

# Masraf Yönetim Uygulaması - Yazılım Gereksinimleri Belgesi

**Versiyon:** 1.0

**Tarih:** 13 Ekim 2023

## 1. Giriş

### 1.1 Amaç

Bu belge, masraf yönetim uygulamasının geliştirilmesi için gereksinimleri tanımlar. Uygulamanın, minimumda bir firma çalışanının web tabanlı bir yazılıma masraf fişleri girmesi, yöneticisinin onaylama veya reddetme yetkisine sahip olması, onaylanan masraf fişlerinin veritabanına saklanması ve ödeme hazırlığına getirilmesi, ayrıca ödenecek masraf fişlerinin detaylarının ve ödenecek miktarlarının görüntülenmesi gibi temel işlevselliği sağlamasını amaçlar.

### 1.2 Kapsam

Bu SRS dokümanı, uygulamanın kullanıcıları, sistem davranışları ve teknik gereksinimlerine yönelik detaylı bilgi sağlar. Ayrıca, sistem performansı, güvenlik gereksinimleri ve kullanıcı dostu arayüz tasarımı gibi konulara da değinilir.

## 2. Gereksinimler

### 2.1 Kullanıcı Gereksinimleri

#### 2.1.1 Çalışanlar

- Çalışanlar, uygulamaya giriş yapabilmelidir.
- Masraf fişleri girebilmelidir.
- Girilen masraf fişlerini düzenleyebilmelidir.
- Onay bekleyen masraf fişlerini görüntüleyebilmelidir.

#### 2.1.2 Yöneticiler

- Yöneticiler, uygulamaya giriş yapabilmelidir.
- Çalışanların girdiği masraf fişlerini onaylayabilmelidir.

- Onaylanan masraf fişlerini reddedebilmelidir.
- Onaylanan masraf fişlerini raporlayabilmelidir.

## 2.2 Sistem Gereksinimleri

- Web tabanlı bir platformda çalışmalıdır.
- Kullanıcı kimlik doğrulama ve yetkilendirme sağlamalıdır.
- Veritabanında masraf fişleri ve kullanıcı bilgileri saklanmalıdır.
- Raporlama işlevselliği sunmalıdır.

## 3. Genel Tanım

### 3.1 Ürün Perspektifi

Masraf yönetim uygulaması, çalışanların masraf fişlerini oluşturmaya, yöneticilerin bunları onaylamasına veya reddetmesine, verilerin saklanmasına ve ödeme sürecinin yönetilmesine olanak tanır. Bu uygulama, bir web tabanlı yazılım olarak sunulacaktır.

### 3.2 Kullanıcılar

Bu uygulamayı kullanacak temel kullanıcılar iki kategoriye ayrılabilir:

- **Çalışanlar:** Masraf fişlerini girebilecek ve izleyebilecek.
- **Yöneticiler:** Çalışanların girdiği masraf fişlerini onaylayabilecek veya reddedebilecek.

### 3.3 Teknik Gereksinimler

- Uygulama, web tabanlı bir arayüze sahip olmalıdır ve tarayıcılar aracılığıyla erişilebilmelidir.
- Uygulama, güvenli bir kimlik doğrulama mekanizması kullanmalıdır.
- Veritabanı sistemi, kullanıcı bilgilerini ve masraf fişi verilerini güvenli bir şekilde depolamalıdır.
- Uygulama, yüksek kullanılabilirlik ve performans sağlamak için gerektiğinde ölçeklenebilir olmalıdır.
- Tüm iletişim HTTPS protokolü ile şifrelenmelidir.
- Uygulama, kullanıcıların masraf fişlerini düzenleyebilmesi için kullanıcı dostu bir arayüze sahip olmalıdır.

### 3.4 Güvenlik Gereksinimleri

- Kullanıcıların kişisel bilgileri ve masraf verileri, veri sızıntısı ve yetkisiz erişim önlemek için güçlü bir şekilde şifrelenmelidir.
- Oturum süreleri otomatik olarak sonlandırılmalı ve kullanıcıların oturum açıkken uzun süreli inaktivite durumunda otomatik olarak oturumları kapatılmalıdır.

