

Şirket İçi – Bilgi İşlem Talep Yönetim (Ticket) Sistemi

Yazılım Mühendisliği Ana Bilim Dalı

Tezsiz Yüksek Lisans / Bitirme Projesi

Murat Merdoğan

Y210240055

Tez Danışmanı:

Prof. Dr. Ayşegül Alaybeyoğlu

Haziran 2023

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Murat Merdoğan** tarafından hazırlanan **Şirket İçi – Bilgi İşlem Talep Yönetim Sistemi** başlıklı bu çalışma tarafımızca okunmuş olup, kapsam ve nitelik açısından başarılı bulunarak YÜKSEK LİSANS / PROJE BİTİRME ÖDEVİ olarak kabul edilmiştir.

**ONAYLAYANLAR:**

**Proje Danışmanı: Prof. Dr. Ayşegül Alaybeyoğlu**

İzmir Kâtip Çelebi Üniversitesi

Şirket İçi – Bilgi İşlem Talep Yönetim Sistemi

# Öz

Bu proje, işletmelerin karşılaştığı bilgi işlem taleplerinin etkin ve verimli bir şekilde yönetilmesi sağlamak amacıyla geliştirilecektir. Günümüz iş dünyasında şirketlerin verimliliklerini artırmak, iş süreçlerini optimize etmek ve müşteri memnuniyetlerini sağlamak için bilgi işlem hizmetlerine ihtiyaç duydukları bir gerçektir. Bu taleplerin yönetiminde bazı sorunlar ortaya çıkabilmektedir. Bu sorunlardan bazıları, taleplerin yanlış anlaşılması, takibin güçlüğü, gecikmeler tekrarlanan talepler ve etkili bir raporlama sürecinin eksikliği gibi konuları içerir.

Bu proje ile bilgi işlem talepleri kâğıt üzerindeki formlarla veya e-postalarla iletilmek zorunda olmayacaktır. Kolay erişebilir çevrim için bir masaüstü uygulama talepler oluşturulabilecektir, takip edilebilecek ve güncel durumlarını kontrol edebileceksiniz. Sistem aynı zamanda taleplerin önceliklendirilmesini sağlayacak ve en uygun uzmanlara yönlendirme sağlayarak hızlı çözümlerle geliştirilmesini sağlayacaktır.

**Anahtar kelimeler:** c#, ittalepyönetimi, webapi, yazılım

IT Help Desk Management Software

# Abstract

This project aims to develop a solution for effectively and efficiently managing the information technology (IT) requests faced by businesses. In today's business world, it is a known fact that companies require IT services to enhance their productivity, optimize their business processes, and ensure customer satisfaction. However, there can be challenges in managing these requests. Some of these challenges include misunderstanding of requests, difficulty in tracking them, delays, repeated requests, and a lack of an effective reporting process.

With this project, IT requests will no longer need to be communicated through paper forms or emails. A desktop application will be developed, providing an easily accessible online platform where users can create, track, and monitor their requests. The system will also prioritize the requests and direct them to the most suitable experts, enabling prompt resolution.

**Keywords:** c#, ithelpdesk, webapi, software

# Teşekkür

Yüksek Lisans Bitirme Projemi hazırlarken kıymetli bilgi ve tecrübeleri ile bana yol gösterici ve destek olan danışman hocam Sayın Prof. Dr. Ayşegül ALAYBEYOĞLU’na teşekkür ederim. Ayrıca çalışmalarım boyunca destekleri ile beni hiçbir zaman yalnız bırakmayan anneme minnettarım.

# İçindekiler

[Öz ii](#_Toc136902038)

[Abstract iii](#_Toc136902039)

[Teşekkür iv](#_Toc136902040)

[İçindekiler v](#_Toc136902041)

[Şekiller Listesi vii](#_Toc136902042)

[Kısaltmalar Listesi viii](#_Toc136902043)

**1.** [**Talep Yönetim Sistemlerine Bakış** 1](#_Toc136902045)

[1.1 Change Management (Değişiklik Yönetimi) 2](#_Toc136902046)

[1.2 Request Management (İstek Yönetimi) 3](#_Toc136902047)

[1.3 Project Management (Proje Yönetimi) 5](#_Toc136902048)

[1.4 Problem Management (Problem Yönetimi) 6](#_Toc136902049)

[1.5 Devops (Sürekli Entegrasyon) 8](#_Toc136902050)

**2.** [**Geliştirilen Sistem** 10](#_Toc136902052)

[2.1 Gereklilik Sebepleri 10](#_Toc136902053)

[2.2.1 İş Sürekliliği 10](#_Toc136902054)

[2.2.2 İş Verimliliği 10](#_Toc136902055)

[2.2.3 İşbirliği ve İletişim 10](#_Toc136902056)

[2.2.4 Veri Analizi ve İyileştirme 11](#_Toc136902057)

[2.2 Arayüzler 11](#_Toc136902058)

[2.2.1 Giriş Ekranı 11](#_Toc136902059)

[2.2.2 Menü Ekranı 12](#_Toc136902060)

[2.2.3 Talep Listesi Ekranı 12](#_Toc136902061)

[2.2.4 Yeni Talep Ekranı 13](#_Toc136902062)

[2.2.5 Yönetici Talep Listesi 14](#_Toc136902063)

[2.2.6 Yönetici Talep Kapatma Ekranı 15](#_Toc136902064)

[2.2.7 Yönetici Talep Atama Ekranı 16](#_Toc136902065)

**3.** [**Metod ve Yöntem** 17](#_Toc136902067)

[3.1 Web Api Alt Yapısı 17](#_Toc136902068)

[3.2 Windows Form Alt Yapısı 18](#_Toc136902069)

[3.3 Swagger 18](#_Toc136902070)

**4.** [**Yaygın Help Desk Sistemleri** 20](#_Toc136902072)

[4.1 Jira 20](#_Toc136902073)

[4.2 Zendesk 22](#_Toc136902074)

[4.3 Freshdesk 25](#_Toc136902075)

**5.** [**OsTicket Gerçek Hayat Örneği (Vesacons Başarı Hikayesi)** 28](#_Toc136902077)

**6.** [**Sonuç** 34](#_Toc136902079)

[**Kaynaklar** 36](#_Toc136902080)

# Şekiller Listesi

[Şekil 2.1: Giriş ekranı 11](#_Toc136901135)

[Şekil 2.2: Menü ekranı 12](#_Toc136901136)

[Şekil 2.3: Talep listesi ekranı 13](#_Toc136901137)

[Şekil 2.4: Yeni talep ekranı 13](#_Toc136901138)

[Şekil 2.5: Kayıt işlemi sonrası talep ekranı 14](#_Toc136901139)

[Şekil 2.6: Yönetici talep listesi 14](#_Toc136901140)

[Şekil 2.7: Yönetici talep kapatma ekranı 15](#_Toc136901141)

[Şekil 2.8: Yönetici talebi kapandıktan sonra ekran görünümü 15](#_Toc136901142)

[Şekil 2.9: Yönetici talep atama ekranı 16](#_Toc136901143)

[Şekil 2.10: Yönetici talep atama sonrasi ekran görünümü 16](#_Toc136901144)

[Şekil 3.1: Web api token alma 17](#_Toc136901145)

[Şekil 3.2: Token ile servis çağırımı 18](#_Toc136901146)

[Şekil 3.3: Swagger arayüzü 19](#_Toc136901147)

[Şekil 4.1: Back log ekranı 20](#_Toc136901148)

[Şekil 4.2: Scrum board ekranı 21](#_Toc136901149)

[Şekil 4.3: Jira Login ekranı 21](#_Toc136901150)

[Şekil 4.4: Zendesk ticket tarihçesi ekranı 24](#_Toc136901151)

[Şekil 4.5: Zendesk login ekranı 24](#_Toc136901152)

[Şekil 4.6: Freshdesk ana ekranı 27](#_Toc136901153)

[Şekil 4.7: Fresh desk login sayfası 27](#_Toc136901154)

[Şekil 5. 1: Ana ekran 30](#_Toc136901155)

[Şekil 5.2: Yeni talep açma ekranı 30](#_Toc136901156)

[Şekil 5.3: Temsilci giriş ekranı 31](#_Toc136901157)

[Şekil 5.4: Profil ekranı 31](#_Toc136901158)

[Şekil 5.5: Talep karşılama ekranı 32](#_Toc136901159)

[Şekil 5.6: Talep çözüm ekranı 32](#_Toc136901160)

[Şekil 5.7: Talep çözüm grafiği 33](#_Toc136901161)

[Şekil 5.8: Talep yardım konuları tanımlama ekranı 33](#_Toc136901162)

# Kısaltmalar Listesi

ASP.NET Active Server Pages

BT Bilgi Teknolojileri

c# Csharp

IT Information Technology

# Bölüm 1

# Talep Yönetim Sistemlerine Bakış

Son dönemlerde teknolojik gelişmelerin hız kesmeden ilerlemesiyle birlikte işletmelerin bilgi işlem süreçleri de önemli ölçüde dönüşüm geçirmiştir. Günümüzde şirketlerin bilgi işlem altyapıları, veri yönetimi, yazılım uygulamaları ve donanımları, iş süreçleri ve güvenlik önlemleri gibi unsurlar, işlerin etkin ve verimli bir şekilde yürütülmesi için kritik bir rol oynamaktadır.

