



# <Proje Adı> Tasarım Belgesi

## Yazarlar:

- <isim 1>
- <isim 2>
- <isim 3>
- <isim 4>

Grup: 0

*Bu belgenin amacı, projeniz için yazılım tasarım belgesini yazarken size bir kılavuz sağlamaktır.*

*Hatırlanması gereken noktalar:*

- İçerik önemlidir, hacim değil. **Başka bir ekip sadece bu belgeden yola çıkarak bu sistemi geliştirebilmelidir.**
- Ayrıntılara dikkat edin.
- Eksiksizlik ve tutarlılık ödüllendirilecektir.

- *Okunabilirlik önemlidir.*

Bu sayfa kasıtlı olarak boş bırakılmıştır.

## Belge Revizyon Geçmişi

Tarih	Versiyon	Açıklama	Yazar
10/25/2012	0.0	İlk taslak	
10/25/2012	0.1	<ex: Sınıf Diyagramı Eklendi>	John Q. Doe

Bu sayfa kasıtlı olarak boş bırakılmıştır.

## İçindekiler

<b>1</b>	<b>Giriş</b> .....	<b>6</b>
1.1	Amaç.....	6
1.2	Sisteme Genel Bakış .....	6
1.3	Tasarım Hedefleri.....	6
1.4	Referanslar .....	6
1.5	Tanımlar, Kısaltmalar ve Kısaltmalar .....	6
<b>2</b>	<b>Tasarıma Genel Bakış</b> .....	<b>7</b>
2.1	Giriş .....	7
2.2	Çevreye Genel Bakış .....	7
2.3	Sistem Mimarisi .....	7
2.3.1	Yakıt seviyesi kontrolörünün üst düzey sistem yapısı .....	7
2.3.2	Seviye İzleme Alt Sistemi.....	7
2.3.3	Yakıt Alt Sistemi .....	8
2.4	Kısıtlamalar ve Varsayımlar .....	8
<b>3</b>	<b>Arayüzler ve Veri Depoları</b> .....	<b>9</b>
3.1	Sistem Arayüzleri.....	9
3.1.1	Basınç Sensörü Arayüzü .....	9
3.1.2	Pompa Valfi Kontrolör Arayüzü .....	9
3.2	Veri Depoları .....	9
<b>4</b>	<b>Yapısal Tasarım</b> .....	<b>10</b>
4.1	Tasarım Tartışması ve Gerekçesi .....	10
4.2	Sınıf Diyagramı.....	11
4.2	Sınıf Açıklamaları.....	11
4.3	Yakıt Alt Sistemindeki Sınıflar .....	11
4.3.1	Sınıf Tank.....	11
<b>5</b>	<b>Dinamik Model</b> .....	<b>12</b>
5.1	Senaryolar .....	12
<b>6</b>	<b>Tamamlayıcı Dokümantasyon</b> .....	<b>13</b>

## 1 Giriş

Bu bölüm, hedef kitle, soruna giriş ve proje tasarımının ayrıntılı bir görünümü dahil olmak üzere bu belgenin amacını ele almaktadır. Tartışmada, birkaç ayrıntılı diyagram da dahil olmak üzere nihai sistemin tasarımı ayrıntılı olarak açıklanacaktır.

### 1.1 Amaç

Belgeye kısa bir giriş yapınız. Belgenin amacı nedir? Düzeni nasıldır?

### 1.2 Sisteme Genel Bakış

Sistemin amacının, hedef kullanıcılarının ve sistemin kullanılacağı ortamın kısa bir tanımını sağlayın. Bu, temel olarak, varsa revizyonlarla birlikte gereksinimler belgesinin başlangıç kısmından alınmıştır.

### 1.3 Tasarım Hedefleri

Bu bölümde genel sistem tasarım hedeflerini tartışın. Tasarımın hangi işlevleri kapsamaya çalıştığından ve neleri kapsamadığından genel olarak bahsedin. Ayrıca, tasarımın ele aldığı performans ve kullanılabilirlik gibi işlevsel olmayan gereksinimlere de atıfta bulunun. Gereksinim spesifikasyon belgesi bu bölümün yazılmasında faydalı olabilir. Gereksinim spesifikasyonundaki işlevsel ve işlevsel olmayan gereksinim tanımlarına bir göz atın.

