

WebPush Bildirim Sistemi Dokümantasyonu

1. Sistem Genel Bakış

WebPush bildirim sistemi, web sitesi ziyaretçilerine tarayıcı üzerinden bildirim göndermeyi sağlayan bir platformdur. Sistem; kullanıcı aboneliklerini yönetme, segment oluşturma, bildirim şablonları tasarlama, kampanyalar planlama ve bildirimler gönderme yeteneklerine sahiptir. Ayrıca, CDP (Müşteri Veri Platformu) ve CEP (Müşteri Etkileşim Platformu) entegrasyonları ile zenginleştirilmiştir.

Kimlik Doğrulama Notu: Sistemin kimlik doğrulama işlemi için gereken token'lar ve müşteri ID'leri harici bir uygulamadan sağlanmaktadır.

2. Aktörler

2.1. İnsan Aktörleri

1. Pazarlama Yöneticisi

- Kampanyaları oluşturur ve yönetir
- Bildirim şablonlarını tasarlar
- Sonuçları analiz eder

2. Sistem Yöneticisi

- Sistem yapılandırmasını yönetir
- Kullanıcı hesaplarını ve izinleri yönetir
- Webhook entegrasyonlarını ayarlar
- CDP ve CEP entegrasyonlarını yapılandırır

3. Son Kullanıcı

- Web sitesini ziyaret eder
- Bildirimlere abone olur veya olmayı reddeder
- Bildirimleri alır ve etkileşimde bulunur

2.2. Sistem Aktörleri

1. Tarayıcı

- Kullanıcının bildirimlere erişim izni vermesi için aracı olur
- Push token'ı oluşturur
- Bildirimleri gösterir

2. CDP (Müşteri Veri Platformu)

- Müşteri verilerini merkezi olarak depolar
- Kullanıcı profillerini zenginleştirir
- Segment oluşturmak için veri sağlar

3. CEP (Müşteri Etkileşim Platformu)

- Kullanıcıya en uygun iletişim kanalını belirler
- Gönderim zamanlamasını optimize eder
- Kanal performansını değerlendirir

3. Varlıklar (Entities)

1. Kullanıcı (User)

- Kimlik bilgileri
- İletişim bilgileri
- Tercih bilgileri
- Push abonelik durumu ve token'ı

2. Rol (Role)

- Rol tanımı
- İzin grupları

3. İzin (Permission)

- Eylem türleri
- Kaynak türleri
- Erişim düzeyleri

4. Segment

- Segment adı ve açıklaması
- Segment tipi (Statik, Dinamik, İçer Aktarılmış)
- Filtre kriterleri
- Kullanıcı listesi

5. Şablon (Template)

- Şablon adı
- Bildirim başlığı ve içeriği
- Medya öğeleri
- Eylem URL'i
- Dinamik değişkenler

6. Kampanya (Campaign)

- Kampanya adı ve açıklaması
- Kampanya türü (Tek seferlik, Tekrarlanan, Tetikleyici bazlı, A/B Test)
- Hedef segment
- Seçilen şablon
- Zamanlama bilgileri

7. Tetikleyici (Trigger)

- Tetikleyici adı
- Olay türü
- Koşullar
- Gecikme süresi
- Hedef segment

8. Analitik (Analytics)

- Gönderim istatistikleri
- Teslim istatistikleri
- Açılma oranları
- Tıklanma oranları

- Dönüşüm verileri

9. Webhook

- Webhook adı ve açıklaması
- Endpoint URL'i
- Tetikleyici olaylar
- Güvenlik bilgileri

4. Aktiviteler ve İş Akışları

4.1. Kullanıcı Abonelik Akışı

1. Kullanıcı web sitesini ziyaret eder
2. Sistem, kullanıcının abonelik durumunu kontrol eder
3. Eğer kullanıcı abone değilse, abonelik istemi gösterilir
4. Kullanıcı aboneliği kabul ederse, tarayıcı izni istenir
5. İzin verilirse, push token oluşturulur ve saklanır
6. Onay bildirimi gönderilir
7. Kullanıcı "Abone" segmentine eklenir
8. Eğer abonelik zaten varsa, geçerli olup olmadığı kontrol edilir
9. Geçerli değilse, yenileme denemesi yapılır