Bu bağlamda, şirketlerin içerisinde yoğun bir şekilde kullanılan bilgi işlem hizmetlerinin etkin bir şekilde yönetilmesi büyük önem taşımaktadır. Bilgi işlem talepleri, çalışanların teknik destek talepleri, yazılım güncellemeleri, donanım talepleri gibi birçok farklı konuyu içerebilir. Bu taleplerin düzenli, hızlı ve etkin bir şekilde takip edilmesi, iş sürekliliği ve çalışan memnuniyeti açısından oldukça kritik bir faktördür.

Bu noktada, şirket içi bir Bilgi İşlem Talep Yönetim Sistemi'nin devreye alınması gerekmektedir. Bu sistem, işletmelerin bilgi işlem taleplerini izlemek, yönetmek ve çözmek için bir çerçeve sağlar. Çalışanlar, bu sistemi kullanarak taleplerini iletebilir, durumlarını takip edebilir ve sonuçları hakkında bilgilendirilebilirler. Aynı zamanda, yöneticiler ve destek ekipleri, sistem aracılığıyla talepleri etkin bir şekilde yönetebilir, önceliklendirebilir ve çözüm sürecini izleyebilirler.

Benim yaptığım katkı ile, şirketin mevcut bilgi işlem süreçlerini daha verimli hale getirecek, kullanıcıların taleplerini kolayca iletebilecekleri bir platform oluşturmayı hedefliyorum. Diğer sistemlerden en büyük farkı, Sistem çok katmanlı web api ve Windows form teknolojileri üzerinde çalışmasıdır.

IT Help desk sistemi aşağıdaki kavramlarla da ilişkilidir. Onları şu şekilde sıralayabiliriz.

* Change Management
* Request Management
* Project Management
* Problem Management
* Devops Sürekli Entegrasyon

## 1.1 Change Management (Değişiklik Yönetimi)

Değişiklik yönetimi, organizasyonun değişime uyum sağlamasını, riskleri azaltmasını ve istenilen sonuçları elde etmesini sağlar. İş sürekliliğini ve performansı etkilemeden, kontrol altında ve yapılandırılmış bir şekilde değişikliklerin gerçekleştirilmesini hedefler.

Değişiklik yönetimi süreci genellikle aşağıdaki adımlardan oluşur:

**İhtiyaç ve Planlama:** Değişikliğin gerekçesi, hedefleri ve kapsamı belirlenir. Değişiklik yönetimi ekibi, değişikliğin planını oluşturur, kaynakları tahsis eder ve sürecin aşamalarını belirler.

**Değerlendirme ve Analiz**: Değişikliğin etkileri, riskleri ve faydaları analiz edilir. İlgili paydaşlarla görüşmeler yapılır ve değişiklikten etkilenecek süreçler, sistemler ve kullanıcılar belirlenir. Uygunluk analizi yapılır ve gerektiğinde test veya pilot uygulamalar gerçekleştirilir.

**İletişim ve Paydaş Yönetimi**: Değişiklik, ilgili paydaşlara iletilir ve iletişim planı oluşturulur. Paydaşlar, değişikliğin nedenleri, süreci, etkileri ve beklenen sonuçları hakkında bilgilendirilir. İlgili tarafların katılımı ve desteği sağlanır.

**Uygulama:** Değişiklik planına göre uygulama gerçekleştirilir. Değişiklikler, test ortamlarında veya planlı kesintilerle üretim ortamına geçiş yaparak gerçekleştirilebilir. İlgili ekipler, değişiklikleri takip eder ve gerektiğinde sorunları çözer.

**İzleme ve Değerlendirme:** Değişikliklerin etkileri izlenir ve değerlendirilir. Performans ölçütleri, kullanıcı geri bildirimleri ve iş sürekliliği takibi gibi faktörler göz önünde bulundurulur. Değişikliklerin beklenen sonuçları sağlayıp sağlamadığı değerlendirilir.

**İyileştirme ve Öğrenme:** Değişiklik yönetimi süreci, yapılan değişikliklerden dersler çıkarılmasını ve sürekli iyileştirmeyi teşvik eder. Değişikliklerle ilgili deneyimler ve geri bildirimler, gelecekteki projeler için bir kaynak haline gelir.

Değişiklik yönetimi, aşağıdaki yararları sağlar:

1. Kontrol ve İş Sürekliliği: Değişikliklerin planlı ve yapılandırılmış bir şekilde gerçekleştirilmesi, iş sürekliliğini sağlar. Kontrol altında gerçekleştirilen değişiklikler, olumsuz etkileri en aza indirir ve beklenen sonuçların elde edilmesini sağlar.

2. Risk Yönetimi: Değişikliklerin etkileri ve riskleri önceden değerlendirilir. Uygunluk analizi ve testler yapılır. Bu sayede, potansiyel sorunları ve hataları önceden tespit etmek ve riskleri minimize etmek mümkün olur.

3. İletişim ve Paydaş Yönetimi: Değişiklik yönetimi süreci, ilgili paydaşları bilgilendirir ve onları değişiklik sürecine dahil eder. İyi bir iletişim ve paydaş yönetimi, değişikliklerin kabulünü ve desteklenmesini artırır.

4. İyileştirme ve Öğrenme: Değişiklik yönetimi süreci, yapılan değişikliklerden dersler çıkarılmasını ve sürekli iyileştirmeyi teşvik eder. Değişikliklerle ilgili deneyimler ve geri bildirimler, gelecekteki projeler için bir kaynak haline gelir.

## 1.2 Request Management (İstek Yönetimi)

İstek yönetimi (request management), bir organizasyonun veya şirketin kullanıcıların taleplerini kaydetmek, takip etmek, yönlendirmek ve çözmek için kullanılan bir yönetim sürecidir. İstek yönetimi, kullanıcıların ihtiyaçlarını ve taleplerini etkin bir şekilde ele alarak müşteri memnuniyetini artırmayı hedefler.

**1. Talep Kaydı:** Kullanıcıların talepleri, bir istek yönetim sistemi veya yazılım aracılığıyla kaydedilir. Kullanıcılar, sorunlarını bildirebilir, yardım talep edebilir, değişiklik isteyebilir veya diğer isteklerde bulunabilir. Talepler, bir takip numarası veya benzersiz bir kimlikle birlikte sisteme kaydedilir.

**2. Talep Değerlendirmesi:** İstek yönetimi ekibi, kaydedilen talepleri inceler ve önceliklerini belirler. Taleplerin doğruluğu, gereklilikleri ve aciliyeti değerlendirilir. İstekler, uygun ekiplere veya departmanlara yönlendirilir.

**3. İstek Yönlendirmesi:** İstekler, doğru ekiplere veya yetkililere yönlendirilir. Bu, teknik destek ekibi, satın alma departmanı, finans ekibi veya diğer ilgili birimleri içerebilir. İsteklerin doğru kişilere yönlendirilmesi, hızlı ve etkili çözümlerin sağlanmasını sağlar.

**4. Takip ve İletişim:** İstekler, sürecin her aşamasında takip edilir ve kullanıcılarla iletişim sağlanır. Kullanıcılara, isteklerinin durumu hakkında bilgi verilir ve güncellemeler iletilir. İsteklerin zamanında ve doğru şekilde çözülmesi için iletişim ve takip önemlidir.

**5. Çözüm ve Kapatma**: İstekler, ilgili ekipler tarafından çözülür veya gereken aksiyonlar alınır. Kullanıcıya uygun bir çözüm sunulduktan sonra, istek kapatılır. Kullanıcının talebinin tamamlandığı ve memnuniyetinin sağlandığından emin olunur.

İstek yönetimi süreci, aşağıdaki yararları sağlar:

**Kullanıcı Memnuniyeti:** İstek yönetimi, kullanıcıların taleplerini etkin bir şekilde yöneterek memnuniyetlerini artırır. Hızlı yanıtlar, takip ve iletişim sağlama, isteklerin doğru şekilde yönlendirilmesi gibi faktörler, kullanıcıların beklentilerini karşılamalarını sağlar.

**İş Sürekliliği:** İstek yönetimi, kullanıcıların sorunlarını veya taleplerini hızlı bir şekilde çözmek için bir yapı sağlar. Bu, iş süreçlerinin kesintiye uğramasını önler ve iş sürekliliğini sağlar.

**Raporlama ve Analiz:** İstek yönetimi süreci, kaydedilen taleplerin verilerini toplar ve raporlar oluşturur. Bu raporlar, işletmenin performansını izlemek, trendleri belirlemek ve iyileştirme fırsatlarını tanımlamak için kullanılabilir.

