilir, sürdürülebilir ve genişletilebilir olmasını sağlar.

• Sanal DOM Mekanizması: React, sanal DOM (Document Object Model) mekanizması kullanarak arayüz güncellemelerini optimize eder. Bu, sayfa performansını artırır ve daha hızlı kullanıcı deneyimi sunar.

Node.js: JavaScript tabanlı bir arka uç çalıştırma ortamıdır. Server-side (sunucu tarafı) işlemleri gerçekleştirmek için kullanılır. Özellikle hızlı, hafif ve ölçeklenebilir olmasıyla bilinir.

Express.js: Node.js üzerinde çalışan web uygulamaları oluşturmak için kullanılan minimalist ve esnek bir web uygulaması çerçevesidir. RESTful API'lerin oluşturulmasını kolaylaştırır ve HTTP isteklerini işlemek için etkili bir yapı sunar.

Redux Toolkit: React uygulamalarında state yönetimini kolaylaştıran ve Redux'taki tekrarlayan kod miktarını azaltan bir kütüphanedir. Uygulama durumu yönetimi bu kütüphane üzerinden başarıyla sağlanmıştır.

• State Yönetimi: Redux Toolkit, React uygulamalarında state yönetimini kolaylaştırmak için kullanılır. Redux mimarisini temel alır ve uygulama durumu (state) için merkezi bir depo sağlar.

• Tekrarlayan Kod Azaltma: Redux Toolkit, Redux'taki tekrarlayan kod miktarını azaltmayı hedefler. Özellikle, action, reducer ve store oluşturma süreçlerini basitleştirir, böylece geliştiricilere daha verimli bir şekilde çalışma imkanı sunar.

5. Veritabanları:

MongoDB: Belge tabanlı NoSQL veri tabanıdır. Veriler, BSON (Binary JSON) formatında belgeler halinde depolanır. MongoDB'nin ölçeklenebilir ve esnek yapısı, projenin veri tabanı ihtiyaçlarını karşılamada başarılı olmuştur. MongoDB'nin bu özellikleri, özellikle büyük ve karmaşık veri setleri ile çalışan projeler için uygundur. Belge tabanlı yapısı, geliştiricilere daha fazla esneklik sağlar ve MongoDB'nin ölçeklenebilir mimarisi, büyüyen veri trafiğine karşı yüksek performans ve güvenilirlik sunar.

• NoSQL Veri tabanı: MongoDB, NoSQL (Not Only SQL) veri tabanlarından biridir. NoSQL, ilişkisel veri tabanı modellerine alternatif olarak geliştirilen, genellikle daha esnek ve ölçeklenebilir veri tabanı çözümlerini ifade eder.

• Belge Tabanlı Yapı: MongoDB, belge tabanlı bir NoSQL veri tabanıdır. Veriler BSON (Binary JSON) formatında belgeler halinde depolanır. Bu, JSON benzeri bir veri formatını kullanarak veri tabanındaki bilgilerin daha anlaşılır ve okunabilir olmasını sağlar.

• Esnek ve Ölçeklenebilir Yapı: MongoDB'nin esnek veri modeli, farklı türdeki verileri aynı veri tabanında depolama yeteneği sağlar. Bu, geliştiricilere veri tabanı tasarımında daha fazla özgürlük tanır. Ayrıca, MongoDB'nin ölçeklenebilir mimarisi, büyüyen veri setleriyle başa çıkabilme yeteneği sunar.

• Dokümanlar ve Collects: MongoDB, belge koleksiyonları içinde belgeleri depolar. Bu belgeler, çeşitli alanlar ve değerler içeren yapılar olabilir. MongoDB, bu koleksiyonlar ve belgeler arasında hızlı ve etkili sorgular yapma yeteneği sağlar.

• Dinamik Şemalar: MongoDB, dinamik şemaları destekler. Bu, veri tabanındaki belgelerin yapısının esnek bir şekilde değiştirilebileceği anlamına gelir. Bu özellik, projelerin gereksinimlerine uyum sağlamak için kolaylık sunar.

6. Mobil Gerçekleştirme Sorunları:

React Native: Mobil uygulamanın geliştirilmesi amacıyla seçilen React Native, web tarafındaki başarıyı mobil platforma taşıma hedefiyle kullanılmıştır. Ancak, mobil geliştirme sürecinde karşılaşılan spesifik zorluklar neticesinde, projenin bu aşamasında mobil uygulama tamamlanamamıştır.

Geliştirilen son kullanıcı ara yüzleri, projenin kullanıcı deneyimini ve görsel tasarımını ön plana çıkaran önemli bir unsurdur. Bu ekran görüntüleri, kullanıcıların platformu etkili bir şekilde kullanmalarını sağlamak amacıyla tasarlanan arayüz öğelerini içermektedir. Şimdi, bu ekran görüntüleri aşağıda maddeler halinde tanıtılacaktır.

* Giriş yap ekranı

metin, ekran görüntüsü, yazılım, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 20:Giriş Yapma Ekranı

Kullanıcı Girişi (Login):

Giriş ekranında, kullanıcılardan email ve şifre bilgilerini girmeleri istenir. Bu, platforma daha önce kayıt olan kullanıcıların hesaplarına erişim sağlamalarını sağlar. Kullanıcılar, doğru bilgileri girdikten sonra sisteme başarılı bir şekilde giriş yapabilirler. Giriş yapma sürecinde kullanıcılara hata mesajları ve bilgilendirici metinler eklenerek, kullanıcı deneyimi daha kullanıcı dostu hale getirilmiştir

Şifre Unutma (Şifremi Unuttum):

Eğer kullanıcı şifresini unutmuşsa, "Şifremi Unuttum" kısmına tıklaması önerilir. Bu adım, kullanıcının kayıtlı email adresini girmesine olanak tanır. Ardından, sistem kullanıcıya şifre sıfırlama linki gönderir. Bu link üzerinden kullanıcı, yeni bir şifre belirleyebilir ve hesabına tekrar erişebilir. Bu işlem, kullanıcıların şifrelerini güvenli bir şekilde yeniden oluşturmalarına olanak tanır.