Bu bölüm, tasarımın genel kapsamını ve bağlamını sağlamalıdır. Bu bölümü okuduktan sonra okuyucu *hangi* özelliklerin ve işlevlerin ele alındığını anlayabilmelidir. İlerleyen bölümlerde, tasarımınızda bunun *nasıl* başarıldığını açıklayacaksınız.

### 1.4 Referanslar

Diğer belgelere referanslar sağlayın.

*İpucu:* En azından gereksinimler belgesine başvurmalsınız!

### 1.5 Tanımlar, Kısaltmalar ve Kısaltmalar

*Belgede kullanılan çeşitli terimlerin, kısaltmaların veya akronimlerin tanımı.*

*Not: Asistanlarınız "bununla ne demek istiyorsun...?" diye sorarsa, belki de buraya kısa bir tanım eklemeyi düşünmelisiniz.*

*Bazı terimler gereksinim belgesinde zaten tanımlanmışsa, okuyucuyu gereksinim belgesinin sözlüğüne yönlendirebilirsiniz.*

## 2 Tasarıma Genel Bakış

### 2.1 Giriş

Bu bölümde benimsediğiniz tasarım yaklaşımı (örn. nesne odaklı tasarım veya yapılandırılmış tasarım), sistemin önerilen mimarisi (örn. istemci-sunucu) ve kullanılan ilgili teknik ve araçlar (örn. Microsoft Visio, Dia, ArgoUML, vb.) vurgulanmalıdır.

### 2.2 Çevreye Genel Bakış

Bu bölüm sistemin çalıştırılacağı ortamı tanımlamalıdır. Bu, ortamın diyagramlarını ve açıklamalarını içermelidir. Örneğin, bir web uygulamasını tanımlıyorsak, bu bir ağ diyagramı veya topolojisi içerecektir. Bir kütüphane geliştiriyorsak, bizi neyin çağırabileceğini tanımlamalıyız. Bunu etkili bir şekilde yapabilmek için sisteminizin kapsamını önceden belirtmiş olmanız gerekir.

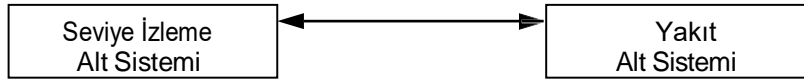
Bu uygulamanın nerede bulunacağını ve nasıl yürütüleceğini belirttiğinizden emin olun.

### 2.3 Sistem Mimarisi

Sistem mimarisinin üst düzey bir tanımını sağlayın. Başlıca bileşenleri ve bunların etkileşimini göstermek için birkaç blok diyagram kullanın. Diyagramlarınızda kullanılan kuralları ve gösterimleri açıklamayı unutmayın.

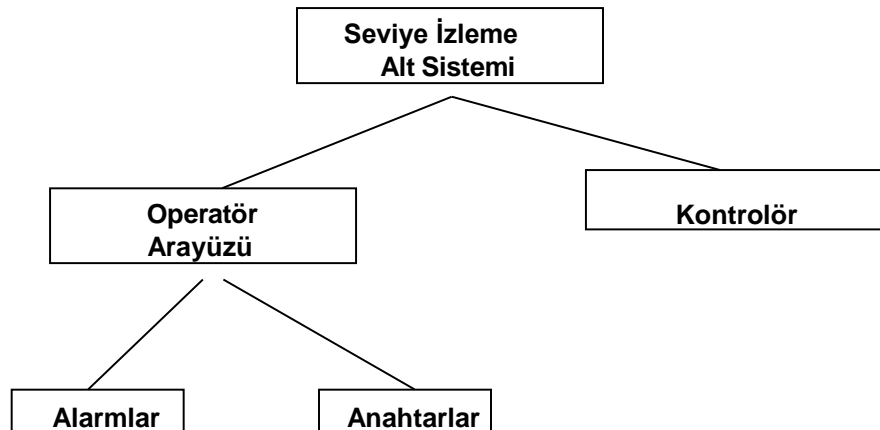
**Aşağıda örnek olarak verilmiştir:**

#### 2.3.1 Yakıt seviyesi kontrolörünün üst düzey sistem yapısı



Sistem, seviye izleme alt sistemi ve yakıt alt sistemi olmak üzere iki ana bileşenden oluşmaktadır. Sistemler arasındaki etkileşim çift yönlüdür. Mevcut sensör ölçümleri yakıt alt sisteminden seviye izleme alt sistemine gönderilir ve mevcut yakıt seviyesine bağlı olarak çeşitli aktüatörlere talimatlar geri gönderilir.

