

Yazılım Testi ve Proje Yönetimi

04.03.2024

Requirements Engineering Elizabeth Hull, Ken Jackson, Jeremy Dick
isimli kitaptan hazırlanmıştır

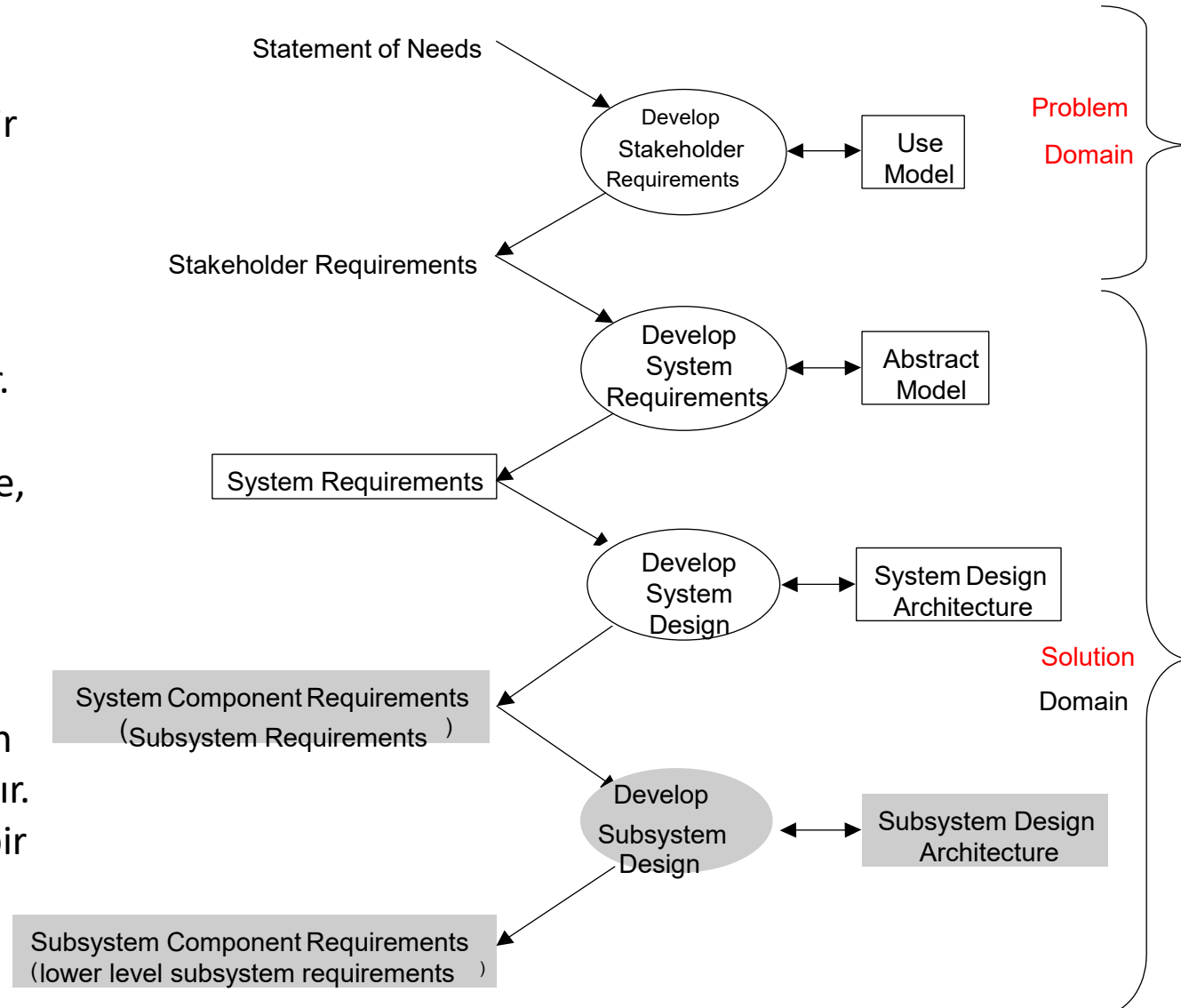
Software System Development Process

Stakeholder (iştirakçi): Bir sistemle doğrudan ilişkisi olan bir kişi, bir grup insan, bir kuruluş veya başka bir varlık

Stakeholder Requirements: Önerilen sistemle neler yapılabileceğinin belirlenmesidir.