Yeni Kullanıcı Kaydı (Yeni Kullanıcı mısınız?):

Eğer kullanıcı daha önce bir hesap oluşturmadıysa, "Yeni Kullanıcı mısınız?" seçeneğine tıklaması istenir. Bu adım, kullanıcının temel bilgilerini (ad, , email, şifre ve avatar.) içeren bir kayıt formuna yönlendirir. Kullanıcı, formu doldurduktan sonra sisteme kaydolabilir.

* Kayıt ol ekranı

metin, ekran görüntüsü, dikdörtgen, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 21:Kayıt Ol ekranı

* Şifremi unuttum ekranı

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 22:Şifremi Unuttum Ekranı

* Mekân sahibi/organizatör ana ekran

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 23:Mekan Sahibi Ana Ekranı

1. Toplam Miktar Bölümü:

Bu bölümde, mekân sahibinin elde ettiği kazancın genel bir özeti yer almaktadır. Etkinlik satışlarından elde edilen toplam gelirin bu bölümde gösterildiği düşünülmektedir. Miktarlar, net bir finansal özet sunmak amacıyla konulmuştur.

2. Etkinlikler Bölümü:

Bu kısımda, mekân sahibinin düzenlediği veya düzenlediği etkinlikleri gösteren bir liste bulunmaktadır. Her bir etkinlik, isim, ücret, stok durumu ve diğer önemli bilgilerle birlikte detaylı bir şekilde listelenmiştir. Bu bölüm, mekânın etkinlik yönetimini kolaylaştırmak adına önemli bir bilgi kaynağıdır.

3. Siparişler Bölümü:

Siparişler bölümü, her bir etkinlik için yapılan bilet satışlarına dair detayları içermektedir. Burada yer alan bilgiler arasında, müşteri adı, bilet türü, miktar, ödeme durumu gibi önemli detaylar bulunmaktadır. Bu bölüm, mali tabloların oluşturulması ve müşteri ilişkilerinin yönetilmesi açısından kritik bir rol oynamaktadır.

4. Kullanıcılar Bölümü:

Etkinlik rezervasyon sistemine kaydolan tüm kullanıcıların bilgilerini içeren bu bölüm, mekan sahibine etkinliklere ilgi gösteren kitle hakkında genel bir bakış sunar. Kullanıcı adları, iletişim bilgileri ve rezervasyon geçmişleri gibi detaylar burada bulunabilir. Bu bilgiler, müşteri tabanını anlamak ve etkileşimleri yönetmek adına değerlidir.

5. Stok Bilgisi:

Ekranın en sonunda, etkinliğin sahip olduğu stok bilgileri bulunmaktadır. Bu bölümde, satılan biletlere göre güncel stok miktarları yer almaktadır. Bu, mekânın bilet yönetimi ve etkinlik kapasitesi planlaması için kritik öneme sahiptir. Bu bilgiler, gelecekteki etkinliklerin planlanması ve bilet satış stratejilerinin oluşturulması açısından önemli bir temel sağlamaktadır.

* Organizatör etkinlikleri görüntüleme

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 24:Organizatör etkinlikleri görüntüleme ekranı

1. Etkinlik Listesi:

Sayfanın merkezinde, oluşturulan tüm etkinliklere ait bir liste bulunmaktadır. Her etkinlik, benzersiz bir Kimlik (ID) numarası, isim, ücret ve stok bilgisi ile görüntülenmektedir. Bu bilgiler, mekân sahibine tüm etkinliklerin genel bir özetini sağlamak adına kullanılır.