### 3.5 Yetkilendirme ve Erişim Kontrolü

- Kullanıcılar, rolleri ve yetkilendirmeleri doğrultusunda erişim haklarına sahip olmalıdır. Örneğin, yöneticilerin masraf fişlerini onaylama yetkisi olmalıdır, ancak diğer kullanıcılar için bu yetki bulunmamalıdır.

### 3.6 Uyarılar ve Bildirimler

- Kullanıcılar, önemli olaylar (masraf fişi onaylama, reddetme, vb.) hakkında anlık bildirimler alabilmelidir.

### 3.7 İzleme ve Günlükleme

- Tüm kullanıcı etkileşimleri ve sistem olayları günlüklenmeli ve izlenmelidir. Bu kayıtlar, denetim ve hata ayıklama amacıyla kullanılmalıdır.

## 4. Fonksiyonel Gereksinimler

### 4.1 Kullanıcı Girişi

- Kullanıcılar, kullanıcı adı ve şifre ile sisteme giriş yapabilecekler.
- Yeni kullanıcılar, kayıt işlemi yaparak sisteme erişim sağlayabilecekler.

### 4.2 Masraf Fişi Oluşturma

- Çalışanlar, masraf fişleri oluşturabilecekler.
- Her masraf fişi için aşağıdaki bilgiler gerekli olacak:
  - Masraf türü
  - Tutar
  - Tarih
  - Açıklama

- Ek olarak, kullanıcılar belgeleri sisteme yükleyebilecekler.

#### **4.3 Masraf Fişi Onaylama ve Reddetme**

- Yöneticiler, çalışanların girdiği masraf fişlerini inceleyip onaylayabilecek veya reddedebilecekler.
- Onaylanan masraf fişleri veritabanında saklanacak ve ödeme sürecine dahil edilecektir.

#### **4.4 Veritabanı Yönetimi**

- Tüm masraf fişleri, kullanıcı bilgileri ve onay süreçleri, güvenli bir şekilde veritabanında saklanacak.
- Verilerin yedeklemesi ve güvenliği sağlanacaktır.

#### **4.5 Ödeme Hazırlığı**

- Onaylanan masraf fişleri, ödeme için hazırlanacak.
- Ödeme talimatları, belirli bir döneme veya kullanıcılara göre düzenlenebilecek.

#### **4.6 Rapor Oluşturma**

- Kullanıcılar, ödeme yapılacak masraf fişlerinin detaylarını ve ödenecek miktarlarını gösteren raporlar oluşturabilecekler.

### **5. Performans Gereksinimleri**

#### **5.1 Kullanıcı Dostu Arayüz**

- Uygulama, kullanıcı dostu bir web arayüzüne sahip olacak.
- Arayüz, kullanıcıların kolayca gezinebilmesini sağlayacak.

#### **5.2 Güvenlik**

- Kullanıcı verileri güvenli bir şekilde saklanacak ve iletilmeleri şifrelenmiş olacak.
- Kullanıcı yetkilendirmesi ve erişim denetimleri uygulanacak.

### **6. Teknik Gereksinimler**

#### **6.1 Teknolojik Altyapı**

- Uygulama web tabanlı olacak ve modern web tarayıcılarında çalışabilecektir.
- Veritabanı yönetimi için güvenilir bir sistem kullanılacaktır.

## 6.2 Dil ve Platform

- Uygulama, yaygın olarak kullanılan programlama dilleri ve platformlar kullanılarak geliştirilecektir.

## 6.3 Kabul Kriterleri

Bu projenin başarıyla tamamlandığı kabul edilecektir:

- Kullanıcıların kayıt olabilmesi ve giriş yapabilmesi.
- Kullanıcıların masraf fişleri oluşturabilmesi ve düzenleyebilmesi.
- Yöneticilerin masraf fişlerini onaylayabilmesi veya reddedebilmesi.
- Verilerin güvenli bir şekilde saklanabilmesi.
- Ödeme talimatlarının oluşturulabilmesi ve ödeme yapılacak fişlerin raporlanabilmesi.

## 6.4 Referanslar

[İlgili Referanslar ve Dokümantasyonlar]

## 7. Geliştirme ve Dağıtım

### 7.1 Geliştirme Dili ve Teknolojileri

Uygulama web tabanlı olacak ve yaygın web teknolojileri kullanılacaktır (örneğin, HTML, CSS, JavaScript ve Ruby on Rails).

