

# **Yazılım Test Planı**

*Joplin*

Tuğcan Topaloğlu

Yıldız Teknik Üniversitesi

02.01.2025

## İçindekiler

1.	Giriş.....	3
2.	Test Edilecek Modüller .....	3
3.	Test Stratejisi .....	4
4.	Test Senaryoları.....	5
5.	Test Ortamı .....	7
6.	Test Süreci .....	7
7.	Test Sonuçları Değerlendirme .....	11

# 1. Giriş

## 1.1 Amaç

Bu test planı, Joplin uygulamasının işlevsel, performans, güvenlik ve kullanılabilirlik gereksinimlerini karşılayıp karşılamadığını doğrulamak için oluşturulmuştur. Plan, sistemin hatasız çalışmasını ve kullanıcıların ihtiyaçlarına uygun bir deneyim sunmasını sağlamak için gerekli test süreçlerini tanımlar.

## 1.2 Kapsam

Joplin uygulaması, kullanıcıların kişisel notlarını kolaylıkla oluşturmaya, düzenlemesine, organize etmesine ve paylaşmasına olanak tanır. Bu test planı, not oluşturma, düzenleme, silme ve listeleme gibi temel işlemlerin yanı sıra etiket ekleme ve kaldırma gibi daha karmaşık işlevlerin doğrulanmasını içerir. Ayrıca, veri içe ve dışa aktarma işlemleri, Markdown, PDF ve JSON formatlarında test edilecektir. Uygulamanın senkronizasyon özelliği, çevrimdışı mod desteğiyle birlikte değerlendirilirken, performans, güvenlik ve kullanılabilirlik gibi kritik gereksinimlerin karşılanması da planın bir parçasıdır.

## 1.3 Hedef Kitle

Proje yöneticileri test sürecini izlemek ve yönlendirmekten sorumludur. Test uzmanları, tanımlanan test senaryolarını uygulayarak yazılımın doğru çalışıp çalışmadığını doğrular. Geliştiriciler, tespit edilen hataları düzeltir ve sistemi iyileştirir. Son kullanıcılar ise gerçek kullanım senaryolarında sistemi test ederek doğruluğunu değerlendirir.

# 2. Test Edilecek Modüller

Joplin uygulaması içerisinde test edilecek temel modüller şunlardır:

- **Not Yönetimi:** Kullanıcıların not oluşturma, düzenleme, silme ve sıralama işlemleri için kullanılan modül. Bu modül, tüm not işlevlerinin doğru şekilde çalıştığını doğrulamak için test edilecektir.
- **Etiket Yönetimi:** Notlara etiket ekleme, kaldırma ve bu etiketler üzerinden filtreleme işlemleri yapılmasını sağlayan modül. Etiketlerin notlarla doğru ilişkilendirilmesi ve arama sonuçlarına yansması test edilecektir.
- **Veri İçe/Dışa Aktarma:** Notların Markdown, PDF ve JSON formatlarında dışa aktarımı ve bu formatlarda verilerin içe aktarımı için kullanılan modül. Bu süreçlerin eksiksiz çalıştığı doğrulanacaktır.
- **Senkronizasyon:** Uygulamanın çevrimdışı mod ve bulut hizmetleri arasında veri eşitleme işlemleri. Veri bütünlüğünün senkronizasyon sırasında korunup korunmadığı test edilecektir.
- **Arama ve Filtreleme:** Kullanıcıların notlarda tam metin arama ve etiket filtreleme işlemlerini gerçekleştirebildiği modül. Doğru sonuçların döndürüldüğü doğrulanacaktır.
- **Kullanıcı Deneyimi:** Menü navigasyonu, tema değişikliği ve dokunmatik cihazlarda etkileşim gibi kullanıcı deneyimi unsurları test edilecektir. Kullanıcı dostu bir arayüz sağlandığı doğrulanacaktır.