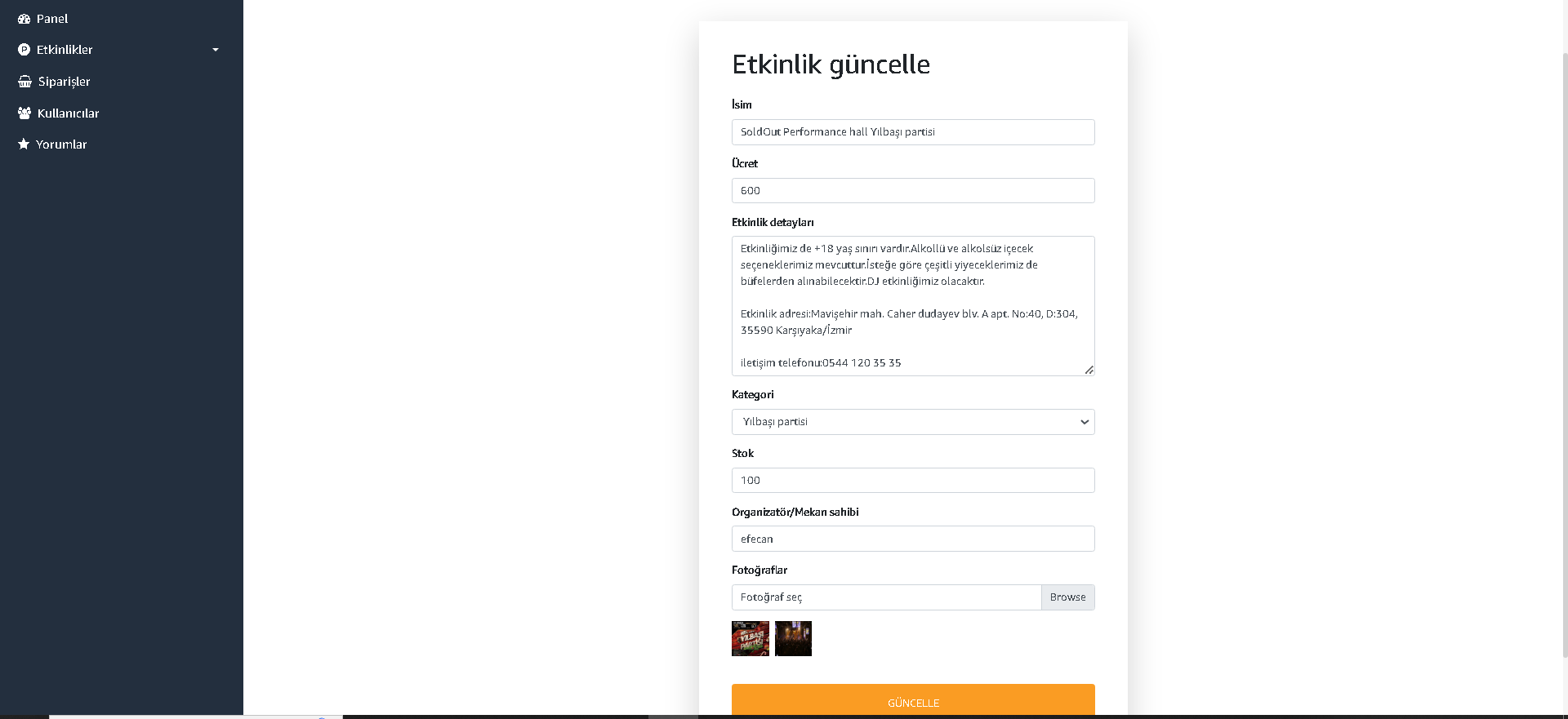
2. Hareketler Bölümü:

Listenin en sağında yer alan "Hareketler" kısmı, mekân sahibine etkinlikleri yönetme imkanı sunmaktadır. Bu bölümden, bir etkinliği silme veya düzenleme işlemleri gerçekleştirilebilir. "Sil" seçeneğiyle etkinlikleri kaldırabilir, ya da "Düzenleme" seçeneğine tıklayarak etkinliği güncelleme sayfasına yönlendirilebilirsiniz.

3. Etkinlik Silme:

"Hareketler" kısmındaki "Sil" seçeneği, mekân sahibine bir etkinliği kalıcı olarak kaldırma olanağı sağlar. Bu işlem, etkinlik planlaması üzerindeki değişikliklere ve güncellemelere olanak tanır. Ancak, bu adımın dikkatlice düşünülmesi ve gerekçelendirilmesi gerektiği unutulmamalıdır.

* Organizatör etkinlik güncelleme



Şekil 25:Organizatör Etkinlik Güncelleme Ekranı

Mekan sahibi yetkisine sahip kişilerin erişimine açık bir önceki sayfada belirlenen etkinliğe ait detaylar, oldukça kapsamlı bir biçimde düzenlenebilir. Seçilen etkinliğin adı ve etkinlik ücreti değiştirilmek istenirse bu kısımdan yapılabilir. Etkinlik kategorisi değiştirilmek isteniyorsa yine bu kısımdan değiştirilebilir.

Etkinlik detayları bölümü, mekân sahibinin veya yetkilisinin etkinliği daha iyi anlatması için ayrılmış bir alandır. Bu bölüm, etkinliğin öne çıkan özellikleri, katılımcılara sağlanacak avantajlar, etkinlikle ilgili kurallar ve diğer önemli bilgileri içerebilir. Mekân sahibi, bu açıklama bölümünü detaylandırarak etkinliğin atmosferini ve özel niteliklerini vurgulama şansına sahiptir.

Etkinliği düzenleyen organizatör bilgisi, mekân sahibinin bu alanda güncellemeler yapmasına olanak tanır. Organizatörle ilgili bilgiler, iletişim bilgileri ve organizatörün profili gibi unsurlar burada yer alır. Bu bilgiler, katılımcılara organizatörle iletişime geçme ve etkinlikle ilgili daha fazla bilgi edinme imkânı sağlar.

Ayrıca, etkinliğe ait sanatçı veya mekân fotoğrafları da bu sayfada güncellenebilir. Bu fotoğraflar, etkinliği görsel olarak zenginleştirir ve potansiyel katılımcılara etkinlik hakkında daha iyi bir izlenim sunar.

Sonuç olarak, mekân sahibi yetkisine sahip kişiler, bu önceki sayfa üzerinden seçmiş olduğu etkinlikle ilgili birçok detayı özelleştirebilir ve güncelleyebilir. Bu özelleştirmeler, etkinliği daha çekici hale getirerek katılımcıları etkileme ve etkinliği daha başarılı kılma konusunda önemli bir rol oynar.

* Organizatör yeni etkinlik oluşturma

metin, yazılım, multimedya yazılımı, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 26:Organizatör Yeni Etkinlik Oluşturma Ekranı

1. İsim:

En üstteki bölümde, etkinliğin ismi ve düzenleneceği mekânın adı yazılabilir. Bu bölümde, etkinliği tanımlayan özel bir ad belirlenir.

2. Ücret:

Bu bölümde etkinlik için belirlenen ücret ifade edilir. Katılımcılardan talep edilen maliyeti belirlemek amacıyla kullanılır. Ücret, etkinliğin türüne ve içeriğine bağlı olarak farklılık gösterebilir.