#### 2.3.2 Seviye İzleme Alt Sistemi





*Seviye İzleme Alt Sistemi Operatör arayüzü ve Kontrolörden oluşur. Operatör arayüzü de Alarmlar ve Anahtarlardan oluşur.*

### 2.3.3 Yakıt Alt Sistemi

*Başka bir diyagram ve biraz açıklama...*

## 2.4 Kısıtlamalar ve Varsayımlar

*Burada başlıca tasarım kısıtlamalarından bahsedin. Bunlar müşteri tarafından dayatılmış olabilir ve gereksinimler belgesinde bulunabilir. Tasarımınızın bu kısıtlamaları nasıl karşıladığını açıklayın.*

*Sisteminizin diğer harici sistemlerle etkileşime girmesi veya işlevselliğin bir kısmını sağlamak için bazı harici sistemlere bağımlı olması nedeniyle uygulanan kısıtlamalar da olabilir. Bu gibi durumlarda, sisteminizin etkileşimde bulunduğu yazılım türünü (örneğin XYZ veritabanı yazılımı, ABC e-posta yazılımı) ve bunun getirdiği kısıtlamaları (örneğin yalnızca metin tabanlı e-posta mesajlarına izin verilir) açıkça belirtin.*

*Uygulama dilleri ve platformları belirli kısıtlamalar getirebilir. Uygulamanızda kullanacağınız diller veya teknolojiler de dahil olmak üzere bunları burada belirtin.*

Tasarım tercihlerinizin getirdiği kısıtlamalar için, sahip olduğunuz seçeneklerden, ilgili ödünleşimlerden ve neden bu seçimi yaptığınızdan kısaca bahsedin.

1. Yakıt seviyesindeki değişikliklere kontrolörden gelen yanıt 100 milisaniye içinde kontrolöre ulaşmalıdır.

*Bu, öncelik şeması ve kontrol ünitesini çalıştıran temel yürütme yazılımının bir kombinasyonu ile ele alınır. Olaylar yönetici tarafından önceliğe göre ele alınmak üzere programlanır. Yönetici her 25 milisaniyede bir çalışır ve olay kuyruğunda bekleyen en yüksek öncelikli olay için olay-işleyiciyi programlar. Tüm olay-işleyicileri 25 milisaniyelik zaman sınırı içinde çalışır.*

2. Başka bir kısıtlama...

Kısıtlama getiren bir tasarım tercihi örneği:

*E-posta yazılımı seçimi mesajları düz metinle kısıtlar.*

*Temel işlevleri sağlayan kullanımı kolay bir yazılım ya da tam özellikli bir mesajlaşma sistemi seçme seçeneğimiz vardı. İlk seçenek sadece metin tabanlı mesajlar kullanabileceğimiz anlamına gelirken, ikinci seçenek yazılımı kurmak ve desteklemek için çok fazla ek yük anlamına geliyordu. Uygulamamız ağırlıklı olarak metin tabanlı mesajlar gerektirdiği için ilk seçeneği tercih ettik. Gelecekte ikili veriler bazı kodlama şemaları kullanılarak metin tabanlı sistem üzerinden gönderilebilir.*

### 3 Arayüzler ve Veri Depoları

Bu bölümde sisteme giren ve çıkan arayüzlerin yanı sıra sisteminize dahil edeceğiniz veri depoları da açıklanmaktadır.