**İzlenebilirlik ve Sorumluluk:** İstek yönetimi süreci, taleplerin takip numaraları veya benzersiz kimliklerle kaydedilmesini sağlar. Bu, isteklerin takibini kolaylaştırır ve sorumluluğu belirli ekiplere veya kişilere atama imkanı verir.

**Sürekli İyileştirme:** İstek yönetimi süreci, kullanıcı geri bildirimlerini ve deneyimleri dikkate alarak sürekli olarak iyileştirilir. Geri bildirimler, süreçteki zayıf noktaların belirlenmesine ve daha iyi hizmet sunma amacıyla düzeltici önlemler alınmasına yardımcı olur.

Sonuç olarak, istek yönetimi süreci, kullanıcıların taleplerini etkin bir şekilde yönetmek, çözmek ve takip etmek için kullanılan bir yönetim sürecidir. Bu süreç, kullanıcı memnuniyetini artırır, iş sürekliliğini sağlar ve verimliliği artırır. Ayrıca, raporlama ve analiz araçlarıyla sürekli iyileştirme için bir temel oluşturur.

Kullanıcıların taleplerini etkin bir şekilde yöneterek, işletmelerin müşteri memnuniyetini artırmasına ve operasyonel verimliliklerini iyileştirmesine olanak sağlar. Doğru yönlendirme, takip ve iletişim sağlama, hızlı çözüm sunma gibi faktörler, kullanıcıların taleplerinin zamanında ve etkili bir şekilde karşılanmasını sağlar. Bu da işletmelerin kullanıcı memnuniyetini sağlamak, iş sürekliliğini korumak ve sürekli olarak iyileşmek için önemli bir süreçtir.

## 1.3 Project Management (Proje Yönetimi)

Proje Yönetimi, bir organizasyonun hedeflerine ulaşmak için kaynakları ve süreçleri planlama, yönetme ve uygulama disiplinidir. Projeler genellikle ekipler arası işbirliği gerektirir ve süreçler boyunca çeşitli sorunlar ortaya çıkabilir. Bu noktada, bir Help Desk Sistemi devreye girer. Help Desk Sistemi, kullanıcıların sorunlarını bildirebilecekleri ve çözüm taleplerini iletebilecekleri bir platform sağlar. Proje Yönetimi ile Help Desk Sistemi arasındaki ilişkiyi ve bu ilişkinin proje başarısına olan katkısını inceleyeceğiz.

**İletişim ve Koordinasyon:** Proje Yönetimi, ekip üyeleri arasında etkili iletişimi ve koordinasyonu sağlamak için bir dizi yöntem ve araç kullanır. Help Desk Sistemi, proje ekibinin üyeleri arasında sorunları ve çözüm taleplerini iletmek için bir iletişim kanalı sağlar. Ekip üyeleri, sistem aracılığıyla sorunları bildirebilir, işbirliği yapabilir ve güncellemeleri takip edebilir. Bu, ekip üyeleri arasında etkili bir iletişim sağlar ve proje sürecindeki gecikmeleri en aza indirir.

**Sorun İzleme ve Yönetimi:**  Proje sürecinde ortaya çıkan sorunların zamanında ve etkili bir şekilde yönetilmesi, başarı için kritik öneme sahiptir. Help Desk Sistemi, sorunların izlenmesini ve yönetilmesini kolaylaştırır. Ekip üyeleri, sorunları sisteme bildirebilir, öncelik düzeylerini belirleyebilir ve sorunları takip edebilir. Proje yöneticileri, bu sistemi kullanarak sorunları çözme sürecini izleyebilir ve gerektiğinde müdahale edebilir. Bu, proje sürecinde sorunların hızla çözülmesini ve projenin ilerlemesini sağlar.

**Verimlilik ve Performans Analizi:** Proje Yönetimi, proje verimliliğini ve performansını analiz etmek için çeşitli metrikler kullanır. Help Desk Sistemi, proje sürecindeki sorunları ve çözüm süreçlerini izleyerek veriler sağlar. Bu veriler, ekip üyelerinin performansını değerlendirmek ve sürekli iyileştirmeler yapmak için kullanılabilir. Proje yöneticileri, sistem

**Sonuç:** Proje Yönetimi ve Help Desk Sistemi, proje sürecinin başarılı bir şekilde yönetilmesi için birbirini tamamlayan unsurlardır. Help Desk Sistemi, proje ekiplerine etkili iletişim, sorun izleme ve yönetimi, verimlilik analizi, belge paylaşımı ve kullanıcı memnuniyeti değerlendirmesi gibi faydalar sağlar. Bu sistem, proje yöneticilerine gerçek zamanlı bilgiler sunar ve proje sürecindeki sorunları hızlı bir şekilde çözme ve sürekli iyileştirme fırsatı verir. Proje Yönetimi ve Help Desk Sistemi arasındaki entegrasyon, proje başarısını artırmak ve proje ekiplerinin verimliliğini artırmak için önemli bir faktördür.

## 1.4 Problem Management (Problem Yönetimi)

Bir organizasyon içinde doğabilecek sorunlar, iş sürekliliği ve verimlilik açısından önemli bir etkiye sahip olabilir. Sorunların etkin bir şekilde yönetilmesi, işletmenin sürdürülebilirliği için kritik önem taşır. Bu noktada, "Problem Yönetimi" ve "Help Desk Yönetimi" ön plana çıkar. Problem Yönetimi, sorunları tanımlama, analiz etme, çözme ve önlem alma sürecidir. Help Desk Yönetimi ise kullanıcıların sorunlarını bildirdiği, takip edildiği ve çözümlendiği bir destek sistemidir. Bu yazıda, Problem Yönetimi ve Help Desk Yönetimi arasındaki ilişkiyi ve organizasyonlara sağladıkları faydaları inceleyeceğiz.

**Sorunların Bildirimi ve Kayıt Altına Alınması:** Help Desk Yönetimi, kullanıcıların sorunları bildirebileceği ve kayıt altına alabileceği bir destek sistemidir. Kullanıcılar, sorunlarını Help Desk aracılığıyla iletebilir ve bu sorunlar bir veritabanına kaydedilir. Problem Yönetimi süreci için bu kayıtlar, sorunların türünü, önceliğini ve tekrarlama sıklığını belirlemek açısından önemlidir. Sorunlar düzenli olarak kaydedilerek analiz edilebilir ve gelecekte benzer sorunların önlenmesi için önleyici önlemler alınabilir.

**Sorunların İzlenmesi ve Önceliklendirilmesi:** Help Desk Yönetimi, sorunların izlenmesi ve önceliklendirilmesi sürecini kolaylaştırır. Help Desk sistemi, bildirilen sorunları bir takip mekanizmasıyla yönetir ve öncelik düzeylerini belirleyebilir. Problem Yönetimi süreci için bu bilgiler, sorunların çözüm sürecini planlamak ve kaynakları etkin bir şekilde tahsis etmek açısından önemlidir. Öncelikli sorunlar, hızlı bir şekilde çözülmeli ve iş sürekliliğine en az etkiyle çözümlenmelidir.

**Sorunların Çözümü ve Dokümantasyonu:** Problem Yönetimi süreci, sorunların etkin bir şekilde çözülmesini sağlar. Help Desk Yönetimi, kullanıcıların sorunlarını takip eder, sorunların çözüm sürecini yönetir ve çözümleri sağlar. Çözümler, Help Desk sisteminde kaydedilir ve sorunun tekrarlanması durumunda referans olması amacıyla dokümantasyon yapılır. Problem Yönetimi süreci, çözüm süreçlerini analiz eder, etkili çözümleri belirler ve gelecekte benzer sorunların hızlı bir şekilde çözülmesini sağlar. Aynı zamanda, çözüm süreçlerinin belgelendirilmesi, ekip üyeleri arasında bilgi paylaşımını artırır ve gelecekteki sorunlara daha hızlı bir şekilde yanıt verilmesine yardımcı olur.

**Sürekli İyileştirme ve Önleyici Önlemler:** Problem Yönetimi süreci, sorunların tekrarlanmasını önlemek için sürekli iyileştirme ve önleyici önlemler almaya odaklanır. Help Desk Yönetimi, sorunların kaydedilmesi ve analiz edilmesiyle elde edilen verileri kullanarak, sorunların nedenlerini belirler ve gelecekte benzer sorunların önlenmesi için adımlar atar. Bu, organizasyonun sürekli olarak iyileştirme yapmasını ve sorunların tekrarlanma olasılığını azaltmasını sağlar.

**Sonuç:** Problem Yönetimi ve Help Desk Yönetimi, organizasyonların sorunları etkin bir şekilde çözmelerine ve iş sürekliliğini sağlamalarına yardımcı olan önemli süreçlerdir. Help Desk Yönetimi, kullanıcıların sorunlarını bildirebilecekleri ve çözüm süreçlerini takip edebilecekleri bir destek sistemidir. Problem Yönetimi süreci ise, sorunları analiz eder, çözümler üretir, sorunların tekrarlanmasını önlemek için önleyici önlemler alır ve sürekli iyileştirme sağlar. Bu iki süreç birlikte çalışarak, organizasyonların sorunları etkin bir şekilde yönetmelerini ve iş sürekliliğini sürdürmelerini sağlar.