3. Etkinlik Detayları:

Bu kısımda, etkinlikle ilgili detaylar sunulur. Adres, iletişim numarası gibi iletişim bilgileri, etkinlikle ilgili özel detaylar (örneğin, etkinliğin yemek ve içecek içerip içermediği) bu bölümde paylaşılır. Katılımcılar için önemli bilgilerin yer aldığı alandır.

4. Kategori Seçimi:

Etkinliğin ait olduğu kategori buradan seçilir. Parti, konser, tiyatro gibi farklı kategoriler arasından uygun olanı seçilerek, etkinliğin doğru hedef kitleye ulaşması sağlanır.

5. Stok (Kontenjan):

Bu kısımda etkinliğin kaç kişilik kontenjanı olduğu belirlenir. Belirli bir sınırla sınırlı olan etkinliğe kaç kişinin katılabileceği bu alandan kontrol edilir.

6. Organizatör/Mekân Sahibi:

Etkinliği düzenleyen organizatörün veya mekân sahibinin adı bu bölüme girilir. Bu sayede etkinliğin sorumlusu net bir şekilde belirlenir.

7. Fotoğraflar:

Etkinliğe ait mekân ve varsa sanatçının fotoğrafları burada eklenir. Bu, kullanıcılara görsel bir deneyim sunarak etkinliği daha çekici hale getirir.

En altta bulunan "Oluştur" butonuna tıklanarak, girilen bilgilerle etkinlik resmi olarak oluşturulur ve kullanıcılar tarafından görüntülenebilir hale gelir. Bu sayede etkinlikle ilgili tüm bilgiler düzenli bir şekilde paylaşılarak katılımcıların etkinliği daha iyi anlamaları ve buna göre bilet almaları sağlanır.

* Organizatörün kullanıcılar listesini görüntülemesi ve rol belirleme

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 27:Organizatör Kullanıcı listesi ekranı

Bu sayfa, sisteme kayıtlı olan tüm kullanıcıları içeren bir liste sunar. Her bir kullanıcı için ID, isim, email ve rol bilgileri görüntülenir. Bu bilgiler, kullanıcı yönetimi ve sistemdeki rolleri belirleme açısından önemlidir.

Kullanıcı Silme:

Bu seçenekle, istenmeyen kullanıcılar sistemden silinebilir. Bu işlem dikkatlice yapılmalıdır, çünkü silinen kullanıcıların verileri kalıcı olarak kaybolabilir. Silme işlemi öncesinde kullanıcının rolleri ve bu rolle ilişkili veriler gözden geçirilmelidir.

Düzenleme Bölümü:

Bu seçenek, belirli bir kullanıcının detaylarını güncellemek için kullanılır. Kullanıcının ismi, email adresi veya rolü gibi bilgiler bu bölümde düzenlenebilir. Bu özellik, kullanıcıların bilgilerini güncel ve doğru tutmak için kullanılır.

Bu sayfa, kullanıcı yönetimi açısından önemli bir araçtır ve sistem yöneticilerine tüm kullanıcıların bilgilerini tek bir noktadan yönetme imkanı sunar. Bu tür bir arayüz, kullanıcı verilerinin güvenliği ve bütünlüğü açısından kritik bir rol oynar. Ayrıca, sistemi kullananların rol ve yetkilerini düzenlemek için kullanışlı bir araçtır.

* Organizatör tarafından Kullanıcı detayları güncelleme

metin, yazılım, multimedya yazılımı, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 28:Organizatör Kullanıcı Detayları Ekranı

1. Ad Güncelleme:

Kullanıcının adını değiştirmek için bu alanda yeni ad girilebilir. Bu özellik, kullanıcının kişisel bilgilerini güncel tutma ve doğru bir şekilde görüntüleme imkanı sağlar.

2. Email Güncelleme:

Kullanıcının email adresi bu alanda güncellenebilir. Yeni bir email adresi eklemek veya mevcut olanı değiştirmek için ilgili bölüme yeni bilgiler girilir.

3. Rol Güncelleme:

Kullanıcının rolü buradan değiştirilebilir. Rol, kullanıcının sistem içindeki yetkilerini belirler. Özellikle belirtilmiştir ki, kullanıcı rolü olanlar etkinlik oluşturma ve belirli panellere erişme yetkisine sahip değildir. Ancak, rolü admin olanlar etkinlik oluşturma ve panellere erişim yetkilerine sahiptir.

4. Rol Bazlı Yetkilendirme:

Kullanıcı Rolü: Kullanıcıların rolü, sistem içindeki yetkilerini belirler. Ekranda belirtilmiş olan rol bazlı yetkilendirmede, "Kullanıcı" rolüne sahip kullanıcıların etkinlik oluşturma ve özel panellere erişim yetkisi yoktur. "Admin" rolüne sahip kullanıcılar ise bu yetkilere sahiptir. Bu, sistem yöneticilerinin belirli görevlere odaklanabilmesini ve güvenli bir kullanıcı yönetimi sağlamasını amaçlar.

5. Yetki Kontrolü:

Kullanıcının etkinlik oluşturma ve belirli panellere erişim yetkileri, belirli bir rolle ilişkilendirilmiştir. Bu, sistemi daha güvenli ve yönetilebilir hale getirir.

Bu sayfa, kullanıcıların bilgilerini güncelleme işlemlerini basitleştirerek, sistem yöneticilerine rol bazlı yetkilendirme ile güvenli bir kullanıcı yönetimi sağlama imkânı sunar. Bu, sistem içindeki farklı kullanıcı gruplarına özgü yetkilerin etkin bir şekilde yönetilebilmesini sağlar.