### 7.2 Geliştirme Ortamı

Uygulama geliştirme ve test süreçleri için uygun bir geliştirme ortamı sağlanmalıdır.

### 7.3 Dağıtım

Uygulama, firmaların kendi sunucularına veya bulut altyapısına kurulabilir. Dağıtım işlemi, kullanıcıların gereksinimlerine göre belirlenmelidir.

### 7.4 Güvenlik

Uygulama, kullanıcıların verilerini ve gizliliğini korumak için gerekli güvenlik önlemlerini almalıdır. Kullanıcı oturumları güvenli bir şekilde yönetilmelidir. Veritabanı erişimi, yetkilendirilmiş kullanıcılarla sınırlanmalıdır.

## **8. İş Akışları**

### **8.1 Kullanıcı Girişi ve Kayıt İşlemi**

- Kullanıcılar, ana sayfada kullanıcı adı ve şifreleri ile giriş yapabilirler.
- Yeni kullanıcılar kayıt olmak için "Kayıt Ol" seçeneğini seçebilirler. Kayıt işlemi, temel kullanıcı bilgileri ve erişim yetkilerini içerecektir.

### **8.2 Masraf Fişi Oluşturma**

- Kullanıcılar, giriş yaptıktan sonra "Masraf Fişi Ekle" seçeneğini seçerek yeni masraf fişleri oluşturabilirler.
- Masraf fişi oluştururken, gerekli bilgileri girip belgeleri ekleyebilirler.

### **8.3 Masraf Fişi Onaylama ve Reddetme**

- Yöneticiler, giriş yaptıktan sonra "Onaylama" veya "Reddetme" seçeneğini seçerek çalışanların masraf fişlerini inceleyebilirler.
- Yöneticiler, masraf fişlerini onayladıklarında, bu fişler veritabanında saklanır ve ödeme hazırlığına alınır.

### **8.4 Veritabanı Yönetimi**

- Veritabanı, kullanıcılar, masraf fişleri ve onay süreçleri gibi verileri güvenli bir şekilde saklar.
- Günlük yedeklemeler ve veri güvenliği önlemleri alınır.

### **8.5 Ödeme Hazırlığı**

- Onaylanan masraf fişleri, ödeme talimatlarına dahil edilir ve belirli bir döneme veya kullanıcılara göre düzenlenir.

### **8.6 Rapor Oluşturma**

- Kullanıcılar, "Rapor Oluştur" seçeneği ile ödeme yapılacak masraf fişlerinin detaylarını ve ödenecek miktarlarını görüntüleyebilirler.

## 9. Sistem Sınırlamaları

- Bu uygulama, yalnızca web tabanlı bir yazılım olarak sunulacaktır.
- Ödeme işlemleri bu uygulamanın kapsamı dışındadır ve başka bir sistemle entegre edilmelidir.

## 10. Değişiklik Kontrolü

Bu SRS dokümanı, proje sürecinde gereksinimlerin değişmesi veya güncellenmesi gerektiğinde güncellenebilir. Herhangi bir değişiklik, proje ekibi ve ilgili paydaşlar tarafından onaylanmalıdır.

## 11. Onay

Bu SRS dokümanı, projenin ilgili tüm taraflar tarafından onaylanmalıdır. İlgili paydaşlar ve proje ekibi, gereksinimlerin tam ve doğru bir şekilde anlaşıldığına ve kabul edildiğine dair onaylarını verecektir.

### 11.1 Ekler

[Herhangi bir ek bilgi veya belge]

## 12. Gereksinimlerin İzlenmesi ve Güncellenmesi

Proje süresince, gereksinimlerin izlenmesi ve güncellenmesi büyük önem taşır. Bu süreç, gereksinim değişikliklerinin ve güncellemelerinin yönetimini içerir. Değişikliklerin yönetimi ve güncellemelerin belgelenmesi şu adımları içerir:

### 12.1 Gereksinim Değişiklikleri

- Herhangi bir gereksinim değişikliği veya ekleme önerisi, proje ekibine veya proje yöneticisine iletilmelidir.
- Değişiklik talepleri, projenin etkileri ve faydaları açısından değerlendirilmelidir.
- Değişiklik talepleri kabul edildiğinde, dokümantasyon güncellenmelidir.