## 1.5 Devops (Sürekli Entegrasyon)

Günümüz iş dünyasında, sürekli değişen ve hızla büyüyen teknolojik ortamlarda başarılı olmak için işletmelerin iyi bir IT altyapısına ve etkin destek hizmetlerine ihtiyaç duyduğu bir gerçektir. Bu ihtiyaçlar, DevOps ve Help Desk gibi iki önemli kavramın ortaya çıkmasına yol açmıştır. DevOps, yazılım geliştirme ve operasyon ekipleri arasındaki işbirliği ve bütünleşmeyi vurgularken, Help Desk ise müşteri hizmetleri ve destek süreçlerini yönetmeyi hedefler. Bu iki alan arasında sağlıklı bir ilişki, işletmelere daha iyi sonuçlar ve müşteri memnuniyeti sağlayabilir.

DevOps, yazılım geliştirme ve operasyon ekiplerinin birlikte çalışmasını teşvik eden bir kültür ve işbirliği metodolojisidir. Yazılım projelerinin hızlı bir şekilde geliştirilip dağıtılması ve işletim ortamlarının sürekli iyileştirilmesi hedeflenir. DevOps, yazılımın yaşam döngüsünün her aşamasında otomasyon, sürekli entegrasyon ve sürekli dağıtım gibi uygulamaları kullanarak verimlilik ve kaliteyi artırır. Böylece, işletmeler hızlı ve güvenilir bir şekilde yenilikçi çözümler sunabilir.

Help Desk ise müşteri hizmetleri ve destek süreçlerini yönetmek için kullanılan bir sistem veya ekibin adıdır. Müşterilerin sorunlarını çözmek, taleplerini yanıtlamak ve hizmet düzeyi anlaşmalarına (SLA'lar) uygun olarak destek sağlamak için kullanılır. Help Desk, bilet sistemi, iletişim kanalları ve bilgi tabanı gibi özellikler sunarak müşterilere etkili bir hizmet sunmayı hedefler. İyi bir Help Desk ekibi, müşterilerin sorunlarını hızlı ve profesyonel bir şekilde çözerken, müşteri memnuniyetini de artırır.

DevOps ve Help Desk arasındaki ilişki, yazılımın yaşam döngüsünün her aşamasında destek sağlamak ve müşteri hizmetlerini optimize etmek için işbirliği ve etkileşimi vurgular. İşletmeler, DevOps prensiplerini uygulayarak hızlı ve sürekli bir şekilde yazılım geliştirebilirken, Help Desk ekibi de bu süreçte müşteri taleplerini yönetir ve çözümler sunar.

Aşağıda sürekli entegrasyon araçlarından bahsetmek istiyorum. Sürekli entegrasyon araçları help desk sistemleri %100 entegre çalışır.

**Jenkins:** Jenkins, en popüler açık kaynak sürekli entegrasyon araçlarından biridir. Çeşitli projelerde kullanılabilir, geniş bir eklenti ve entegrasyon desteği sunar ve esnek yapılandırma imkanı sağlar.

**Travis CI:** Travis CI, GitHub projeleriyle entegrasyonu kolaylaştıran bir bulut tabanlı sürekli entegrasyon aracıdır. Hem açık kaynak hem de özel projelerde kullanılabilir.

**CircleCI:** CircleCI, GitHub, Bitbucket ve GitLab gibi popüler kod depolarıyla entegrasyonu kolaylaştıran bulut tabanlı bir sürekli entegrasyon platformudur. Hem Linux hem de macOS ortamlarında çalışabilir.

**GitLab CI/CD:** GitLab, Git tabanlı bir kod depolama platformudur ve kendi sürekli entegrasyon ve dağıtım (CI/CD) araçlarını da içerir. GitLab CI/CD, projelerinizi kolayca otomatikleştirmenize, test etmenize ve dağıtmanıza olanak tanır.

**Bamboo:** Atlassian tarafından sunulan Bamboo, Jira, Bitbucket ve diğer Atlassian ürünleriyle entegrasyonu sağlayan bir sürekli entegrasyon ve dağıtım aracıdır. Hem Java hem de .NET projeleri için destek sunar.

**TeamCity:** JetBrains tarafından geliştirilen TeamCity, .NET ve Java projeleri için kullanılabilen bir sürekli entegrasyon aracıdır. Ölçeklenebilir ve esnek bir yapıya sahiptir.

# Bölüm 2

# Geliştirilen Sistem

## 2.1 Gereklilik Sebepleri

Bilgi işlem talep yönetim sisteminin neden gerekli olduğunu aşağıdaki maddelerde açıklayabiliriz.

### 2.2.1 İş Sürekliliği

Şirketlerin bilgi işlem hizmetlerinin kesintisiz ve etkin bir şekilde çalışması, iş sürekliliği açısından büyük önem taşır. Bilgi işlem taleplerinin düzensiz veya kontrolsüz bir şekilde yönetilmesi, sistemlerin çökmesine veya çalışanların işler ini verimli bir şekilde sürdürememesine neden olabilir. Bu nedenle, bir Bilgi İşlem Talep Yönetim Sistemi, taleplerin sistematik bir şekilde kaydedilmesini, takip edilmesini ve çözüme kavuşturulmasını sağlayarak iş sürekliliğini artırır.

### 2.2.2 İş Verimliliği

Bilgi işlem taleplerinin etkin bir şekilde yönetilmesi, şirketin genel iş verimliliğini artırır. Taleplerin hızlı bir şekilde kaydedilmesi ve ilgili ekip veya departmana atanması, zaman kaybını önler ve taleplerin zamanında çözüme kavuşmasını sağlar. Böylece, çalışanlar daha verimli bir şekilde çalışabilir ve iş süreçleri aksamadan ilerler.

### 2.2.3 İşbirliği ve İletişim

Bir Bilgi İşlem Talep Yönetim Sistemi, farklı ekipler veya departmanlar arasındaki işbirliğini ve iletişimi kolaylaştırır. Talepler, sisteme kaydedildiğinde ilgili ekiplere veya departmanlara otomatik olarak iletilir ve takip edilir. Bu sayede, sorunların daha hızlı ve etkin bir şekilde çözülmesi sağlanır ve çalışanlar arasında iş birliği ve iletişim süreci kolaylaşır.

### 2.2.4 Veri Analizi ve İyileştirme

Bir Bilgi İşlem Talep Yönetim Sistemi, taleplerin verilerini analiz edebilme ve raporlama özelliğine sahip olabilir. Bu veriler, yöneticilere önemli bilgiler sağlar ve iş süreçlerinin iyileştirilmesi için bir rehber niteliği taşır. Örneğin, taleplerin sıkça tekrarlanan tipleri veya en çok zaman alan süreçler belirlenebilir ve bu bilgilere dayanarak iyileştirme stratejileri geliştirilebilir.

## 2.2 Arayüzler

### 2.2.1 Giriş Ekranı

Aşağıda sistem için tasarlanan login ekranı bulunmaktadır. Sistem Windows form (C#.net) tabanlı mimari ile yazılmıştır. Dağıtık (SOA) sistem mimarisi üzerinde çalışmaktadır. Kullanıcını doğru kullanıcı adı ve şifre ile giriş yaptıktan sonra. Rolune göre (Yönetici veya kullanıcı) yönlendirme yapılacaktır. Aşağıda örnek bir kullanıcı ile giriş yapılmıştır.

A screenshot of a login form

Description automatically generated with low confidence

Şekil 2.1: Giriş ekranı

### 2.2.2 Menü Ekranı

Aşağıda standart kullanıcı yetkisi ile giren kullanıcıyı karşılayan menü ekranı bulunmaktadır. Kullanıcı şirket içerisinde bilgi işlem departmanı dışında (Örnek: Finans) çalışan bir kullanıcıdır. Talep yapmak amacıyla sisteme girmiştir. Menu ekranında login olan kullanıcılar hakkında bilgiler yer almaktadır.