* Organizatör etkinlik yorumları görüntüleme

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 29:Organizatör Etkinlik Yorumları Görüntüleme Ekranı

1. Arama Butonu:

Organizatör, girdiği ID bilgisini kullanarak etkinliklere ait yorum ve puanları aratabilir. Bu, belirli bir etkinlikle ilgili geri bildirimleri odaklı bir şekilde inceleme imkânı sunar.

2. Yorum ve Puan Gösterimi:

-Puanlama: Kullanıcılar tarafından verilen puanlar, genellikle 5 üzerinden değerlendirme olarak gösterilir. Organizatör, bu puanlamalar sayesinde etkinliklerin genel memnuniyet düzeyini anlayabilir.

-Yorumlar: Kullanıcılar tarafından bırakılan detaylı yorumlar, etkinliklerin güçlü ve zayıf yönlerini anlamak için önemli bir kaynaktır. Bu bölüm, her bir yorumun detaylı içeriğini sunar.

3. Kullanıcı Adı:Yorumu bırakan kullanıcının ismi, geri bildirimi daha kişisel hale getirir ve organizatöre hangi katılımcının ne düşündüğünü anlama fırsatı sağlar.

Yorum Silme İşlemi:Organizatör, sağ kısımda bulunan özel bölümde istediği yorumu silme yetkisine sahiptir. Bu, uygunsuz veya istemediği bir yorumu kaldırmak için kullanabilir.

Bu sayfa, organizatörlerin etkinliklere dair katılımcı geri bildirimlerini anlamalarına olanak tanır. Aynı zamanda, sayfa organizatörlere etkinliklerini iyileştirmek ve katılımcı memnuniyetini artırmak için önemli bilgiler sağlar.

Buradan itibaren kullanıcı ile ilgili ekran görüntüleri verilip özellikler açıklanacaktır.

* Kullanıcı profil ekranı

metin, ekran görüntüsü, web sayfası, web sitesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 30:Kullanıcı Profil Ekranı

Kullanıcı Profil Sayfası:

Bu sayfa, kullanıcının kişisel bilgilerini görüntülemesi ve güncellemesi amacıyla tasarlanmıştır. Kullanıcının profiline dair önemli özellikler şunlardır:

Temel Bilgiler:

1. İsim ve Mail Adresi:Kullanıcı, profil sayfasında ismini ve kayıtlı olduğu mail adresini görebilir. Bu, kişisel bilgilerin doğruluğunu kontrol etmek ve gerekirse güncellemek için kullanılır.

2. Kayıt Tarihi:Kullanıcının sisteme kaydolduğu tarih bu bölümde görüntülenir. Bu, kullanıcının sisteme ne zaman katıldığını takip etmesine olanak tanır.

Profil Düzenle Butonu:Sol kısımda bulunan "Profil Düzenle" butonuna tıklanarak kullanıcı, kişisel bilgilerini güncelleyebilir.

Şifreyi Değiştir Butonu:Sağ kısımda bulunan "Şifreyi Değiştir" butonuna tıklanarak kullanıcı, mevcut şifresini güncelleyebilir.

* Kullanıcı profili güncelleme

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 31:Kullanıcı Profil Güncelleme Ekranı

Kullanıcı, profilini güncellemek amacıyla kullanıcı panelinde yer alan "Profil Güncelle" sayfasına yönelir. Bu sayfada, mevcut bilgilerini görebilir ve istediği değişiklikleri yapabilir. İlk olarak, adını güncellemek isterse, "İsim" bölümüne yeni adını girer. Ardından, e-posta adresini güncellemek için "Email" bölümüne yeni e-posta adresini ekler. Profil fotoğrafını güncellemek istiyorsa, "Avatar" bölümünden yeni bir fotoğraf yükler. Tüm güncellemeleri tamamladıktan sonra, "Güncelle" butonuna tıklar. Bu butona tıkladığında, sistem yeni bilgileri kaydeder ve kullanıcının profilindeki değişiklikler aktifleşir.

* Kullanıcı şifre güncelleme ekranı

metin, ekran görüntüsü, yazılım, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 32:Kullanıcı Şifre Güncelleme Ekranı

Kullanıcı, hesap güvenliğini sağlamak amacıyla mevcut şifresini güncellemek için "Şifre Güncelle" sayfasına yönelir. Bu sayfada, mevcut şifresini güncellemek istediği hesabına giriş yapar. Ardından, "Mevcut Şifre" bölümüne mevcut şifresini gizli karakterlerle girer. Güvenli bir şifre oluşturmak adına, "Yeni Şifre" alanına ise yeni kullanmak istediği şifreyi girer.

Tüm bilgilerin doğru olduğunu teyit ettikten sonra, "Şifreyi Güncelle" butonuna tıklar. Bu butona tıkladığında, sistem mevcut şifreyi yeni belirlenen şifreyle günceller.

Bu güncelleme işlemi, hesap güvenliğini sağlamak ve kullanıcının hesap bilgilerini güncel tutmak adına önemli bir adımdır. Kullanıcı, şifresini başarıyla güncelledikten sonra, daha güvenli bir şekilde hesabını kullanmaya devam edebilir.

* Kullanıcı etkinlik görüntüleme

metin, ekran görüntüsü, web sitesi, çevrimiçi reklamcılık içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 33:Kullanıcı Etkinlik Görüntüleme Ekranı

Bu sayfa, kullanıcının platformdaki etkinliklere erişim sağladığı merkezi bir noktadır. Sayfa, kullanıcıya çeşitli etkinlikleri görüntüleme ve istediği etkinliği hızlıca bulma imkânı sunar.