Doğrudan tasarıma geçmektense, sistemin hangi özelliklere sahip olması gerektiği belirlenir.

Sistem gereksinimlerinin belirlenmesi sürecidir. Önerilen sistemin soyut bir modeli çıkarılır. Bu, belgelendirir. Her öge bir bölüm olarak değerlendirilir.



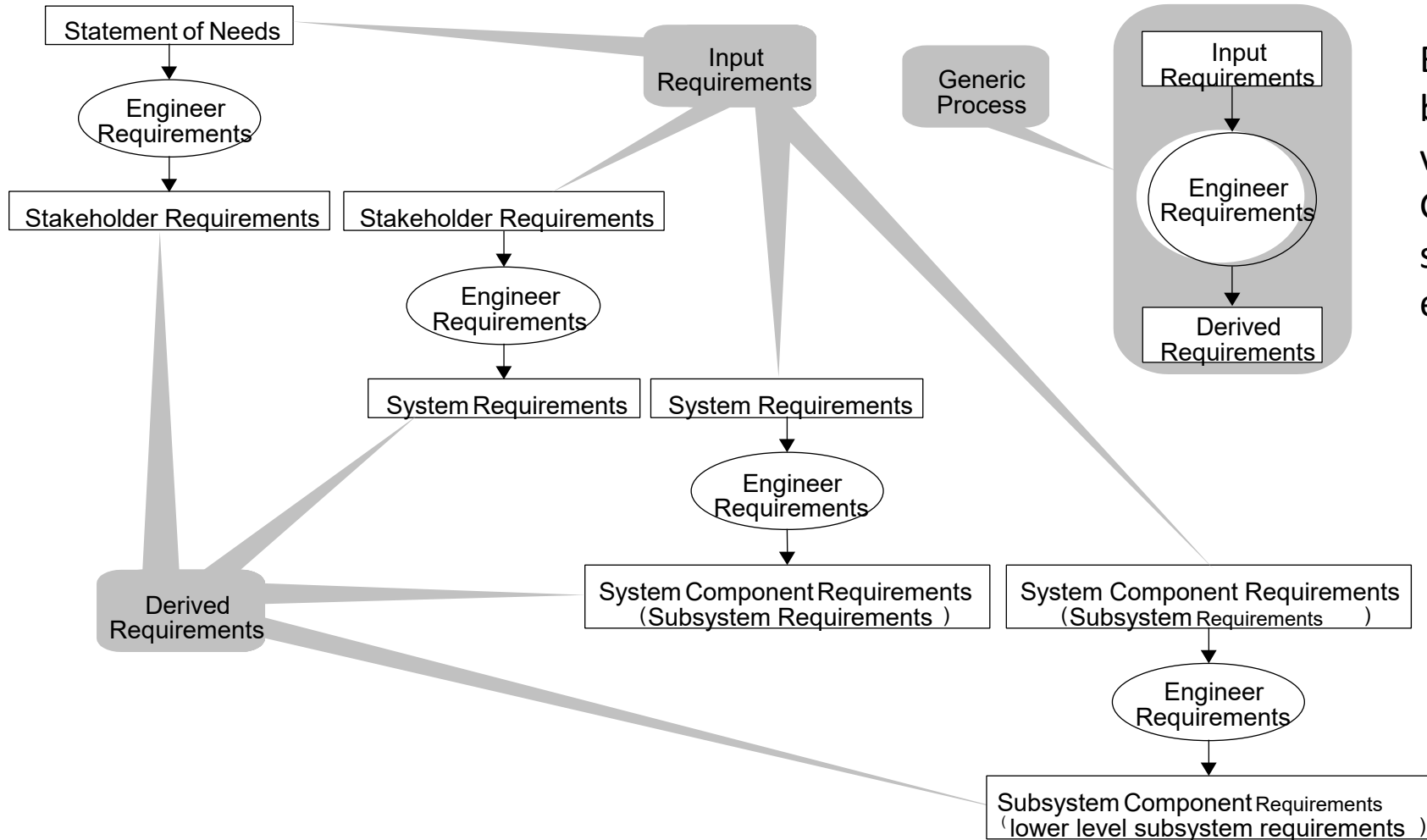
Tasarım mimarisi: Özelliklerin birlikte sergilendiği, etkileşim halindeki bileşenler kümesidir.