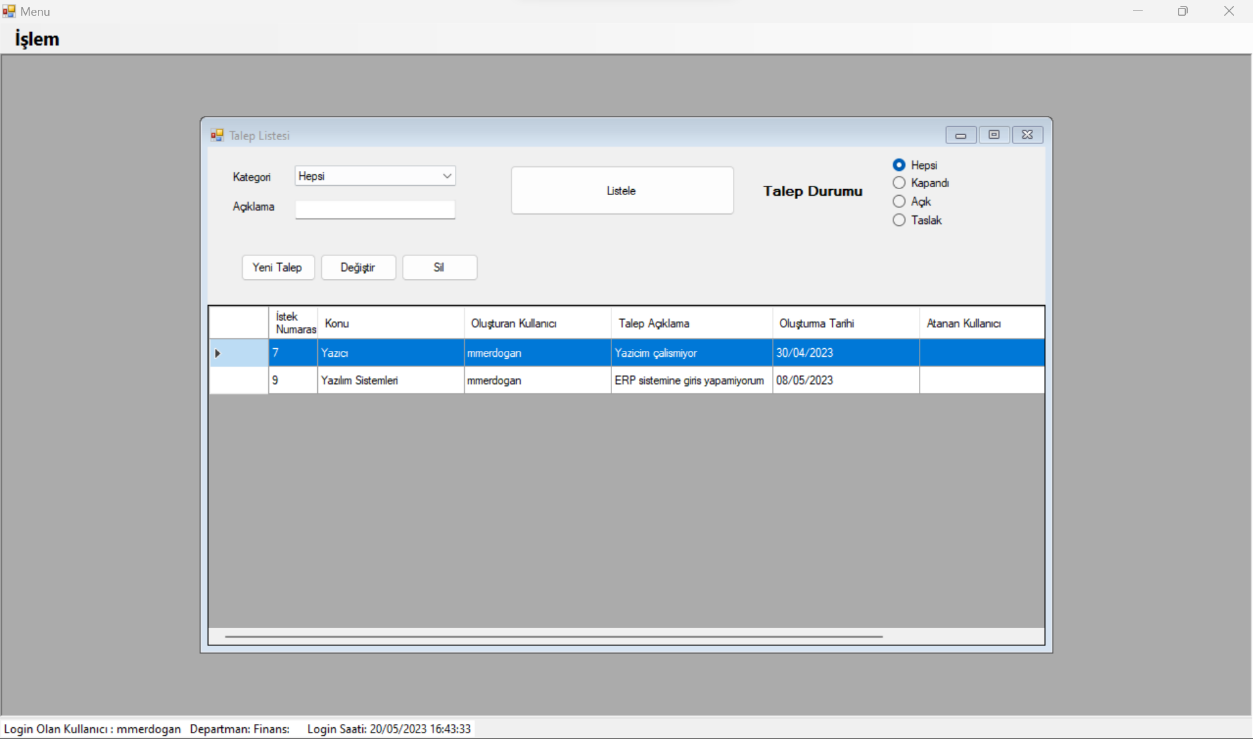
A screenshot of a computer

Description automatically generated with medium confidence

Şekil 2.2: Menü ekranı

### 2.2.3 Talep Listesi Ekranı

Aşağıdaki ekranda daha önce oluşturulan kullanıcı talepleri listelenmektedir. Oluşturan talepleri detaylı olarak filtrelemek için bu ekran kullanılmaktadır. “Kategori”, “Talep Açıklama” ve “Talep Durumu” alanına göre filtreleme yapılabilmektedir. Listele butonundan ilgili taleplere göre filtreleme yapılmaktadır. Listeleme ekranı üzerindeki “Yeni Talep” , “Değiştir” ve “Sil” butonları ile ilgili işlemler yapılmaktadır.



Şekil 2.3: Talep listesi ekranı

### 2.2.4 Yeni Talep Ekranı

Aşağıdaki ekranda kullanıcının yeni help desk talebi oluşturacağı ekran gösterilmiştir. Bu ekranda kullanıcı belirli bir kategori ve önem derecesi seçerek yeni talebini oluşturabilir. Yeni talebini seçerken talep detayı bölümünden talep ile ilgili açıklamasını girebilir. Talep oluştururken direk kayıt yapabilir ya da taslak olarak kaydedebilir ve bu taslak kayıt üzerinden daha sonra devam edebilir.

A screenshot of a computer

Description automatically generated

Şekil 2.4: Yeni talep ekranı

Normal kayıt işlemi olduktan sonra bir değişlik yapılmamaktadır.

A screenshot of a computer

Description automatically generated

Şekil 2.5: Kayıt işlemi sonrası talep ekranı

### 2.2.5 Yönetici Talep Listesi

Yönetici gelen talepleri aşağıdaki gibi karşılayabilmektedir. Gelen talepleri isterse başka kullanıcıya direk atayabilir ve talep kapatma işlemi yapabilir.

A screenshot of a computer

Description automatically generated with medium confidence

Şekil 2.6: Yönetici talep listesi

### 2.2.6 Yönetici Talep Kapatma Ekranı

Aşağıdaki ekran üzerinden çözüme kavuşturulan talepler kapatılabilir.

A screenshot of a computer

Description automatically generated

Şekil 2.7: Yönetici talep kapatma ekranı

Talep kapandıktan sonra aşağıdaki ekranda talep aşağıdaki gibi gözükecektir.

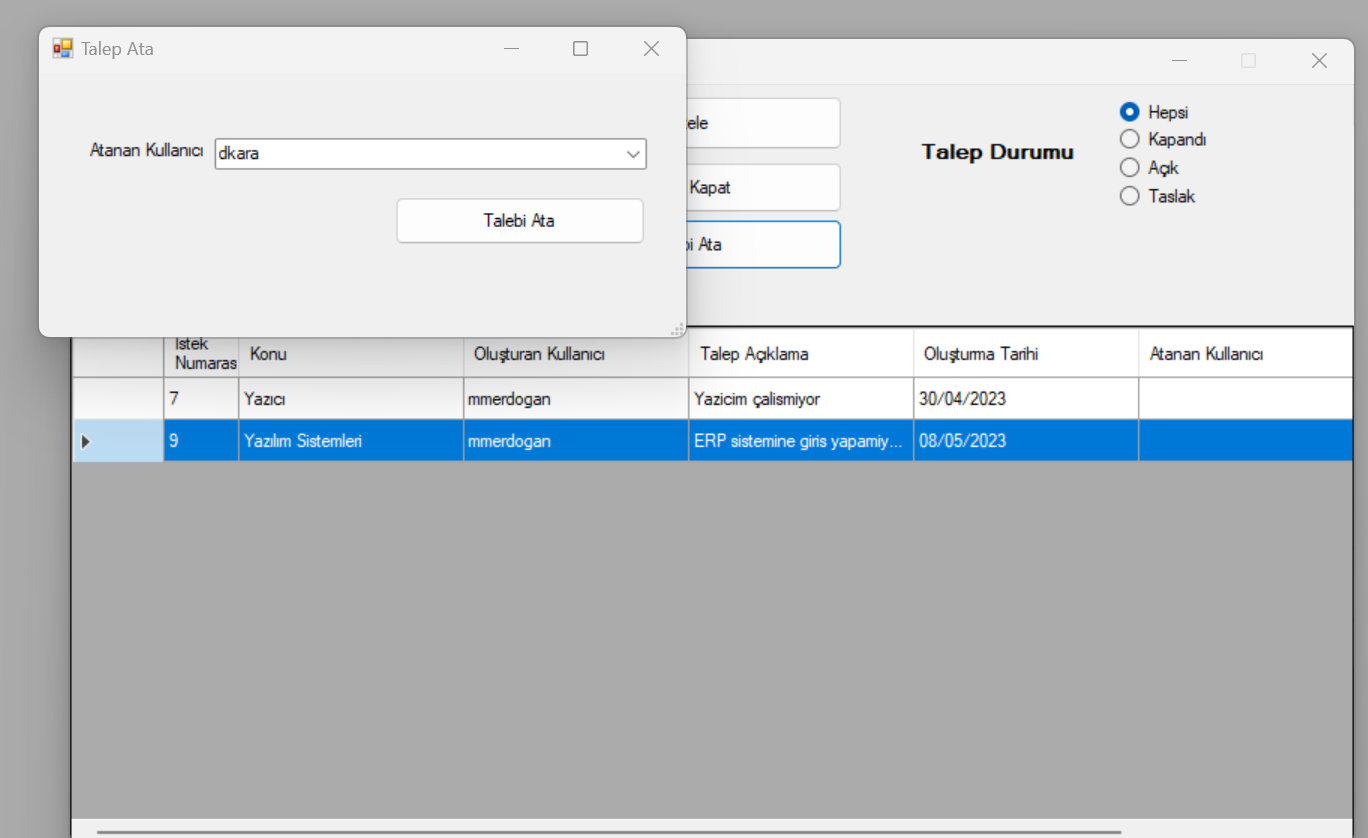
A screenshot of a computer

Description automatically generated with medium confidence

Şekil 2.8: Yönetici talebi kapandıktan sonra ekran görünümü

### 2.2.7 Yönetici Talep Atama Ekranı

Aşağıdaki ekran üzerinde ise dkara kullanıcısına talep atama gerçekleştirilmiştir.



Şekil 2.9: Yönetici talep atama ekranı

Talep atama gerçekleştikten sonra atanan kullanıcı bölümünde atanan kullanıcı dkara olarak gözükmektedir.