- **Güvenlik:** SQL enjeksiyonu, XSS koruması, şifreleme ve yetkisiz erişim koruması gibi kritik güvenlik gereksinimlerini sağlayan modül. Zararlı veri girişlerinin ve yetkisiz erişimlerin engellendiği doğrulanacaktır.
- **Performans:** Sistem yanıt süresi, bellek ve CPU kullanımı gibi performans gereksinimlerini test eden modül. Uygulamanın kaynak verimliliği analiz edilecektir.

### 3. Test Stratejisi

#### 3.1 Test Türleri

Test süreci, sistemin tüm özelliklerini farklı boyutlardan değerlendirebilmek için çeşitli test türlerini kapsamaktadır:

- **Birim Testleri:** Yazılımın temel yapı taşları (fonksiyonlar, sınıflar vb.) izole edilmiş ortamlarda test edilerek doğru çalışıp çalışmadıkları denetlenir. Örneğin, bir not oluşturma fonksiyonunun doğru şekilde bir not oluşturup oluşturmadığı kontrol edilir.
- **Entegrasyon Testleri:** Birim testleri onaylanmış farklı yazılım modüllerinin birlikte ve birbirleri ile etkileşimleri doğrulanır. Özellikle farklı sistem parçalarının uyum içinde çalıştığının kanıtlanmasına önem verilir. Örneğin, not oluşturma işlemi sırasında eklenen etiketlerin doğru şekilde kaydedilip gösterildiği kontrol edilir.
- **Sistem Testleri:** Tüm sistemin, belirlenen işlevsel ve işlevsel olmayan gereksinimlere tam uyum sağladığının test edilmesidir. Örneğin, tüm modüllerin birlikte çalıştığı bir senaryoda veritabanı tutarlılığı ve arayüz entegrasyonu doğrulanır.
- **Kabul Testleri:** Sistemin gerçek dünya koşullarında, kullanıcı beklentilerini karşılama düzeyini belirlemeye yöneliktir. Örneğin, yeni kullanıcı rehberinin tamamlanması ve temaların doğru şekilde çalışması gibi kullanıcı odaklı işlemler bu test kapsamında ele alınır.
- **Fonksiyonel Testler:** Sistemin temel fonksiyonlarını doğrular ve her fonksiyonun beklediği şekilde çalışıp çalışmadığını inceler. Örneğin, not arama özelliğinin doğru sonuçlar döndürüp döndürmediği test edilir.
- **Performans Testleri:** Sistemin performansını değerlendirir. Yanıt verme hızı, veri işleme kapasitesi ve kaynak kullanımı (işlemci, bellek, ağ) gibi sistemin genel verimi analiz edilir ve belirlenen süre limitleri içinde kalıp kalmadığı değerlendirilir.
- **Güvenlik Testleri:** Olası güvenlik açıklarını belirlemeye yönelik, saldırı ve sızma denemeleri ile sistemin genel güvenilirliği değerlendirilir. Örneğin, SQL enjeksiyonu ve XSS saldırılarına karşı sistemin dayanıklılığı test edilir.
- **Kullanılabilirlik Testleri:** Sistem arayüzünün, kullanıcıların kolay, etkili ve memnuniyet duyarak kullanabileceği ölçüde tasarlanıp tasarlanmadığı belirlenir. Menü gezintisi ve kullanıcı rehberinin etkinliği bu kapsamda test edilir.

#### 3.2 Test Kapsamı

Test planı, Joplin uygulamasının aşağıdaki gereksinimlerini kapsamaktadır:

- **Fonksiyonel Gereksinimler:** Not yönetimi, etiketleme, arama, senkronizasyon ve veri yönetimi.

- **Performans Gereksinimleri:** Sistem hızının ve kaynak kullanımı etkinliğinin doğrulanması.
- **Güvenlik Gereksinimleri:** Zararlı veri girişlerine ve yetkisiz erişimlere karşı koruma.
- **Kullanılabilirlik Gereksinimleri:** Arayüzün kullanıcı dostu ve erişilebilir olması.