#### 3.1 Sistem Arayüzleri

*Kullanıcılara ve/veya diğer harici sistemlere sağlanan çeşitli arayüzler burada tanımlanmalıdır. Gereksinimler dokümanınızda kullanıcı arayüzü açıklamalarına yer verdiğinizde, bunlara burada atıfta bulunabilirsiniz. Diğer sistemlere arayüzler sağlıyorsanız, örneğin farklı bir yazılıma veri aktarıyor ve alıyorsanız, bunları burada belirtmelisiniz.*

*JavaDoc'ları buraya dahil etmeyin. Arayüzünüz bir GUI içeriyorsa, arayüzleri düzyazı ve grafik kullanarak açıklamalısınız.*

**Aşağıda örnek olarak verilmiştir:**

##### 3.1.1 Basınç Sensörü Arayüzü

*Bu arayüz yakıt seviyesinin hesaplandığı tankın altındaki sıvı basıncını ölçmek için kullanılır. Sensör basıncı inç kare (psi) başına pound cinsinden ölçer. Kontrolör yazılımından sensör okumasına erişmek için bir donanım arayüzü sağlanmıştır. Verilere yazılım tarafından sabit bir bellek konumundan erişilebilir (bellek eşlemeli I/O kullanılarak).*

##### 3.1.2 Pompa Vana Kontrolör Arayüzü

*Açıklama buraya...*

#### 3.2 Veri Depoları

*SİSTEMİNİZİN BİR PARÇASI OLARAK oluşturacağınız dahili veritabanlarını/veri depolarını - sabit sürücüde kalıcı olarak depolanan tüm verileri - tanımlayın. Bu, sisteminize harici olan veritabanlarını veya diğer veri depolarını İÇERMEMELİDİR. Bu bilgiler önceki bir bölümde yer almalıdır.*

### 4 Yapısal Tasarım

#### 4.1 Tasarım Açıklaması ve Gerekçesi

*Seçtiğiniz tasarımın, sisteminizin nasıl yapılandırıldığını, isteklerin nasıl ele alındığını ve sisteminizdeki iletişim ve kontrol akışını açıklayan yazılı bir açıklaması.*

*Bu bölüm aynı zamanda tasarımda alınan kararlara ilişkin özel gerekçeleri de içeren tasarım gerekçelerini de içermelidir (örneğin, tasarımınızın neden diğerinden daha iyi olabileceği veya neden alt sistemlerinizden birinde belirli bir tasarım modelini uygulamayı seçtiğiniz gibi).*

## 4.2 Sınıf Diyagramı

Sistem için sınıf diyagramını sağlayın. Sınıf diyagramı çok büyükse, diyagramı bazı makul kriterler kullanarak bölümlere ayırın. Örneğin, istemci tarafı ve sunucu tarafı nesne modellerini ayrı diyagramlar olarak sağlayabilirsiniz.

*Bir sınıf diyagramına ne girmelidir?*

Tüm sınıflar burada bir yer bulmalıdır. Sınıflar arasındaki tüm ilişkiler tanımlanmalı ve ilişkiler doğru kardinalite ile dekore edilmelidir. Toplama ve kalıtım ilişkileri tanımlanmalıdır. **Tasarımınız için gereksellerinizi içeren kısa bir açıklama her diyagrama eşlik etmelidir.** Belgedeki şekillere atıfta bulunmanıza yardımcı olması için şekil numaralarını eklediğinizden emin olun.

Unutmayın: Sisteminiz için doğru sınıf diyagramı üzerinde anlaşmaya varmadan önce birkaç kez yinelemeniz gerekebilir.

## 4.3 Sınıf Açıklamaları

Bu bölümde her bir sınıfı, niteliklerini ve yöntemlerini ayrıntılı olarak açıklayacaksınız. Sınıfları mantıksal olarak birlikte gruptandırmanız. Örneğin, bir alt sistemdeki sınıfları birlikte gruptandırmak için mimari diyagramlarınızı kullanabilirsiniz.

Her sınıf için bir alt bölüm oluşturunuz. Her sınıf için amacını, kısıtlamalarını (örn. yalnızca tek bir örnek) kısaca açıklayın ve sınıf diyagramındaki her sınıfın niteliklerini ve yöntemlerini listeleyin.

Her sınıf için, özniteliklerinin her birini şu ayrıntılarla tanımlayın: ad, tür, anlamı sezgisel değilse özniteliğin bir satırlık açıklaması ve öznitelik üzerindeki kısıtlamalar (örneğin, öznitelik her nesne için benzersiz bir değere sahip olmalıdır veya değer aralığı pozitif tam sayılarla sınırlıdır).