4.2. Segment Yönetimi Akışı

1. Yönetici yeni bir segment oluşturur
2. Segment için isim ve açıklama belirlenir
3. Segment tipi seçilir (Statik, Dinamik, İç Aktarılmış)
4. Statik segment için kullanıcılar manuel olarak seçilir
5. Dinamik segment için filtre kriterleri tanımlanır
6. İç aktarma için kullanıcı listesi yüklenir
7. Segment kaydedilir ve büyüklüğü hesaplanır
8. CDP entegrasyonu etkinse, veriler zenginleştirilir
9. Segment etiketleri eklenir ve son değişiklikler kaydedilir

4.3. Şablon Oluşturma Akışı

1. Yönetici yeni bir şablon oluşturur
2. Şablon adı belirlenir
3. Bildirim başlığı ve içeriği yazılır
4. Medya ögesi eklemek istenirse, görüntü yüklenir
5. Görüntü boyutu optimize edilir
6. Eylem URL'i ayarlanır
7. Dinamik değişkenler tanımlanır ve varsayılan değerler atanır
8. Şablon önizlenir ve farklı cihazlarda görünümü kontrol edilir
9. Şablon kaydedilir ve etiketlenir
10. Şablon versiyonu oluşturulur

4.4. Kampanya Oluşturma Akışı

1. Yönetici yeni bir kampanya oluşturur
2. Kampanya adı ve açıklaması belirlenir
3. Kampanya türü seçilir
4. Seçilen türe göre içerik tanımlanır
5. Bildirim şablonu seçilir

6. Kişiselleştirme ayarları yapılır
7. Hedef kitle seçilir
8. Kampanya önizlenir ve kontrol edilir
9. Kampanya doğrulanır
10. Kampanya zamanlanır, kaydedilir ve aktifleştirilir

4.5. Bildirim Gönderim Akışı

1. Kampanya zamanı geldiğinde, kampanya detayları alınır
2. Hedef segment kullanıcıları ve bildirim şablonu getirilir
3. Her kullanıcı için:
 - Kullanıcının abone olup olmadığı kontrol edilir
 - Mesaj kişiselleştirilir
 - CEP entegrasyonu etkinse, en uygun kanal belirlenir
 - WebPush uygunsa, bildirim hazırlanır ve gönderilir
 - Gönderim kaydedilir ve durumu takip edilir
 - Tıklama ve açılma izlemesi ayarlanır
4. Kampanya istatistikleri güncellenir
5. İlgili webhook'lar tetiklenir

4.6. CDP Entegrasyon Akışı

1. Kullanıcı aktivitesi gerçekleştiğinde veri toplanır
2. Veri kaynağına göre (Web, Uygulama, WebPush) veri yakalanır
3. Veriler normalize edilir
4. CDP için veri paketi hazırlanır
5. Veriler CDP'ye gönderilir
6. CDP verileri işler ve kullanıcı profilini günceller
7. Gerekliyse, veriler WebPush sistemine geri senkronize edilir
8. Kullanıcı segmentleri yeniden hesaplanır

4.7. Analitik İzleme Akışı

1. Bildirim gönderildiğinde, gönderim olayı kaydedilir
2. Teslimat durumu beklenir ve kaydedilir
3. Açılma durumu izlenir ve kaydedilir
4. Tıklama durumu izlenir ve kaydedilir
5. Hedef sayfada etkileşim takip edilir
6. Dönüşüm durumu izlenir
7. Kampanya istatistikleri hesaplanır
8. Raporlar güncellenir
9. Analitik webhook'ları tetiklenir
10. CDP entegrasyonu varsa, veriler CDP'ye gönderilir

4.8. Webhook İşleme Akışı

1. Sistemde bir olay gerçekleştiğinde, kayıtlı webhook'lar kontrol edilir
2. Eşleşen webhook'lar için veri paketi hazırlanır
3. Güvenlik imzası eklenir
4. Her webhook için:
 - Webhook'un aktif olup olmadığı kontrol edilir
 - HTTP isteği gönderilir
 - Yanıt beklenir ve doğrulanır
 - Sonuç kaydedilir