### 3.3 Test Metodolojisi

- **Birim Testleri:** Her modülün bağımsız işlevselliği test edilecektir. Örneğin, "Not oluşturma" modülü için bir notun doğru şekilde oluşturulup oluşturulmadığı kontrol edilecektir. "Etiket ekleme" modülünde ise bir etiketin doğru şekilde ilişkilendirilip ilişkilendirilmediği doğrulanacaktır.
- **Entegrasyon Testleri:** Modüller arasındaki veri akışı ve işbirliği doğrulanacaktır. Örneğin, "Not Yönetimi" ile "Etiket Yönetimi" arasındaki etkileşim test edilerek bir nota eklenen etiketin doğru şekilde kaydedildiği ve arama sonuçlarına yansıdığı kontrol edilecektir. Ayrıca "Senkronizasyon" ve "Veri İçer/Dışa Aktarma" modülleri arasındaki veri akışının doğru çalıştığı doğrulanacaktır.
- **Sistem Testleri:** Sistemin bir bütün olarak çalışması test edilecektir. Bu kapsamda, "Not Yönetimi" ile "Etiket Yönetimi" arasındaki etkileşimlerin doğruluğu, veri içe ve dışa aktarma işlemlerinin eksiksiz çalışması ve tüm kullanıcı arayüzü elementlerinin entegrasyonu öncelikli olarak test edilecektir. Ayrıca, uygulamanın senkronizasyon işlemleri sırasında veri tutarlılığını koruyup korumadığı kontrol edilecektir.
- **Kabul Testleri:** Gerçek kullanım senaryolarında son kullanıcılar tarafından yapılacak testlerdir. Örneğin, bir kullanıcı rehberinin başarıyla tamamlanması, tema değişikliği işleminin sorunsuz gerçekleştirilmesi ve menü navigasyonunun kolayca anlaşılması testin başarılı sayılması için kriterler arasındadır. Ayrıca, yeni kullanıcıların sisteme sorunsuz adapte olması ve dokunmatik cihazlarda etkileşimlerin doğru şekilde çalışması kabul testlerinin önemli unsurlarıdır.

### 3.4 Kullanılacak Araçlar

- **Python:** API Test senaryolarını uygulamak için.
- **Postman:** API testleri için.
- **Jest:** Birim testler için.

## 4. Test Senaryoları

Test senaryoları, Joplin uygulamasının tüm işlevselliğini kapsamlı bir şekilde doğrulamak için tasarlanmıştır. Her bir senaryo, uygulamanın belirli bir modülünü veya özelliğini test etmeye yöneliktir ve belirli bir amaca hizmet eder. Bu senaryolar, uygulamanın işlevsel gereksinimlerini karşılayıp karşılamadığını doğrulamanın yanı sıra, performans, güvenlik ve kullanılabilirlik açısından da uygunluğunu değerlendirmeyi amaçlar. Her test senaryosu, uygulamanın hem olağan hem de beklenmeyen koşullar altında nasıl çalıştığını anlamak için özenle hazırlanmıştır. Aşağıdaki test senaryoları, modül bazlı ve detaylı olarak sunulmuştur.