Her sistem bileşenine ait gereksinimler, işlevsellik ve etkileşim yükümlülüklerini tanımlamalıdır.

Sistem bileşeni gereksinimleri fiziksel boyut, performans, güvenilirlik, sürdürülebilirlik gibi özellikleri içermelidir.

Tasarım mimarisine ait bileşenler karmaşık oldukları için doğrudan uygulanamazlar. Bunlar alt sistemlerdir. Ama yüksek seviyeli sistemin yalnızca bir parçasıdır.

Proje Gereksinimleri Betimlemelerinin Genel Süreçleri



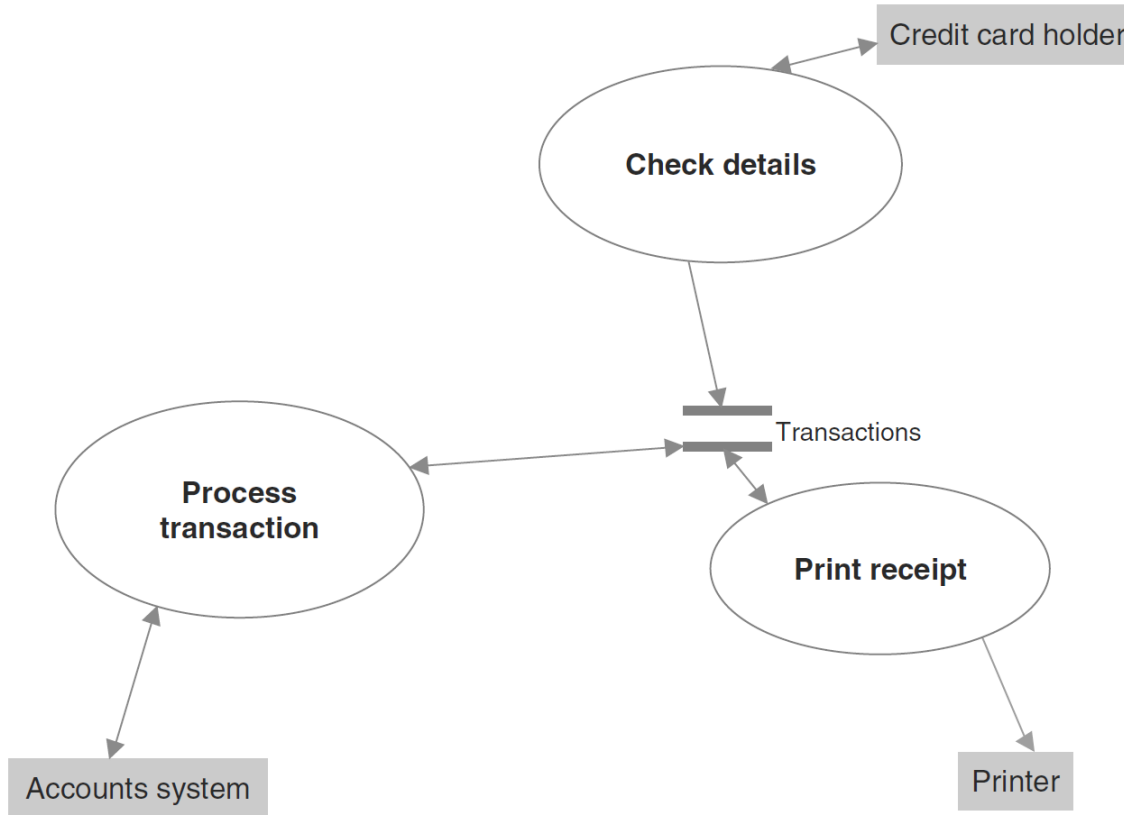
Bir süreç tarafından türetilen gereksinimler başka bir sürecin girdi gereksinimlerini vurgular.