### 12.2 Dokümantasyon Güncellemesi

- SRS dokümanı, herhangi bir gereksinim değişikliğine veya güncellemesine uygun bir şekilde güncellenmelidir.
- Her güncelleme veya değişiklik, sürüm numarası ve tarih ile işaretlenmelidir.

### 12.3 Onay ve İletişim

- Gereksinim değişiklikleri, ilgili paydaşlar ve proje ekibi tarafından onaylanmalıdır.
- İlgili tarafların güncellemeler hakkında bilgilendirilmesi sağlanmalıdır.

## 13. Riskler ve İlerleme İzleme

### 13.1 Riskler

Proje süresince, potansiyel riskler ve sorunlar göz önünde bulundurulmalıdır. Özellikle veri güvenliği, sistem performansı ve kullanıcıların ihtiyaçlarına yönelik riskler özel öneme sahiptir.

Risklerin belirlenmesi, değerlendirilmesi ve yönetilmesi için bir risk yönetimi planı oluşturulmalıdır.

### 13.2 İlerleme İzleme

Proje ilerlemesi düzenli olarak izlenmelidir. Proje yöneticisi, herhangi bir aksaklık veya gecikme durumunda hızlı bir şekilde müdahale etmelidir.

İlerleme raporları, projenin başarıyla tamamlanmasını sağlamak için kullanılmalıdır.

## 14. Proje Zaman Çizelgesi

Proje zaman çizelgesi, projenin başlangıç ve bitiş tarihleri ile projenin farklı aşamalarının ve teslimatlarının ne zaman tamamlanacağını göstermektedir. Ayrıca, ilerleme izleme ve değişiklik yönetimi için önemli zamanlamaları içermektedir.

### 14.1 Başlangıç ve Bitiş Tarihleri

Proje Başlangıç Tarihi: 2023-01-01, Proje Bitiş Tarihi: 2023-06-30

### 14.2 Proje Aşamaları ve Teslimatlar

#### Aşama 1: Gereksinim Analizi

- Başlangıç Tarihi: 2023-01-01
- Bitiş Tarihi: 2023-01-31
- Ana Teslimatlar: Gereksinim Belgesi

#### Aşama 2: Tasarım ve Planlama

- Başlangıç Tarihi: 2023-02-01



- Bitiş Tarihi: 2023-02-28
- Ana Teslimatlar: Tasarım Dokümanı, Proje Planı

### **Aşama 3: Geliştirme**

- Başlangıç Tarihi: 2023-03-01
- Bitiş Tarihi: 2023-03-31
- Ana Teslimatlar: Beta Sürüm

### **Aşama 4: Test ve Kalite Güvence**

- Başlangıç Tarihi: 2023-04-01
- Bitiş Tarihi: 2023-04-30
- Ana Teslimatlar: Test Raporu

### **Aşama 5: Uygulama ve Canlı Yayın**

- Başlangıç Tarihi: 2023-05-01
- Bitiş Tarihi: 2023-06-14
- Ana Teslimatlar: Canlı Ortama Taşıma, Eğitim

## **14.3 İlerleme İzleme ve Raporlar**

- Haftalık İlerleme Raporları: Haftalık olarak, her Çarşamba günü sunulacak.
- İlerleme Toplantıları: Haftalık olarak, her Perşembe günü saat 14:00'te düzenlenecek.

## **14.4 Değişiklik Yönetimi**

- Değişiklik İstekleri Değerlendirme: Değişiklik istekleri, isteğin geldiği tarihten itibaren en fazla 3 iş günü içinde değerlendirilecek.
- Değişiklik Kabul ve Uygulama: Kabul edilen değişiklikler, uygun aşamalara ve zamanlamalara göre uygulanacaktır.

Bu zaman çizelgesi, projenin zaman yönetimi ve ilerlemesini düzenlemek için kullanılacaktır. Tüm ekip üyeleri ve paydaşlar, bu zaman çizelgesini dikkatlice takip etmelidir.

## **14. Dil Desteği**

- Uygulama, kullanıcıların tercih ettikleri dilde görüntülenebilmesi için çoklu dil desteği sunmalıdır.

#### **15. Teknolojik Altyapı**

- Uygulama, modern web teknolojileri (örneğin, HTML5, CSS3, JavaScript) ve uygun programlama dilleri (örneğin, Python, Java, veya C#) kullanılarak geliştirilmelidir.
- Veritabanı yönetimi için güvenilir ve performanslı bir veritabanı sistemi kullanılmalıdır (örneğin, PostgreSQL, MySQL, veya Microsoft SQL Server).