Her yöntem aşağıdaki ayrıntılarla açıklanmalıdır: yöntem adı, dönüş türü ve değeri, parametreler, amaç ve kullanılan algoritmanın kısa bir açıklaması (önemsiz değilse). Argümanlar veya geri dönüş değerleri hakkında herhangi bir varsayım varsa, ön koşullar ve son koşullar burada belirtilmelidir. Bu yöntem ve bu yöntem tarafından çağrılan diğer yöntemler tarafından okunan ve değiştirilen öznitelikleri listeleyin.

**Aşağıda örnek olarak verilmiştir:**

## 4.4 Yakıt Alt Sistemindeki Sınıflar

### 4.4.1 Sınıf Tank

- Amaç: Yakıt depolayan fiziksel tankın ilgili yönlerini modellemek
- Kısıtlamalar: Hiçbiri
- Kalıcı: Hayır (sistem başlatılırken mevcut diğer verilerden oluşturulur)

#### 4.4.1.1 Öznitelik Açıklamaları

1. Özellik: yakıt seviyesi  
Tür: gerçek (çift hassasiyetli)  
Açıklama: Depodaki mevcut yakıt seviyesini saklar  
Kısıtlamalar: 0 ile maksimum seviye arasında bir değer olmalıdır

2. Öznitelik: *max-level*  
Tür: *gerçek (Çift hassasiyetli)*  
Açıklama: *deponun tutabileceği maksimum yakıt seviyesi*  
Kısıtlamalar: *negatif olmayan*

3. Başka bir özellik ...

#### 4.4.1.2 Yöntem Açıklamaları

1. Yöntem: *AdjustLevel(double pressure)*  
Dönüş Türü: *boolean*  
Parametreler: *basınç - tank için geçerli basınç değeri*  
Dönüş değeri: *başarılı veya başarısız*  
Ön koşul: *yakıt seviyesi 0 ile maksimum seviye arasında* Son koşul: *yakıt seviyesi 0 ile maksimum seviye arasında* Öznitelikler okunan/kullanılan:  
*yakıt, pompa, alarm, yakıt seviyesi*  
Çağrılan yöntemler: *fuel.get\_density(), alarm.sound\_alarm()*

İşleme mantığı:

*Yakıt yoğunluğu tankın yakıt özelliğinden elde edilir. Yeni yakıt seviyesi yoğunluk ve basınçtan hesaplanır. Seviye [düşük seviye, yüksek seviye) aralığının dışına düşerse, tankla ilişkili alarm çalınır ve pompa durdurulur.*

Test durumu 1: *AdjustLevel'i basınç X ve yakıt seviyesi Y ile çağırın. Beklenen çıktı: ....*

## 5 Dinamik Model

*Bu bölümün amacı, sistemin çeşitli olaylara nasıl tepki verdiğini, yani sistemin davranışını modellemektir. Bunu UML sıra diyagramlarını kullanarak yapıyoruz.*

*İlk adım, farklı senaryoları (örneğin Yakıt Seviyesinin Aşılması) tanımlamak ve gereksinimler belgenizde her bir kullanım senaryosunu ele aldığınızdan emin olmaktır. Senaryolar icat etmeyin, bunun yerine genel bir kılavuz, müşteri için anlamlı olabilecek senaryoları dahil etmektir. Örneğin, kurs kayıt sistemi için oturum açmak geçerli bir senaryodur.*

### 5.1 Senaryolar

*Her senaryo için aşağıdaki bilgileri içeren bir alt bölümünüz olacaktır:*

- *Senaryo Adı: Senaryo için anlamlı bir isim verin (bunu alt bölüm adına dahil etmek iyi bir fikirdir.*
- *Senaryo Açıklaması: Senaryonun ne hakkında olduğunun ve gerçekleşen eylemlerin sırasının kısa bir açıklaması*
- *Sıra Diyagramı: Çeşitli olayları ve bunların göreceli zaman sıralamasını gösteren bir dizi diyagramı (sınıfta tartışıldığı gibi).*

## 6 Tamamlayıcı Dokümantasyon

Tasarımın anlaşılmasına yardımcı olabilecek diğer ilgili belgeleri sağlayın.