## 4.1 Fonksiyonel Testler

- **TC001:** Not oluşturma. Kullanıcı bir not oluşturduğunda, notun veritabanına doğru bir şekilde kaydedildiği ve kullanıcı arayüzünde görüntülediği doğrulanır.
- **TC002:** Not güncelleme. Var olan bir notun içeriği güncellendiğinde, değişikliklerin veritabanında ve arayüzde doğru şekilde yansıtıldığı kontrol edilir.
- **TC003:** Not silme. Silinen notların veritabanından tamamen kaldırıldığı ve çöp kutusuna taşındığı doğrulanır.
- **TC004:** Etiket ekleme. Kullanıcı, bir nota etiket eklediğinde, bu etiketin notla ilişkilendirildiği ve arama sonuçlarında görüldüğü doğrulanır.
- **TC005:** Etiket kaldırma. Etiketin nottan kaldırıldığında ilişkisinin tamamen kesildiği kontrol edilir.
- **TC006:** Markdown formatında dışa aktarma. Notlar Markdown formatında başarıyla dışa aktarılmalıdır.
- **TC007:** PDF formatında dışa aktarma. Notlar PDF formatında dışa aktarılabilir olmalıdır.
- **TC008:** Not arama. Kullanıcı belirli bir anahtar kelimeyle arama yaptığında doğru sonuçlar döndürülmelidir.
- **TC009:** Notları sıralama. Notlar, oluşturulma veya güncellenme tarihlerine göre sıralanabilmelidir.
- **TC010:** Çöp kutusundan geri yükleme. Silinen notların çöp kutusundan geri yüklenmesi sırasında bütün veri eksiksiz olmalıdır.

## 4.2 Performans Testleri

- **TC014:** Uygulama başlangıç süresi. Uygulamanın başlangıç süresi 3 saniyeyi geçmemelidir.
- **TC016:** Arama yanıt süresi. Kullanıcı sorgularına verilen yanıt süreleri 1 saniyeden az olmalıdır.
- **TC017:** Büyük notları yükleme. 10.000 karakterlik notlar 3 saniyeden kısa sürede yüklenmelidir.
- **TC018:** Çoklu not yükleme. Aynı anda birçok not yüklendiğinde performans kaybı yaşanmamalıdır.
- **TC019:** Senkronizasyon süresi. Eşitleme işlemleri maksimum 5 saniyede tamamlanmalıdır.
- **TC021:** CPU kullanımı. Uygulama çalışırken CPU kullanımı %50'nin altında olmalıdır.
- **TC022:** Bellek kullanımı. Bellek kullanımı 100 MB'ı geçmemelidir.

## 4.3 Güvenlik Testleri

- **TC024:** SQL enjeksiyonu koruması. Zararlı SQL sorguları engellenmelidir.
- **TC026:** XSS koruması. Zararlı JavaScript kodlarının çalıştırılması önlenmelidir.
- **TC028:** Şifreleme kontrolü. Kullanıcı verileri şifrelenmiş bir şekilde saklanmalıdır.
- **TC029:** Yetkisiz erişim koruması. Yalnızca yetkili kullanıcıların verilere erişmesi sağlanmalıdır.
- **TC031:** Güvenlik loglarının doğrulanması. Tüm güvenlik olayları kaydedilmelidir.

## 4.4 Kullanılabilirlik Testleri

- **TC047:** Menü navigasyonu. Kullanıcı, menüler arasında sorunsuz geçiş yapabilmelidir.
- **TC048:** Yeni kullanıcı rehberi. İlk kez oturum açan kullanıcılar için rehber sağlanmalıdır.
- **TC050:** Kısayol tuşları. Tüm temel işlemler için kısayol tuşları desteklenmelidir.
- **TC051:** Tema değişikliği. Kullanıcılar, açık ve koyu tema arasında geçiş yapabilmelidir.
- **TC052:** Çoklu dil desteği. Uygulama farklı dillerde kullanılabilir olmalıdır.
- **TC053:** Dokunmatik destek. Mobil cihazlarda dokunmatik kontroller doğru çalışmalıdır.

## 5. Test Ortamı

### 5.1 Donanım Gereksinimleri

- Minimum 4 GB RAM.
- 2 GHz çift çekirdekli işlemci.
- 500 MB boş disk alanı.
- Testleri Jest ile çalıştırabilmek ve Devbox kullanabilmek adına Linux işletim sistemi.
- Uygulamayı test edebilmek için mobil bir telefon.
- Yine test amacıyla Windows bilgisayar.