#### **16. Performans ve Ölçeklenebilirlik**

- Uygulama, yüksek kullanılabilirlik sağlamalı ve büyüme durumunda kolayca ölçeklenebilir olmalıdır.
- Uygulama, belirli bir işlem süresi hedefine ulaşmak için yeterli sistem kaynaklarını kullanmalıdır.

#### **17. Sürüm Kontrolü**

- Yazılım sürüm kontrolü için bir sistem kullanılmalıdır (örneğin, Git), böylece kod değişiklikleri takip edilebilir ve geri alınabilir olmalıdır.

#### **18. Lisanslama**

- Tüm kullanılan üçüncü taraf bileşenlerin ve kütüphanelerin lisanslama koşulları incelenmeli ve uygun şekilde takip edilmelidir.

#### **19. İş Sürekliliği ve Yedekleme**

- Uygulama ve veritabanı sistemi için iş sürekliliği planları ve yedekleme süreçleri oluşturulmalıdır.

#### **20. Test ve Kalite Güvencesi**

- Uygulama, ayrıntılı test planları ve senaryoları kullanarak kapsamlı bir test sürecinden geçmelidir. Bu testler, uygulamanın işlevselliği, güvenliği ve performansını değerlendirmelidir.

#### **21. Bakım ve Güncelleme**

- Uygulamanın bakım ve güncelleme süreçleri düzenli olarak izlenmeli ve güncellemeler düzgün bir şekilde uygulanmalıdır.

## **22. Yasa ve Yönetmeliklere Uygunluk**

- Uygulama, ilgili yasa ve yönetmeliklere (örneğin, kişisel verilerin korunması, finansal raporlama vb.) tam olarak uygun olmalıdır.

## **23. Belgelendirme**

- Kullanıcılar ve yöneticiler için kapsamlı bir kullanım kılavuzu veya belgesi sağlanmalıdır.

## **24. Zaman Çizelgesi**

- Uygulama geliştirme sürecinin zaman çizelgesi belirlenmeli ve projenin ilerleyişi düzenli olarak izlenmelidir.

## **25. Proje Sorumluları ve İletişim**

- Proje yöneticileri, geliştiriciler, test ekipleri ve paydaşlar arasında düzenli iletişim kurmak için bir mekanizma oluşturmalıdır.

## **26. Onay ve Değişiklik İstekleri**

- Bu SRS dökümanı, tüm paydaşlar tarafından gözden geçirilmeli ve onaylanmalıdır. Herhangi bir gereksinim değişikliği isteği, değişiklik sürecine tabi tutulmalıdır.

## **27. Kabul Kriterleri**

- Uygulamanın kullanıcılar tarafından kabul edilmiş sayılabilmesi için aşağıdaki kabul kriterleri sağlanmalıdır:
- Tüm işlevsel gereksinimler, test edilip doğrulandığında çalışır durumda olmalıdır.
- Uygulama, güvenli bir şekilde çalışmalı ve kullanıcı bilgilerini korumalıdır.
- Uygulama performansı, belirlenen hedefler doğrultusunda olmalıdır.
- Tüm belgeler ve kullanıcı eğitim materyalleri hazır ve kullanılabilir durumda olmalıdır.

## **28. Geliştirme Süreci**

- Geliştirme süreci, aşağıdaki adımları içermelidir:
- Gereksinim analizi ve tasarım: İhtiyaçlar belirlenmeli ve tasarım belgesi oluşturulmalıdır.
- Uygulama geliştirme: Uygulama, belirlenen gereksinimlere uygun olarak kodlanmalıdır.

- Test ve kalite güvencesi: Tüm işlevsel ve teknik gereksinimler doğrulanmalı ve test edilmelidir.
- Dağıtım ve kullanıma alma: Uygulama, kullanıcıların erişimine açılmalıdır.

### 29. Değişiklik Kontrolü

- Gereksinimlerin değişiklikleri, değişiklik isteği formu kullanılarak belgelendirilmeli ve değişiklik sürecine tabi tutulmalıdır. Değişiklikler, proje ekibi tarafından dikkatle değerlendirilmeli ve onaylanmalıdır.