*A screenshot of a computer

Description automatically generated*

Şekil 2.10: Yönetici talep atama sonrasi ekran görünümü

# Bölüm 3

# Metod ve Yöntem

## 3.1 Web Api Alt Yapısı

Bu projede Windows Form uygulaması ile birlikte ASP.NET Web API kullanarak token tabanlı kimlik doğrulama yöntemini kullanarak güvenli bir uygulama geliştirdim. Token tabanlı kimlik doğrulama, kullanıcıların kaynaklara erişmek için bir token almasını ve bu token'ı kullanarak isteklerini doğrulamasını sağlar. Bu şekilde, uygulamada yetkilendirme ve güvenlik önlemleri uygulayarak kullanıcı bilgilerinin ve kaynakların güvenliği sağlanmaktadır. Aşağıda login ekranından sonra başarılı giriş yaptıktan sonra global bir değişkene token atanıyor ve diğer isteklerde bu token kullanılarak erişim sağlanıyor.

A screenshot of a computer program

Description automatically generated with medium confidence

Şekil 3.1: Web api token alma

Token ile bir servis çağırım örneği aşağıda verilmiştir. Token data değişkeni servisin header kısmına eklendikten sonra çağırım yapılmaktadır.

A screen shot of a computer program

Description automatically generated with low confidence

Şekil 3.2: Token ile servis çağırımı

## 3.2 Windows Form Alt Yapısı

Windows Form, Microsoft tarafından geliştirilen ve kullanıcı arayüzü oluşturmak için kullanılan bir teknolojidir. Windows Form uygulamaları, Windows işletim sistemlerinde çalışan masaüstü uygulamalarıdır. Bu teknoloji, kullanıcı dostu arayüzler oluşturmanıza ve Windows ortamında zengin işlevselliğe sahip uygulamalar geliştirmenize olanak sağlar.

Windows Form uygulamaları, Visual Studio gibi entegre geliştirme ortamlarında oluşturulur ve C# veya Visual Basic gibi programlama dilleriyle geliştirilebilir. Projemde Windows form ve c# kullanarak yazılım geliştirdim.

## 3.3 Swagger

Bu projede swagger teknolojisini kullandım .Swagger, API belgeleme ve test etme için popüler bir açık kaynaklı bir araç ve çerçevedir. Swagger, RESTful web servislerini tanımlayan, belgeleyen ve görsel olarak etkileşimli bir arayüz sağlayan bir JSON tabanlı açık standart olan OpenAPI Specification (OAS) üzerine inşa edilmiştir. Swagger, API'nin sunduğu kaynakları, parametreleri, yanıtları ve diğer önemli bilgileri açık ve anlaşılır bir şekilde belgelemek için kullanılır. Aşağıda projemden swagger arayüzü bulunmaktadır. Projemdeki methodlar bu arayüz ile kolay şekilde test edi

A screenshot of a computer

Description automatically generated with medium confidence

Şekil 3.3: Swagger arayüzü

# Bölüm 4

# Yaygın Help Desk Sistemleri

## 4.1 Jira

Jira, Atlassian tarafından geliştirilen ve en popüler proje yönetimi ve hata takip sistemlerinden biridir. İlk olarak 2002 yılında piyasaya sürülen Jira, o zamandan beri büyük bir kullanıcı tabanına sahip olmuştur. İşletmelerin, yazılım geliştirme, projelerin yönetimi ve destek süreçlerinin takibi gibi çeşitli alanlarda kullanabileceği kapsamlı bir platform sunar.

Aşağıda jiradan’ örnek bir ekranlar verilmiştir.

A screenshot of a computer

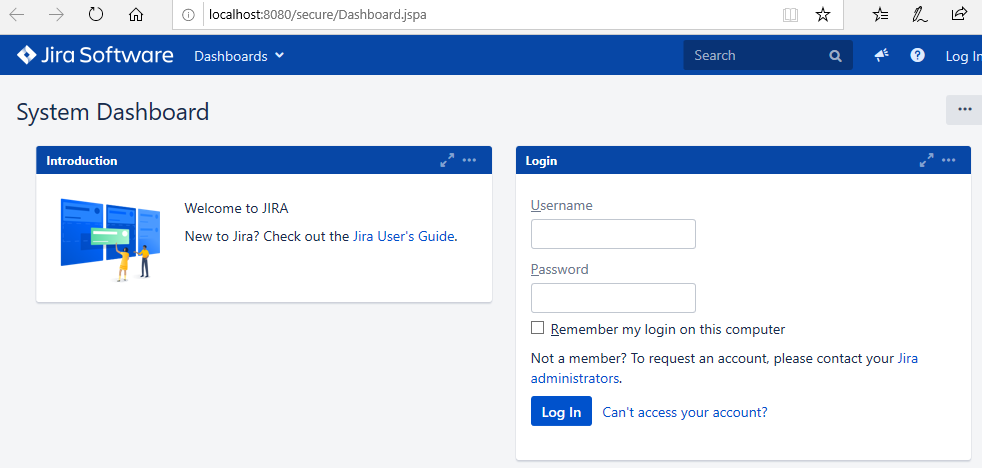
Description automatically generated with medium confidence

Şekil 4.1: Back log ekranı

A screenshot of a computer

Description automatically generated with medium confidence

Şekil 4.2: Scrum board ekranı



Şekil 4.3: Jira Login ekranı

Jira'nın özellikleri ve faydaları şunlardır.

1. Proje yönetimi: Jira, kullanıcılara proje oluşturma, görev atama, zamanlama ve proje ilerlemesini takip etme gibi projelerin yönetimini kolaylaştıran bir dizi araç sunar. Ayrıca, Gantt grafikleri, iş akışları ve raporlama gibi özelliklerle proje takibini optimize eder.

2. Hata takibi: Jira, yazılım geliştirme sürecinde hata takibi için etkili bir çözümdür. Hataları oluşturabilir, atayabilir, önceliklendirebilir ve takip edebilirsiniz. Ayrıca, hatalarla ilgili dosya ekleri, yorumlar ve geçmiş günlükleri gibi bilgileri depolayabilirsiniz.

3. Esnek iş akışı: Jira'nın özelleştirilebilir iş akışı özelliği, farklı projeler ve ekipler için uygun süreçler oluşturmanızı sağlar. Bu, iş akışlarınızı kurumsal gereksinimlerinize ve çalışma şeklinize göre uyarlamak için gereken esnekliği sunar.

4. Entegrasyonlar ve eklentiler: Jira, birçok popüler araç ve hizmetle entegrasyon sağlar. Slack, Confluence, Bitbucket gibi diğer Atlassian ürünleriyle entegre çalışabilir ve çeşitli eklentilerle işlevselliği genişletebilirsiniz. Bu, Jira'yı işletme ekosisteminizdeki diğer araçlarla kolayca entegre edilebilen bir çözüm haline getirir.

5. Raporlama ve analitik: Jira, kullanıcıların proje performansını ve iş takibini değerlendirmek için kapsamlı raporlama ve analitik özellikler sunar. Görsel raporlar, özelleştirilebilir panolar ve ilerleme izleme gibi özelliklerle projelerinizi verimli bir şekilde analiz edebilirsiniz.

## 4.2 Zendesk

Zendesk, 2007 yılında Mikkel Svane, Morten Primdahl ve Alexander Aghassipour tarafından kurulan bir yazılım şirketidir. Şirketin merkezi San Francisco, California'dadır. Zendesk'in temel amacı, şirketlere müşteri hizmetleri süreçlerini yönetmek için kullanabilecekleri kullanıcı dostu ve etkili bir platform sağlamaktır.

Kuruluşundan itibaren Zendesk, müşteri destek yazılımı alanında çığır açıcı bir hizmet sunmaktadır. İlk olarak 2007 yılında e-posta tabanlı müşteri destek sistemini sunan Zendesk, hızla büyüyerek farklı kanallardan gelen müşteri taleplerini yönetme, izleme ve yanıtlama konusunda yeteneklerini genişletmiştir.

2010 yılında Zendesk'in hizmetleri, sosyal medya entegrasyonu özelliğiyle güçlendirildi. Bu sayede şirketler, müşteri taleplerini Facebook, Twitter ve diğer sosyal medya platformları üzerinden takip edebilir ve yanıtlayabilir hale geldi.

Zendesk, kullanıcı deneyimini iyileştirmek ve müşteri hizmetlerini daha etkili bir şekilde yönetmek için sürekli olarak yenilikler ve geliştirmeler yapmaktadır. 2012 yılında canlı sohbet desteği eklenerek, şirketler müşteri taleplerini gerçek zamanlı olarak yanıtlayabilir hale geldi. Ayrıca, mobil uygulama desteği sunarak müşteri hizmetlerine erişimi kolaylaştırdı.

Zendesk, 2014 yılında halka arz edilerek hızla büyümeye devam etti. Şirket, kullanıcı tabanını genişletti ve dünya genelinde birçok büyük şirketin müşteri destek süreçlerini yönetmek için tercih ettiği bir platform haline geldi.

Günümüzde Zendesk, müşteri destek süreçlerinin yanı sıra satış, pazarlama ve iş birliği gibi alanlarda da kullanılan kapsamlı bir yazılım platformudur. Şirketler, Zendesk'i müşteri taleplerini etkili bir şekilde yönetmek, müşteri memnuniyetini artırmak ve iş süreçlerini optimize etmek için tercih etmektedir.