Kullanıcı, bu bölümden tüm etkinlikleri gözden geçirebilir ve ilgisini çekenleri seçebilir.

Arama İmkânı: Arama çubuğu, kullanıcının istediği etkinliği hızlıca bulmasını sağlar. Kullanıcı, etkinlik adını arama çubuğuna girerek ilgili etkinliği kolayca bulabilir. Otomatik tamamlama özelliği, kullanıcının etkinlik adını doğru ve hızlı bir şekilde girmesine yardımcı olur.

Kullanım Kolaylığı:Sayfa tasarımı, kullanıcının etkinliklere kolay ve hızlı bir şekilde erişimini sağlamak üzere optimize edilmiştir. Büyük ve okunabilir yazılar, net ve düzenli bir arayüz, kullanıcı dostu bir deneyim sunar.

Bu sayede, kullanıcılar platformdaki etkinlikleri görüntüleyebilir, arayabilir ve kendi ilgi alanlarına uygun etkinliklere kolayca erişebilirler. Bu kullanıcı odaklı arayüz, etkinlikleri daha etkili bir şekilde keşfetmelerine olanak tanır.

* Kullanıcı etkinlik filtreleme ve arama

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 34:Kullanıcı Etkinlik Filtreleme ve Arama Ekranı

Bu etkinlik arama ve filtreleme sayfası, kullanıcılara özelleştirilmiş bir etkinlik keşfi deneyimi sunmak üzere tasarlanmıştır. Kullanıcılar, arama çubuğunu kullanarak istedikleri etkinliği hızlıca bulabilir ve aynı zamanda çeşitli filtreleme seçeneklerini kullanarak arama sonuçlarını daha da özelleştirebilirler.

Filtreleme Seçenekleri: Sol tarafta bulunan filtreleme paneli, kullanıcılara etkinlikleri daha spesifik bir şekilde sınıflandırma imkânı sunar.

Kullanıcılar, istedikleri fiyat aralığını girebilir, tercih ettikleri kategoriyi seçebilir ve buna ek olarak etkinlik puanlarına göre filtreleme yapabilirler. Bu filtreleme seçenekleri, kullanıcının bireysel tercihlerine uygun etkinlikleri bulmalarına yardımcı olur.

Bu özellikler, kullanıcılara daha fazla esneklik ve kişiselleştirme sağlayarak, platformdaki etkinlikleri keşfetmelerini ve en uygun olanları bulmalarını kolaylaştırır. Bu sayede, kullanıcılar istedikleri kriterlere uygun etkinlikleri bularak etkileşimlerini daha anlamlı hale getirebilirler.

* Etkinlik Detay görüntüleme, bileti sepete ekleme ve yorum paylaşma

metin, ekran görüntüsü, web sitesi, insan yüzü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 35:Etkinlik Detayları Görüntüleme Ekranı

Etkinlik Detayları ve Bilet İşlemleri:

Kullanıcı, ilgisini çeken bir etkinlik bulduğunda, etkinlik kartına tıklayarak detaylı bilgilere erişebilir. Etkinlik sayfasında, etkinliği detayları, adresi, iletişim numarası ve varsa yiyecek içecek bilgileri bulunmaktadır. Ayrıca, kullanıcının ilgisini çekebilecek görseller ve etkinlikle ilgili ek detaylar da bulunabilir.

Bilet Durumu Kontrolü:

Etkinlik sayfasında, kullanıcı stokta bulunan bilet sayısını görebilir. Bu, kullanıcının hızlı bir şekilde etkinlik için bilet temini yapabilmesine olanak tanır. Eğer bilet stokta varsa, kullanıcı istediği miktarda bilet seçip "Sepete Ekle" butonuna tıklayarak sepetine ekleyebilir.

Yorum ve Görüş Paylaşımı:

Etkinlik sayfasının altında bulunan "Yorumunu Paylaş" butonu, kullanıcının etkinlikle ilgili görüşlerini paylaşmasına olanak tanır. Kullanıcı, etkinliği deneyimledikten sonra bu bölümden kendi yorumunu yazabilir ve diğer kullanıcılarla paylaşabilir. Bu, kullanıcıların etkinlik hakkında bilgi alışverişinde bulunmalarını ve platform içinde etkileşimde bulunmalarını sağlar.

* Kullanıcı sipariş oluşturma

Şekil 36:Kullanıcı Sipariş Oluşturma Ekranı

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu sayfadan tamamla diyerek bilet faturasının ulaşacağı kargo adresi sayfasına yönlendirilir.

metin, ekran görüntüsü, yazılım, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 37:Kullanıcı Kargo Bilgileri Ekranı

Kullanıcı, biletleri satın almak için önce kargo bilgilerini girmelidir. "Kargo Bilgileri" sayfasında, kullanıcı, biletlerin ulaştırılacağı adresi belirtir. Bu aşamada, adı, telefon numarası, şehir, posta kodu ve ülke bilgilerini doğru ve eksiksiz bir şekilde girmesi beklenir. Bu bilgiler, biletlerin güvenli ve doğru bir şekilde kullanıcıya ulaştırılabilmesi için önemlidir. Daha sonrasına devam et butonuna tıklayarak siparişi onayla sayfasına ulaşır.

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 38:Kullanıcı Sipariş Onaylama Ekranı

Bu aşamada, kullanıcının fatura detayları yer alır. Kullanıcı, fatura üzerinde yer alacak isim, telefon numarası ve adres bilgilerini girebilir. Bu bilgiler, biletin ulaştırılacağı veya etkinlikle ilgili diğer iletişimlerin gerçekleştirileceği adresi içerir.