### 30. İletişim ve Paydaşlar

- Proje ekibi, paydaşlarla düzenli iletişim kurmalı ve herhangi bir önemli değişiklik veya güncelleme hakkında bilgilendirme yapmalıdır. İletişim kanalları ve iletişim planı oluşturulmalıdır.

### 31. Risksizlik ve Acil Durum Planı

- Proje ekibi, potansiyel riskleri tanımalı ve bunlarla başa çıkabilmek için bir risk yönetimi planı oluşturmalıdır. Ayrıca, beklenmeyen acil durumlar için bir acil durum planı hazırlanmalıdır.

### 32. Bitirme Kriterleri

- Proje bitirme kriterleri ve sonlandırma süreci tanımlanmalıdır. Bu kriterler, projenin tamamlandığını belirlemek için kullanılacaktır.

### 33. Test Stratejisi ve Planı

- Detaylı bir test stratejisi ve test planı oluşturun. Bu plan, işlevsel ve teknik gereksinimlerin doğrulandığından ve uygulamanın kararlılığının test edildiğinden emin olmanızı sağlar.

### 34. Performans İzleme ve İyileştirme

Uygulamanın performansını düzenli olarak izleyin ve geliştirme fırsatları belirleyin. Bu, uygulamanın verimliliğini artırmanıza yardımcı olur.

### 35. Uygulama ve Geliştirme

Bu aşamada, yazılım geliştirme ekibi, SRS belgesine dayanarak uygulamayı tasarlar ve geliştirir. Bu süreçte, belgede tanımlanan tüm gereksinimlere uygun bir şekilde yazılım kodu oluşturulmalıdır. İşte bu sürecin bazı adımları:

Gereksinim analizi: SRS belgesi üzerindeki gereksinimler detaylı bir şekilde incelenir ve yazılımın nasıl geliştirileceği planlanır.

Tasarım: Yazılımın mimarisi tasarlanır, kullanıcı arayüzü oluşturulur ve veritabanı şeması belirlenir.

Geliştirme: Kodlama süreci başlar. Yazılım geliştirme ekibi, tasarlanan yapının temelinde yazılımı inşa eder.

Testler: Yazılım, belgedeki gereksinimlere uygunluğu doğrulamak için kapsamlı testlerden geçer.

### **36. Kabul Testleri**

Yazılımın geliştirilmesi tamamlandığında, proje ekibi ve paydaşlar tarafından kabul testleri yapılmalıdır. Bu testler, yazılımın tüm gereksinimleri karşıladığını doğrulamak için gerçekleştirilir.

### **37. Canlı Ortama Geçiş**

Yazılım, canlı bir ortama geçirilmeden önce son kez test edilmeli ve gerekli düzeltmeler yapılmalıdır. Canlı ortama geçiş planı, veritabanı geçişi, sunucu konfigürasyonu ve diğer ilgili işlemleri içermelidir.

### **38. Sürdürülebilirlik ve Bakım**

Yazılım projesi tamamlandığında, sürdürülebilirlik ve bakım süreçleri devreye girmelidir. Bu aşamada, aşağıdaki görevler önemlidir:

- **Düzenli bakım:** Yazılımın işleyişini kontrol etmek ve herhangi bir hata veya sorun tespit edildiğinde bunları düzeltmek için düzenli bakım yapılmalıdır.
- **Güvenlik güncellemeleri:** Yazılımın güvenliği için gerektiğinde güncellemeler yayımlanmalı ve uygulanmalıdır.
- **Performans izleme:** Yazılımın performansı izlenmeli ve gerektiğinde iyileştirmeler yapılmalıdır.
- **Kullanıcı desteği:** Kullanıcıların sorularını yanıtlamak ve yardım etmek için bir destek sistemi kurulmalıdır.
- 

### **39. Geri Bildirim ve İyileştirmeler**

Kullanıcıların geri bildirimlerini toplamak ve bu geri bildirimlere dayalı olarak yazılımı sürekli olarak iyileştirmek önemlidir. Kullanıcılar ve işletme sahipleri ile düzenli olarak iletişim kurulmalı ve yazılımın kullanımı

### **40. Raporlama ve Değerlendirme**

Proje sonunda bir raporlama ve değerlendirme aşaması yapılmalıdır. Bu aşama, proje performansının analiz edilmesini, proje süreçlerinin ve sonuçlarının gözden geçirilmesini içerir. Bu

değerlendirme, gelecekteki projeler için öğrenilen derslerin belirlenmesine ve iyileştirmelerin yapılmasına yardımcı olacaktır.