### 5.2 Yazılım Gereksinimleri

- Node.js 14 veya üzeri.
- Python 3.10 veya üzeri.
- MySQL veri tabanı.
- Devbox
- Jest kütüphanesi

## 6. Test Süreci

Test süreci, Joplin uygulamasının tüm modüllerini ve özelliklerini eksiksiz bir şekilde değerlendirmek için planlanmıştır. Her bir aşama dikkatle tasarlanmış ve uygulamanın farklı gereksinimlerini karşılayacak şekilde yapılandırılmıştır. Süreç aşağıdaki adımlardan oluşur.

### 6.1 Test Ortamı

- **İşletim Sistemi:** Windows ve Linux kullanılacaktır.
- **Tarayıcı:** Chrome tarayıcısı kullanılacaktır.
- **Geliştirme Ortamı:** Node.js, Python 3.x, MySQL, Devbox ve Joplin'e özel bağımlılıklar.
- **Test Araçları:**
  - Python: Fonksiyonel ve performans testleri için.

- Postman: API testleri için.
- Jest: Birim testleri için.
- SonarQube: Güvenlik açıklarını analiz etmek için.

- **Donanım Gereksinimleri:** Minimum 4 GB RAM ve çift çekirdekli işlemci.

## 6.2 Test Adımları (V-Model Odaklı)

### 6.2.1 V-Model Açıklaması

V-Model yazılım geliştirme sürecinde, her geliştirme aşamasını bir test aşaması ile eşleştirerek sistemin tüm yönlerini erken dönemde doğrulamayı amaçlar. Bu modelin temel felsefesi, testlerin yazılım geliştirme süreci boyunca paralel olarak planlanması ve uygulanmasıdır. Aşağıda test adımları detaylı olarak açıklanmıştır:

### 6.2.2 Adım Adım Test Süreci

- **Gereksinim Analizi Aşaması:**
  - **Test Planı İncelemesi:** Test planı, proje paydaşları tarafından gözden geçirilir ve onaylanır. Planın kapsamı ve hedefleri netleştirilir.
  - **Kabul Kriterlerinin Tanımlanması:** Her kullanıcı hikayesi ve gereksinim için kabul kriterleri belirlenir.
- **Sistem Tasarımı Aşaması:**
  - **Sistem Test Vaka Geliştirme:** Sistem test senaryoları, sistem tasarım dokümanlarına dayanılarak oluşturulur. Bu senaryolar, sistemin genel tasarımı ile uyumlu olmalıdır.
- **Modül Tasarımı Aşaması:**
  - **Entegrasyon Test Vaka Geliştirme:** Modüller arasındaki veri akışını ve etkileşimleri doğrulamak için entegrasyon test senaryoları hazırlanır.
- **Uygulama Geliştirme Aşaması:**
  - **Test Vakaları Oluşturma:** Her özellik için detaylı test vakaları hazırlanır. Bu vakalar, senaryoları, test verilerini ve beklenen sonuçları içerir.
  - **Test Verisi Hazırlama:** Testler sırasında kullanılacak gerçekçi veri setleri oluşturulur. Örneğin, not içeriği, etiketler ve kullanıcı profilleri gibi.
- **Birim Testleri:**
  - **Birim Test Çalıştırma:** Modüller tamamlandıktan sonra birim testler uygulanır. Bu testler, kodun beklenen şekilde çalıştığını doğrular.
- **Entegrasyon Testleri:**
  - **Entegrasyon Test Çalıştırma:** Modüller bir araya getirildikten sonra entegrasyon testleri uygulanır. Modüller arasındaki veri akışının doğru çalıştığı doğrulanır.
- **Sistem Testleri:**
  - **Sistem Test Çalıştırma:** Sistemin tamamı test edilerek işlevsel, performans ve işlevsel olmayan gereksinimlerin karşılanıp karşılanmadığı değerlendirilir.
- **Kabul Testleri:**
  - **Kabul Test Çalıştırma:** Gerçek kullanıcı senaryoları uygulanır. Sistem, iş gereksinimlerini karşılayıp karşılamadığı açısından değerlendirilir.
  - **İzlenebilirlik Matrisi Oluşturma:** Her testin hangi gereksinimi karşıladığını belgelemek için bir izlenebilirlik matrisi oluşturulur.