5. Gerekirse yeniden deneme yapılır

4.9. A/B Test Akışı

1. Yeni A/B testi oluşturulur ve kampanyaya bağlanır
2. Test varyantları tanımlanır
3. Her varyant için şablon oluşturulur
4. Dağıtım yüzdeleri belirlenir
5. Başarı metrikleri belirlenir
6. Test hedef kitlesi ve örnek büyüklüğü tanımlanır
7. Test süresi ayarlanır
8. Test etkinleştirilir ve gönderimleri varyantlara dağıtılır
9. Performans verileri toplanır
10. Test tamamlandığında sonuçlar analiz edilir
11. Kazanan varyant belirlenir ve kampanyaya uygulanır

4.10. CEP Karar Akışı

1. İletişim tetiklendiğinde, kullanıcı profili ve bağlamsal veriler alınır
2. Kullanıcı geçmişini analiz edilir
3. Kanal tercihleri kontrol edilir
4. Optimal zamanlama belirlenir
5. Mevcut kanallar değerlendirilir ve her birine puan atanır
6. Kanallar karşılaştırılır ve en yüksek puanlı kanal seçilir
7. WebPush seçilirse, bildirim parametreleri optimize edilir
8. Optimal zaman için zamanlama yapılır
9. Bildirim yürütülür ve performansı izlenir
10. Sonuçlar CEP'e geri beslenir

5. Entegrasyonlar

5.1. CDP (Müşteri Veri Platformu) Entegrasyonu

- **Veri Senkronizasyonu:** Kullanıcı profil verileri iki yönlü olarak senkronize edilir
- **Segment Zenginleştirme:** CDP verileri kullanılarak segmentler zenginleştirilir
- **Kişiselleştirme:** CDP'deki kullanıcı verileri bildirim kişiselleştirmede kullanılır
- **Performans Analizi:** Bildirim performans verileri CDP'ye gönderilir

5.2. CEP (Müşteri Etkileşim Platformu) Entegrasyonu

- **Kanal Seçimi:** Kullanıcı ile iletişim için en uygun kanal belirlenir
- **Zamanlama Optimizasyonu:** En yüksek etkileşim için optimal gönderim zamanı hesaplanır
- **Performans Geri Beslemesi:** Bildirim performans verileri CEP algoritmasını besler
- **Kanal Koordinasyonu:** Diğer iletişim kanallarıyla çakışmaların önlenmesi sağlanır

5.3. Webhook Entegrasyonu

- **Olay Tetikleme:** Sistem içindeki olaylar dış sistemleri tetikler
- **Veri Paylaşımı:** Bildirim ve etkileşim verileri dış sistemlere aktarılır
- **İşlem Doğrulama:** Dış sistemlerden gelen yanıtlar işlenir
- **Güvenlik:** Webhook talepleri güvenlik imzasıyla korunur

5.4. Kimlik Doğrulama Entegrasyonu

- **Harici Token:** Kimlik doğrulama token'ları harici bir uygulamadan sağlanır
- **Müşteri ID:** Her istek müşteri ID'si içerir

- **Yetkilendirme:** Roller ve izinler merkezi kimlik doğrulama sistemiyle eşleştirilir

6. Genel Sistem Mimarisi

WebPush bildirim sistemimiz dört ana bileşenden oluşmaktadır:

1. **Kullanıcı Yönetimi:** Abonelikler ve segmentasyon
2. **İçerik Yönetimi:** Şablonlar, kampanyalar ve testler
3. **Bildirim Sistemi:** Gönderim motoru ve optimizasyon
4. **Veri ve Analitik:** İzleme, raporlama ve entegrasyonlar

Bu bileşenler, kullanıcılara doğru zamanda, doğru içerikle, doğru kanaldan ulaşmak için birlikte çalışır.

Not: Bu doküman, WebPush bildirim sistemimizin tasarımı için wireframe çalışmalarında kullanılmak üzere hazırlanmıştır. Sistemin işleyişi ve entegrasyonlar hakkında genel bir bakış sunmaktadır.

Pandoc Dönüşüm Komutları

```
pandoc sistem_dokumantasyonu.md -o sistem_dokumantasyonu.pdf
pandoc teknik_arayuz.md -o teknik_arayuz.pdf
```