#### **41. Arşivleme ve Dokümantasyon**

Proje belgeleri, dokümantasyonlar ve kaynak kodlar düzenlenmeli ve arşivlenmelidir. Bu belgeler, gelecekteki ihtiyaçlar veya referans için saklanmalıdır.

hakkındaki geri bildirimler toplanmalıdır.

#### **42. Proje Sonu**

Yazılım projesi başarıyla tamamlandığında, tüm projeyi kapatmak ve sonuçlandırmak önemlidir. Proje sonu belgesi oluşturulmalı ve tüm ilgili taraflar için iletişim kurulmalıdır.

#### **43. Sonuç**

Yazılım projesinin başarılı bir şekilde sonlandırılması ve sürdürülmesi, projenin hedeflerine ulaşmasını sağlar. Proje sonuçlarının düzenli olarak gözden geçirilmesi ve iyileştirilmesi, yazılımın uzun vadeli başarısını sürdürmek için önemlidir.

SRS belgesi, projenin başlangıç noktası olarak hizmet eder ve projenin tüm aşamalarını yönlendirmek için kullanılır. İşbirliği, iletişim ve proje yönetimi, projenin başarılı bir şekilde tamamlanmasına katkıda bulunur. Projeyi sonlandırmak ve sürdürmek, yazılımın kullanım ömrü boyunca işletme ihtiyaçlarına uygunluğunu ve etkinliğini sürdürmek için kritik öneme sahiptir. Projelerin başarılı bir şekilde tamamlanması, işletmeler ve müşteriler için değer sağlar.

#### **44. Onay ve Kabul**

Bu Yazılım Gereksinimleri Belgesi (SRS) dokümanı, proje yöneticisi, proje ekibi ve ilgili paydaşlar tarafından dikkatle incelenmeli ve onaylanmalıdır. Gereksinimlerin tam ve doğru bir şekilde anlaşıldığından ve kabul edildiğinden emin olunmalıdır.

Bu SRS dokümanı, projenin temel taşıdır ve geliştirme ekibi için yol haritası sağlar. Bu gereksinimlerin karşılanması, projenin başarılı bir şekilde tamamlanmasını sağlamak için kritik bir adımdır.

izlenmesi ve dokümantasyonun güncellenmesi önemlidir. Projeyi yöneten ve projede çalışan tüm taraflar, bu dokümanı dikkatle incelemeli ve onaylamalıdır.

#### **45. Ekler**

- Ek 1: Veritabanı Şeması
- Ek 2: Kullanıcı Kılavuzu
- Ek 3: Geliştirici Belgesi

Bu ekler, SRS belgesine ilişkin daha fazla ayrıntı sağlamak için kullanılabilir. Veritabanı şeması, uygulamanın veri tabanı yapısını ayrıntılı olarak gösterirken, kullanıcı kılavuzu ve geliştirici belgesi, kullanıcıların ve geliştiricilerin uygulamayı daha iyi anlamalarına yardımcı olur.

#### **46. Onaylar**

SRS belgesi, ilgili paydaşlar ve proje ekibi tarafından onaylanmalıdır. Onaylandıktan sonra, belge değişiklikler sırasında izlenmelidir.

İlgili Paydaşlar:

- [Paydaş İsmi 1]
- [Paydaş İsmi 2]

Proje Ekibi:

- [Proje Ekibi Üyesi 1]
- [Proje Ekibi Üyesi 2]

Onay Tarihi: [Onay Tarihi]

#### **47. İletişim Bilgileri**

Bu bölümde, SRS belgesi ile ilgili iletişim bilgileri sağlanmalıdır. Bu, belgeyi oluşturan veya onaylayan kişilerin iletişim bilgilerini içerebilir.

- Proje Yöneticisi: [İsim, E-posta]
- Baş Geliştirici: [İsim, E-posta]

#### **48. Referanslar**

- Bu SRS dökümanının hazırlanmasında kullanılan tüm kaynaklar ve referanslar, dökümanın sonunda listelenmelidir.