- **Risk Tabanlı Testler:** Yüksek risk taşıyan özelliklere öncelik verilerek testler yürütülür.
- **Otomasyon:**
  - **Test Otomasyonu Sağlama:** Tekrarlı testlerin otomatikleştirilmesiyle zamandan tasarruf sağlanır ve testlerin daha geniş kapsamda uygulanması mümkün hale gelir.
- **Test Sonuçlarını Kaydetme:**
  - Her testin sonuçları detaylı olarak belgelenir. Başarılı ve başarısız testler, ilgili hata notlarıyla birlikte kaydedilir.
- **Hata Raporlama:**
  - Bulunan hatalar ayrıntılı olarak raporlanır ve hata takip sistemine kaydedilir.
- **Test Dokümantasyonu:**
  - Test süreçleri ve sonuçları eksiksiz bir şekilde dokümanite edilir.
- **Test Raporu Hazırlama:**
  - Test süreci, sonuçlar, analizler ve öneriler bir test raporu ile özetlenir.

### 6.3 İzlenebilirlik Matrisi

İzlenebilirlik matrisi, test senaryolarının hangi gereksinimleri karşıladığını açıkça göstermek için oluşturulmuştur. Aşağıdaki tablo, Joplin uygulaması için oluşturulan gereksinimler ile test senaryolarını ilişkilendirmektedir bu matris aşağıdaki tablo 1’de görülebilir. İlişkilendirilmemiş testler görsel ve kullanıcı kabul testleridir:

Gereksinim ID	Test Senaryosu ID	Açıklama
FR1	TC001, TC002	Not oluşturma ve düzenleme
FR2	TC003, TC004, TC005	Not silme ve etiket yönetimi
FR3	TC006, TC007, TC008	Veri dışı aktarma ve arama
FR4	TC009, TC010	Çöp kutusu işlemleri
FR5	TC011, TC012, TC013	Favorilere ekleme ve şifreleme
PR1	TC014, TC015	Uygulama başlangıç ve kaydetme süresi
PR2	TC016, TC017, TC018	Arama ve yükleme süreleri
PR3	TC019, TC021, TC022	Senkronizasyon ve kaynak kullanımı
RR1	TC023, TC024, TC025	Hatalı girişler ve güvenlik önlemleri
RR2	TC026, TC027	XSS koruması ve oturum süresi
UR1	TC047, TC048	Menü navigasyonu ve kullanıcı rehberi
UR2	TC050, TC051, TC052	Kısayol tuşları ve dil desteği

Tablo 1. İzlenebilirlik matrisi

## 6.4 Risk Analizi

Test planında, sistemin kritik modüllerinde veya işlevlerinde risk analizi yapılmıştır. Aşağıda risk seviyelerine göre sınıflandırılmış modüller bulunmaktadır:

- **Yüksek Risk:**
  - Kullanıcı oturum yönetimi (TC023, TC027).
  - Veritabanı senkronizasyonu (TC019).
- **Orta Risk:**
  - Çevrimdışı mod desteği (TC019, TC031).
  - Güvenlik açıklarının önlenmesi (TC024, TC026).
- **Düşük Risk:**
  - Menü navigasyonu ve tema değişikliği (TC047, TC051).