Sepet ürünler: Sayfanın alt kısmında, kullanıcı sepetindeki etkinliği detaylı bir şekilde inceleyebilir. Burada, seçilen etkinliğin adı ve bilet adedi gibi bilgiler yer alır. Kullanıcı, bu aşamada bilet sayısını görüntüleyebilir.

Sipariş Özeti ve Toplam Ödeme: Sağ tarafta, kullanıcı sipariş özetini görüntüleyebilir. Bu bölümde, etkinlik fiyatı, vergiler ve diğer ek ücretler dâhil olmak üzere toplam ödemesi gereken tutarı görebilir. Etkinlik fiyatı haricindeki ek ücretler, vergiler veya işlem ücretleri gibi detaylar kullanıcıya net bir ödeme özeti sunar.

Kullanıcı, tüm bilgilerin doğru olduğundan emin olduktan sonra "Ödemeye devam et" butonuna tıklayarak kart bilgilerini gireceği sayfaya yönlendirilir.

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 39:Kullanıcı Ödeme Ekranı

Son olarak bu sayfa üzerinden Kart numarası, kartın son kullanma tarihi ve kart CVC numarasını girerek ödemeyi tamamlar.

* Kullanıcı siparişleri görüntüleme

ekran görüntüsü, metin, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 40:Kullanıcı Sipariş Geçmişi Ekranı

Kullanıcı siparişlerini oluşturduktan sonra anlık durumunu, kargo hareketlerini ve ödediği tutarı bu sayfa aracılığıyla görüntüleyebilir.

**BÖLÜM 6: SONUÇ**

Bu araştırma, online etkinlik rezervasyon sistemlerinin geliştirilmesi ve yönetilmesi üzerine odaklanarak, etkinlik organizatörleri ile katılımcılar arasındaki etkileşimi kolaylaştırmayı hedefleyen bir çözümü detaylı bir şekilde incelemiştir. Geliştirilen bu sistem, etkinlik düzenleme süreçlerini daha verimli ve kullanıcı dostu bir deneyimle buluşturmayı amaçlamıştır.

Elde edilen bulgular, geliştirilen online rezervasyon sisteminin, organizatörlerin etkinliklerini daha etkili bir şekilde yönetmelerine olanak tanıdığını ve aynı zamanda katılımcıların da etkinliklere daha kolay ulaşmalarını sağladığını açıkça göstermektedir. Organizatörler, bu sistem aracılığıyla etkinlik detaylarını düzenleyebilir, güncelleyebilir ve katılımcı geri bildirimlerini toplayarak gelecekteki etkinlikleri daha iyi planlayabilirler.

Geliştirilen çözümün, etkinlik planlama süreçlerinde meydana gelen zaman kayıplarını önemli ölçüde azalttığı ve katılımcıların etkinliklere daha rahat erişim sağlamalarına olanak tanıdığı belirlenmiştir. Ayrıca, kullanıcı dostu arayüzü sayesinde etkinlik organizatörleri ve katılımcılar arasındaki iletişimi güçlendirdiği ve etkinlik deneyimini iyileştirdiği gözlemlenmiştir.

Kullanıcılar açısından, online rezervasyon sistemleri sunduğu avantajlarla etkinliklere katılım deneyimini önemli ölçüde zenginleştirmekte ve kullanıcı memnuniyetini artırmaktadır. Bu sistemlerin getirdiği çeşitli kolaylıklar, katılımcıların etkinliklere daha kolay erişim sağlamalarına olanak tanımaktadır.

Öncelikle, online rezervasyon sistemleri, kullanıcılara etkinliklere hızlı bir şekilde göz atma ve filtreleme seçenekleriyle kişisel tercihlerine uygun etkinlikleri bulma imkânı sunmaktadır. Bu, katılımcıların ilgi alanlarına ve programlarına daha uygun etkinlikleri seçmelerini sağlayarak, etkinlik deneyimini özelleştirmelerine yardımcı olur.

Geliştirilen rezervasyon sisteminin bir diğer önemli avantajı, kullanıcıların kolayca rezervasyon yapabilmeleri ve bu rezervasyonları takip edebilmeleridir. Kullanıcılar, istedikleri etkinliklere kolayca yer ayırtabilir ve rezervasyon takibi yaparak etkinlikle ilgili güncel bilgilere ulaşabilirler. Bu da katılımcılara planlama konusunda güven ve rahatlık sağlar.

Kullanıcıların etkinliklere katılım deneyimini zenginleştiren bir diğer unsuru da yorum paylaşma imkanıdır. Online rezervasyon sistemleri, katılımcılara etkinlik sonrasında deneyimlerini paylaşma ve diğer kullanıcılara geribildirimde bulunma şansı tanır. Bu, gelecekteki etkinliklere katılacak kişiler için değerli bir referans kaynağı oluşturur ve etkinlik düzenleyicilere katılımcı geri bildirimleriyle etkinliklerini geliştirme fırsatı sağlar.

Tüm bu avantajlar, online rezervasyon sistemlerinin katılımcıların etkinliklere daha bilinçli bir şekilde katılmalarına olanak tanıdığını göstermektedir.

Bu projenin başarılı bir şekilde tamamlanması için web tabanlı kısım için çeşitli güçlü teknolojiler tercih edilmiştir. React.js, kullanıcı arayüzünün etkileşimli ve hızlı olmasını sağlamak amacıyla kullanılmıştır. Express.js ve Node.js, sunucu tarafı işlemleri hızlandırarak sistem performansını artırmıştır. MongoDB ise veritabanı olarak seçilerek, etkinlik ve kullanıcı verilerinin etkili bir şekilde depolanmasına ve yönetilmesine olanak tanımıştır. Redux Toolkit ise uygulama durumu yönetimi için kullanılarak, veri akışının daha organize ve öngörülebilir bir şekilde gerçekleşmesine katkı sağlamıştır.