Zendesk'in temel özelliklerinden bazıları aşağıdaki gibidir.

**Müşteri İletişim Kanalları:** Zendesk, e-posta, canlı sohbet, telefon, sosyal medya ve web formları gibi farklı iletişim kanallarından gelen müşteri taleplerini yönetmek için entegre çözümler sunar. Bu sayede şirketler, müşteri taleplerini tek bir merkezi platformda toplayarak daha hızlı ve etkili bir şekilde yanıtlayabilir.

**Bilet Yönetimi:** Zendesk, gelen müşteri taleplerini otomatik olarak bilet formunda kaydeder. Bu biletler, talebin önceliğine göre kategorize edilebilir, atanan kişilere yönlendirilebilir ve takip edilebilir. Böylece şirketler, müşteri taleplerini daha iyi organize edebilir ve zamanında çözüm sağlayabilir.

**Self-Servis Müşteri Portalı:** Zendesk, müşterilerin kendi kendine sorunlarını çözebilecekleri bir self-servis portalı sunar. Bu portal, müşterilere sıkça sorulan soruları (SSS), bilgi tabanını ve kullanıcı kılavuzlarını sağlayarak müşteri memnuniyetini artırır ve destek ekibinin yükünü azaltır.

**İş Akışı Otomasyonu:** Zendesk, iş süreçlerini otomatikleştirmek için özelleştirilebilir iş akışları sunar. Bu sayede şirketler, belirli eylemlere otomatik yanıtlar gönderebilir, görevleri otomatik olarak atayabilir ve belirli durumlarda bildirimler alabilir. İş akışı otomasyonu, müşteri hizmetlerinin daha verimli ve tutarlı olmasını sağlar.

**Raporlama ve Analitik:** Zendesk, müşteri hizmetleri performansını izlemek ve analiz etmek için kapsamlı raporlama ve analitik araçları sunar. Bu araçlar sayesinde şirketler, müşteri taleplerinin yanıt sürelerini, müşteri memnuniyetini, destek ekibinin verimliliğini ve diğer önemli metrikleri izleyebilir. Böylece iyileştirme fırsatlarını tespit edebilir ve sürekli olarak müşteri deneyimini geliştirebilirler.

Bu özellikler, Zendesk'in müşteri hizmetleri süreçlerini etkin bir şekilde yönetmek ve müşteri memnuniyetini artırmak için kullanılan güçlü bir platform olduğunu gösterir.

Aşağıda zendesk’den örnek bir ekranlar verilmiştir.

A screenshot of a computer

Description automatically generated with medium confidence

Şekil 4.4: Zendesk ticket tarihçesi ekranı

A screenshot of a computer

Description automatically generated

Şekil 4.5: Zendesk login ekranı

## 4.3 Freshdesk

Freshdesk, 2010 yılında Girish Mathrubootham tarafından kurulan bir müşteri destek yazılımı şirketidir. Şirketin merkezi Chennai, Hindistan'da bulunmaktadır. Freshdesk, başlangıçta sadece müşteri destek yazılımı olarak başlayan bir girişimken, zaman içinde geniş bir iş süreci yönetimi platformuna dönüşmüştür.

Freshdesk, müşteri destek süreçlerini optimize etmek ve müşteri memnuniyetini artırmak için kullanıcı dostu bir çözüm sunmayı hedefler. Şirketlerin müşteri taleplerini almak, yönlendirmek ve yanıtlamak için kullanabilecekleri bir dizi özellik sunar.

Kuruluşundan sonra Freshdesk, hızla büyüyerek dünya çapında birçok şirketin tercih ettiği bir müşteri destek platformu haline geldi. 2011 yılında Freshdesk, TechCrunch Disrupt etkinliğinde "En İyi Hizmet" ödülünü kazanarak sektörde dikkatleri üzerine çekti.

Freshdesk, zamanla iş süreçleri yönetimi alanında genişleme yoluna gitti. 2012 yılında, Freshdesk'in sunduğu müşteri destek hizmetlerine ek olarak, CRM (Müşteri İlişkileri Yönetimi), SLA (Hizmet Seviyesi Anlaşması) yönetimi, analitik ve raporlama gibi iş süreçleri için yeni özellikler sunuldu.

2016 yılında Freshdesk, Freshworks Inc. adı altında bir yazılım şirketi ailesi olarak yeniden markalandı. Freshworks Inc., Freshdesk'in yanı sıra pazarlama, satış, insan kaynakları ve diğer iş süreçleri için çeşitli yazılım ürünleri sunan geniş bir ürün portföyüne sahiptir.

Günümüzde Freshdesk, müşteri hizmetleri ve destek süreçlerini yönetmek için birçok özelliğe sahip kapsamlı bir yazılım platformudur. Şirketler, Freshdesk'i müşteri taleplerini etkili bir şekilde yönetmek, SLA'ları izlemek, analitik verileri analiz etmek ve müşteri memnuniyetini artırmak için tercih etmektedir. Freshdesk, kolay kullanılabilir arayüzü ve özelleştirilebilir özellikleri ile müşteri hizmetleri alanında önemli bir aktör haline gelmiştir.

Freshdesk, müşteri destek süreçlerini yönetmek ve müşteri memnuniyetini artırmak için bir dizi özellik sunar. İşte Freshdesk'in temel özelliklerinden bazıları:

**Bilet Yönetimi:** Freshdesk, müşteri taleplerini biletler şeklinde kaydetmek ve yönetmek için kullanılır. Gelen talepler otomatik olarak bilet olarak oluşturulur ve önceliklerine göre kategorize edilir. Biletler, atama, takip ve zamanlama gibi işlemlere tabi tutulabilir.

**Çok Kanallı Müşteri İletişimi:** Freshdesk, e-posta, telefon, canlı sohbet, sosyal medya ve web formları gibi farklı iletişim kanallarından gelen müşteri taleplerini tek bir platformda birleştirir. Bu sayede şirketler, müşterilerle etkileşimlerini kolayca yönetebilir ve müşteri memnuniyetini artırabilir.

**Self-Servis Portalı:** Freshdesk, müşterilere kendi kendine yardım sağlamak için bir self-servis portalı sunar. Bu portal, sıkça sorulan sorular (SSS), bilgi tabanı, kullanıcı kılavuzları gibi kaynakları içerir. Müşteriler, kendi sorunlarını çözebilmek için bu kaynaklara erişebilir ve destek ekibinin yükünü azaltır.

**SLA Yönetimi:** Freshdesk, hizmet seviyesi anlaşmalarını (SLA) yönetmek için özellikler sunar. SLA'lar, müşteri taleplerinin belirli bir süre içinde yanıtlanması veya çözülmesi gerektiğini belirler. Freshdesk, SLA'ları otomatik olarak takip eder ve SLA'lara uymayan biletler için uyarılar veya eskalasyonlar sağlar.

**Raporlama ve Analitik:** Freshdesk, müşteri destek süreçlerini izlemek ve analiz etmek için kapsamlı raporlama ve analitik araçları sunar. Şirketler, bilet yanıt süreleri, müşteri memnuniyeti, performans metrikleri gibi verilere dayalı raporlar oluşturabilir ve destek süreçlerini iyileştirmek için verileri analiz edebilir.

**Entegrasyonlar:** Freshdesk, diğer iş süreçleri ve araçlarla entegrasyon imkanı sağlar. Örneğin, CRM sistemleri, sosyal medya platformları, e-ticaret platformları ve diğer yazılımlarla entegrasyon yaparak müşteri verilerini senkronize edebilir ve süreçlerinizi daha verimli hale getirebilirsiniz.

Aşağıda freshdesk’den örnek bir ekranlar verilmiştir.

A screenshot of a computer

Description automatically generated

Şekil 4.6: Freshdesk ana ekranı

**A screenshot of a computer

Description automatically generated**

Şekil 4.7: Fresh desk login sayfası

# Bölüm 5

# OsTicket Gerçek Hayat Örneği (Vesacons Başarı Hikayesi)

OsTicket, 2003 yılında Düşük Arka Uç Sistemleri (L.A.M.P) LLC tarafından kurulan bir açık kaynaklı proje olarak başladı. Projenin amacı, müşteri hizmetleri ve destek süreçlerini daha etkin bir şekilde yönetmek için kullanılabilecek bir yazılım sunmaktı.

OsTicket'in ilk sürümü, PHP programlama dili ve MySQL veritabanı kullanılarak geliştirildi. Yazılımın kaynak kodu, açık kaynak lisansı altında yayınlandı ve böylece kullanıcılar tarafından kolaylıkla erişilebilir ve geliştirilebilir hale geldi.

Geliştirme süreci boyunca, OsTicket'e yeni özellikler ve geliştirmeler eklendi. İlerleyen yıllarda kullanıcı arayüzü geliştirildi, performans iyileştirmeleri yapıldı ve kullanıcı geri bildirimleri doğrultusunda çeşitli güncellemeler yapıldı.