## 6.5 Test Verisi Hazırlığı

Testlerde kullanılacak veri türleri ve hazırlanış yöntemleri:

- **Not Yönetimi:**

```
{
  "title": "Test Note",
  "body": "This is a test note content."
}
```

Şekil 1. Örnek veri

- **Etiket Yönetimi:**

```
{
  "title": "Test Tag",
  "note_id": "12345"
}
```

Şekil 2. Örnek veri 2

- **Performans Testleri:**
  1. 1000 notun eş zamanlı oluşturulması ve sorgulanması için rastgele oluşturulan metinler kullanılacaktır.
- **Güvenlik Testleri:**
  1. SQL enjeksiyon saldırıları için örnek payload: ' OR '1'='1'.
  2. XSS saldırıları için zararlı JavaScript kodları: <script>alert('XSS');</script>.

## 6.6 Performans Test Detayları

Performans testlerinde ölçülecek metrikler:

- **Maksimum Yanıt Süresi:** 3 saniyeyi geçmemelidir.
- **CPU Kullanımı:** %50'nin altında olmalıdır.

- **Bellek Kullanımı:** 100 MB'ı geçmemelidir.

Kullanılacak araçlar:

- **Python:** Yük testi için.
- **Postman:** API performans testi için.

## 7. Test Sonuçları Değerlendirme

### 7.1 Başarı Kriterleri

- **Fonksiyonel Testler:** Uygulamanın tüm temel fonksiyonları (not oluşturma, güncelleme, silme, vb.) doğru şekilde çalışmalıdır. Belirtilen gereksinimlerin %100 karşılanması sağlanmalıdır.
- **Performans Testleri:** Uygulama yanıt süreleri, bellek ve CPU kullanımı gibi performans metrikleri belirtilen sınırlar içinde olmalıdır. Örneğin, yanıt süresi 3 saniyeyi geçmemelidir.
- **Güvenlik Testleri:** Hiçbir güvenlik açığı bulunmamalıdır. SQL enjeksiyonu, XSS gibi saldırılara karşı sistem dayanıklı olmalıdır.
- **Kullanılabilirlik Testleri:** Sistem, kullanıcı dostu bir arayüz sunmalı ve tüm kullanıcı işlemleri sorunsuz bir şekilde gerçekleştirilebilmelidir.
- **Veri İşleme Doğruluğu:** Uygulama, verilerin doğru bir şekilde işlenmesini, saklanmasını ve görüntülenmesini sağlamalıdır.

### 7.2 Hata Yönetimi

- **Hataların Kaydedilmesi:** Tespit edilen tüm hatalar hata takip sistemine kaydedilir. Bu kayıtlar, hatanın türünü, önem düzeyini (kritik, orta, düşük) ve çözüm önerilerini içermelidir.
- **Hataların Düzeltme Süreci:** Hatalar geliştirici ekiplere iletilir ve gerekli düzeltmeler yapılır. Düzeltme sonrası, ilgili testler yeniden çalıştırılarak hataların giderildiği doğrulanır.
- **Hata Önceliklendirme:** Hatalar, sistem üzerindeki etkisine göre önceliklendirilir. Kritik hatalar öncelikli olarak ele alınır ve çözülür.

### 7.3 Test Raporlaması

- **Test Sonuçlarının Raporlanması:** Her bir testin sonucu (başarılı, başarısız) detaylı olarak raporlanır. Rapor, her bir testin amacı, kullanılan senaryolar, sonuçları ve varsa tespit edilen hataları içerir.
- **Test Kanıtlarının Sağlanması:** Raporlar, ekran görüntüleri ve videolar gibi kanıtlarla desteklenir. Bu kanıtlar, test sürecinin şeffaflığını ve doğruluğunu artırır.
- **Genel Değerlendirme:** Test sürecinin genel değerlendirilmesi, başarı oranı, çözülmüş hatalar ve sistemin genel durumu hakkında bilgi verir.
- **Öneriler ve Sonuçlar:** Sistemin iyileştirilmesi için öneriler sunulur. Bu öneriler, gelecekteki geliştirme süreçlerine rehberlik eder.