Bu teknolojilerin seçimi, projenin güvenilirlik, performans ve genişletilebilirlik açılarından sağlam temellere dayanmasını sağlamıştır. Ayrıca, React Native teknolojisinin planlanmış olması, projenin gelecekteki aşamalarında mobil uygulama geliştirmenin de mümkün olabileceğini göstermektedir.

Ancak, mobil uygulama geliştirme aşamasının henüz gerçekleştirilememiş olması, gelecekteki çalışmaların odaklanabileceği bir alan olarak belirtilmiştir. React Native'in entegrasyonu, mobil kullanıcıların daha geniş bir kitleye ulaşabilmesine ve sistem kullanımının artmasına katkıda bulunabilir.

Web kısmında başarılı bir şekilde çalışan proje, etkileyici bir performans sergileyerek kullanıcı dostu bir arayüz sunmaktadır. Organizatörler, etkinliklerini kolayca oluşturabilir, düzenleyebilir ve takip edebilirler. Aynı zamanda, kullanıcılar da bu etkinliklere sorunsuz bir şekilde rezervasyon yapabilmekte ve katılım sağlayabilmektedirler.

Sistem, etkinlikleri oluşturma, katılma ve yönetme gibi organizasyonel süreçleri başarıyla yönetirken, kullanıcıların güvenli bir şekilde ödeme yapmalarını ve etkinliklere katılmalarını sağlayan entegre bir ödeme sistemi ile desteklenmektedir.

Kullanılan teknolojilerin uyumlu bir şekilde entegre edilmesi, projenin performansını olumlu yönde etkilemektedir. React.js ve Redux Toolkit, kullanıcı arayüzünün dinamik ve kullanıcı dostu olmasına katkı sağlarken, Express.js ve MongoDB sayesinde güçlü bir backend altyapısı oluşturulmuştur.

Projenin gelecekteki çalışma dönemlerinde, öncelikli olarak mobil kısmın detaylı bir şekilde planlanması ve uygulanması gerekmektedir. React Native teknolojisinin projeye entegre edilmesi, kullanıcıların daha geniş bir platformda etkinliklere erişmelerine imkân tanıyacaktır. Bu da projenin kullanıcı tabanını artırarak daha etkili bir online etkinlik yönetim platformu olmasını sağlayacaktır.

Ayrıca, kullanıcı geri bildirimleri üzerinden sistemin geliştirilmesi, yeni özelliklerin eklenmesi ve mevcut işlevselliğin optimize edilmesi de önemli bir gelecek çalışma alanını oluşturmalıdır. Bu şekilde, projenin rekabet avantajı elde etmesi ve kullanıcıların beklentilerini karşılaması daha mümkün hale gelecektir.

# **REFERANSLAR**

Dokümanda başvurulan her tür kaynak (internet sayfası dahil) referans gösterilerek kullanılmalıdır. Doküman içerisinde referanslar [x] şeklinde, dokümanda geçtiği sıraya göre numaralandırılarak gösterilmelidir.

# Başvurular

Anonim. (2018, 08 12). *theastrologypage*. theastrologypage: https://tr.theastrologypage.com/data-dictionary#:~:text=Veri%20s%C3%B6zl%C3%BC%C4%9F%C3%BC%2C%20bir%20veritaban%C4%B1n%C4%B1n%20meta,ili%C5%9Fkisel%20veritaban%C4%B1n%C4%B1n%20%C3%B6nemli%20bir%20bile%C5%9Fenidir adresinden alındı [4]

Anonim. (2019, 02 20). *VisureSolutions*. VisureSolutions: https://visuresolutions.com/tr/blog/functional-requirements/#:~:text=İşlevsel%20gereksinim%2C%20bir%20sistemin%20nasıl,kullanıcının%20algıladığı%20özellikler%20olarak%20düşünülebilir adresinden alındı [2]

Anonim. (2022, 07 04). *creately*. creately: https://creately.com/tr/usage/risk-matrisi-%C5%9Fablonu/#:~:text=Risk%20matrisi%2C%20risk%20de%C4%9Ferlendirmesi%20i%C3%A7in,etki%20matrisi%20olarak%20da%20bilinir adresinden alındı [8]

Arslan, E. (2019, 11 20). *cubicl*. cubicl: https://blog.cubicl.io/gantt-cizelgesi adresinden alındı [7]

Güneş, V. (2017, 10 25). *medium*. medium: https://medium.com/@veysel.gunes36/use-case-kullanım-şekli-diyagramları-ile-resmin-bütününü-görmek-fbe4f49494f0 adresinden alındı [5]

Güneş, V. (2020, 23 02). *medium*. medium: https://medium.com/@veysel.gunes36/use-case-kullan%C4%B1m-%C5%9Fekli-diyagramlar%C4%B1-ile-resmin-b%C3%BCt%C3%BCn%C3%BCn%C3%BC-g%C3%B6rmek-fbe4f49494f0 adresinden alındı [6]

rezervem.com. (2021, 09 18). *Rezervem*. Rezervem: https://www.rezervem.com.tr/blog/online-rezervasyon-sistemi-ile-rakiplerinizin-bir-adim-onunde-olun adresinden alındı [1]

VisureSolutions. (2019, 02 20). *VisureSolutions*. VisureSolutions: https://visuresolutions.com/tr/requirements-management-traceability-guide/non-functional-requirements/#:~:text=İşlevsel%20Olmayan%20Gereksinimler%20(NFR'ler,daha%20pek%20çok%20sorunla%20ilgilenir adresinden alındı [3]