OsTicket, açık kaynaklı yapısı ve kullanım kolaylığı sayesinde popülerlik kazandı. Birçok küçük ve orta ölçekli işletme, müşteri destek süreçlerini OsTicket kullanarak yönetmeye başladı. Aynı zamanda birçok geliştirici ve yazılım topluluğu, OsTicket'e katkıda bulunarak yeni özellikler ekledi ve hataları düzeltti.

Günümüzde OsTicket, hala aktif olarak kullanılan ve geliştirilen bir açık kaynaklı projedir. Sürekli olarak güncellenen yeni sürümleri ve geliştirme topluluğu sayesinde, işletmeler müşteri destek süreçlerini yönetmek için güvenilir bir çözüm bulmaktadır. Ayrıca, OsTicket kullanıcıları, ihtiyaçlarına göre özelleştirme ve uyarlamalar yapma özgürlüğüne sahiptir.

OsTicket'in tarihçesi, müşteri destek süreçlerinin daha etkin ve verimli bir şekilde yönetilmesine yönelik bir çözüm olarak ortaya çıktığını göstermektedir. Açık kaynaklı yapısı ve sürekli geliştirme çabalarıyla OsTicket, birçok işletme için tercih edilen bir müşteri destek yazılımı haline gelmiştir.

Projemde kendi çalıştığım firmamda osticket yönetim sistemini tercih ettim.

Vesacons, SAP HR uzmanlarından oluşan bir ekip ile birlikte, müşterilerinin ihtiyaçlarına göre özelleştirilmiş ve uygun çözümler sunmaktadır. Bu süreçte, Vescons'un SAP HR danışmanları, şirketlerin insan kaynakları süreçlerini anlamak, mevcut iş süreçlerini analiz etmek ve SAP HR'nin uyumlu bir şekilde entegre edilmesini sağlamak için çalışırlar.

Vesacons, bir danışmanlık şirketi olarak müşteri memnuniyetini en üst düzeyde tutmayı hedefler. Bu doğrultuda, müşteri taleplerini etkili bir şekilde yönetmek ve hızlı yanıtlar sağlamak amacıyla OsTicket sistemini hayata geçirdi.

OsTicket'in Vescons'a sağladığı çözümler sayesinde, müşteri talepleri daha sistematik bir şekilde yönetilebildi. Her bir talep, OsTicket üzerinde bir bilet olarak kaydedildi ve ilgili departmana yönlendirildi. Bu sayede, taleplerin izlenmesi, atama ve takip süreçleri daha düzenli bir şekilde gerçekleştirildi.

OsTicket sisteminin uygulanmasıyla birlikte, Vesacons müşteri taleplerinin kapanma sürelerinde belirgin bir artış elde etti. Önceden sürecin daha karışık ve el ile yönetilmesi nedeniyle bazen taleplerin cevaplanması gecikebiliyordu. Ancak OsTicket'in sağladığı otomatikleştirmeler ve takip imkanı sayesinde, taleplerin hızlı bir şekilde yanıtlanması ve kapatılması sağlandı.

Bu gelişme sonucunda, Vesacons müşterileri arasında memnuniyet oranı arttı ve işletme itibarı güçlendi. Özellikle, ticket kapanma sürelerinde %30'luk bir artış sağlandı ve müşterilerin taleplerine daha hızlı ve etkin bir şekilde yanıt verildi.

Vesacons'un OsTicket sistemini uygulayarak elde ettiği başarı hikayesi, müşteri hizmetleri ve destek süreçlerinin daha verimli bir şekilde yönetilmesini göstermektedir. OsTicket'in sağladığı otomasyon ve takip imkanları sayesinde, Vescons müşterilerine daha iyi bir hizmet sunabilmiş ve müşteri memnuniyetini arttırmıştır.

Aşağıda Vesacons firmasının kullandığı osticket yönetim sistemlerinden ekran görüntüleri paylaşılmıştır.

Sistem 3 dilde destek vermektedir. Ticket sisteminin şirketteki adı myHrSuite olarak geçmektedir.

Ana ekran Vesacons projesi kapsamında özelleştirilmiştir.

Top of Form

**A screenshot of a computer

Description automatically generated**

Şekil 5. 1: Ana ekran

Vesacons müşterileri aşağıdaki ekran üzerinden talep açmaktadır.

**A screenshot of a computer

Description automatically generated**

Şekil 5.2: Yeni talep açma ekranı

Vesacons danışmanları aşağıdaki ekran üzerinden giriş yapmaktadır.

**A screenshot of a computer

Description automatically generated**

Şekil 5.3: Temsilci giriş ekranı

Giriş yaptıktan kullanıcı bilgileri vs değiştirilmektedir.

**A screenshot of a computer

Description automatically generated with medium confidence**

Şekil 5.4: Profil ekranı

Danışmanların talep karşılama ekranı aşağıdaki gibidir.

**A screenshot of a computer

Description automatically generated with medium confidence**

Şekil 5.5: Talep karşılama ekranı

Vesacons firmasının danışmanları kendi arasındaki talep paylaşımı aşağıdaki ekran üzerinden yapılmaktadır.

**A screenshot of a computer

Description automatically generated with medium confidence**

Şekil 5.6: Talep çözüm ekranı

Talepler aşağıdaki grafik üzerinden görsel olarak takip edilmektedir.

**A screenshot of a computer

Description automatically generated with medium confidence**

Şekil 5.7: Talep çözüm grafiği

Talep Yönetim sisteminde talep konularını kendi özelimizde tanımlama yapabiliyoruz.

**A screenshot of a computer

Description automatically generated**

Şekil 5.8: Talep yardım konuları tanımlama ekranı

# Bölüm 6

# Sonuç

Bu çalışma IT Help desk yönetimi sistemi geliştirilmiştir aşağıdaki teknolojiler kullanılmıştır.

Windows Form uygulaması ile birlikte ASP.NET Web API kullanarak token tabanlı kimlik doğrulama yöntemini uygulamayı ve güvenli bir uygulama geliştirmeyi amaçlamıştır. Bu projenin sonucunda, aşağıdaki başlıklarda önemli kazanımlar elde edilmiştir:

**Güvenli Kimlik Doğrulama**: Token tabanlı kimlik doğrulama yöntemi, kullanıcıların kaynaklara erişmek için token almasını ve bu token'ı kullanarak isteklerini doğrulamasını sağlar. Bu sayede, uygulamanın güvenliği artırılır ve yetkisiz erişimlere karşı koruma sağlanır.

**API Güvenliği:** ASP.NET Web API, token tabanlı kimlik doğrulama yöntemini kullanarak API güvenliğini sağlar. Token'lar, API'ye yapılan her isteğin doğrulanmasında kullanılır ve yetkilendirme işlemlerini gerçekleştirir. Bu sayede, sadece yetkilendirilmiş kullanıcıların API'ye erişim sağlaması sağlanır.

**Veri Güvenliği:** Token tabanlı kimlik doğrulama yöntemi, kullanıcıların hassas verilere erişirken güvenliklerini sağlamak için kullanılır. API üzerinden iletilen veriler, güvenli bir iletişim kanalı kullanılarak şifrelenir ve sadece yetkilendirilmiş kullanıcılar tarafından erişilebilir hale gelir.

**Kullanıcı Deneyimi:** Windows Form uygulaması, kullanıcı dostu arayüzlerin tasarlanmasına olanak sağlar. Kullanıcılar, kolaylıkla uygulamaya giriş yapabilir, kayıt olabilir ve ilgili işlemleri gerçekleştirebilir. Ayrıca, Swagger gibi araçlar kullanılarak API'nin belgelendirilmesi ve görsel arayüzü sağlanarak, kullanıcıların API'yi anlaması ve test etmesi kolaylaşır.

Bu projenin başarı kriterleri olan zaman ve verimlilik açısından da olumlu sonuçlar elde edilmiştir. Token tabanlı kimlik doğrulama yöntemi, geliştirme sürecinde zaman tasarrufu sağlar, çünkü kullanıcı kimlik doğrulama işlemleri otomatik olarak gerçekleştirilir ve her istekte tekrar tekrar kimlik doğrulama gerektirmez. Ayrıca, API güvenliği ve veri güvenliği sayesinde, kullanıcıların hassas verilere güvenli bir şekilde erişmesi sağlanır ve veri güvenliği konusunda etkin bir koruma sağlanır.

Ayrıca osticket sistemi sayesinde Vesacons müşterileri arasında memnuniyet oranı arttı ve işletme itibarı güçlendi. Özellikle, ticket kapanma sürelerinde %30'luk bir artış sağlandı ve müşterilerin taleplerine daha hızlı ve etkin bir şekilde yanıt verildi.

# Kaynaklar

<https://docs.microsoft.com/en-us/aspnet/web-api/>

"Building Web APIs with ASP.NET Web API 2.2" - Jamie Kurtz ve Brian Wortman

"Swagger and ASP.NET Web API: Building and Documenting Scalable RESTful Web Services" - Kevin Dockx

"Windows Forms Programming in C#" - Chris Sells

"Windows Forms Programming with C#" - Erik Brown