Genel «Engineer Requirements» sürecinin Input Requirements olarak kabul edilir ve Derived Requirements üretir.

Sistem Modellemede Farklı Gereksinimler Mühendisliği Betimlemeleri

- ❑ Veri Akış Diyagramları (Data Flow Diagrams / DFD)
- ❑ Varlık İlişki Diyagramları (Entity Relationship Diagrams / ERD)
- ❑ (Durum Diyagramları (Statecharts)
- ❑ Nesneye Yönelik Yaklaşımlar
 - ❖ Sınıf Diyagramları
 - ❖ Use Cases

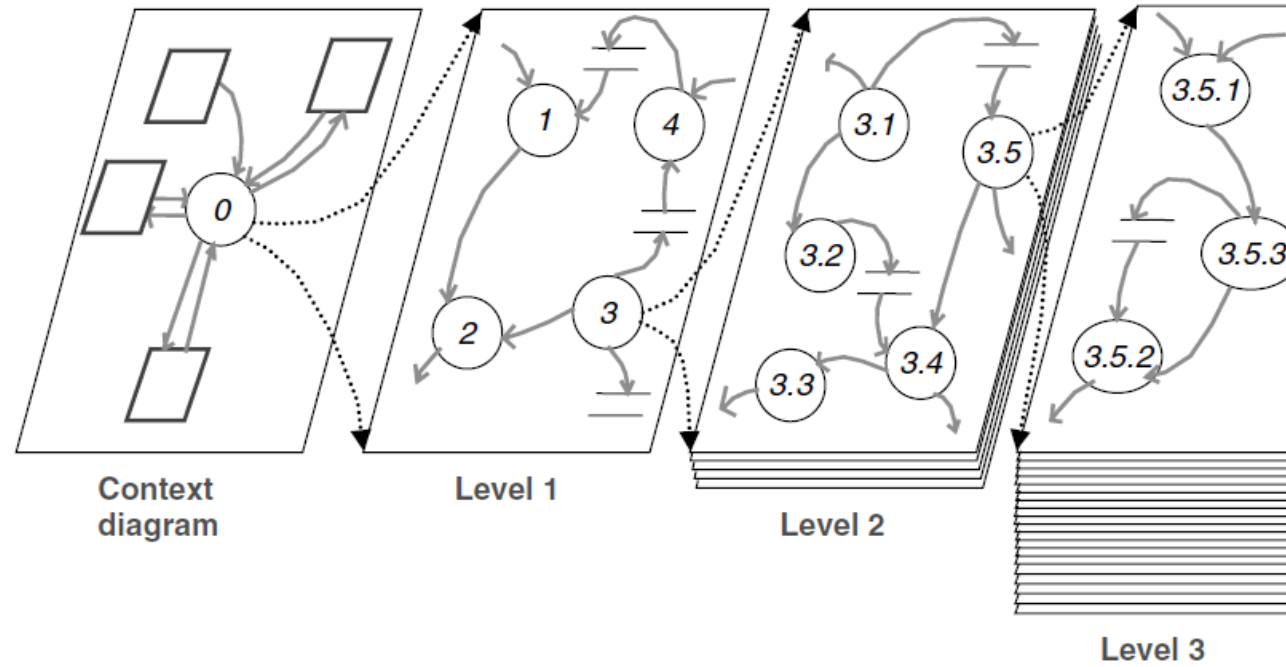
Basit bir DFD Örneği

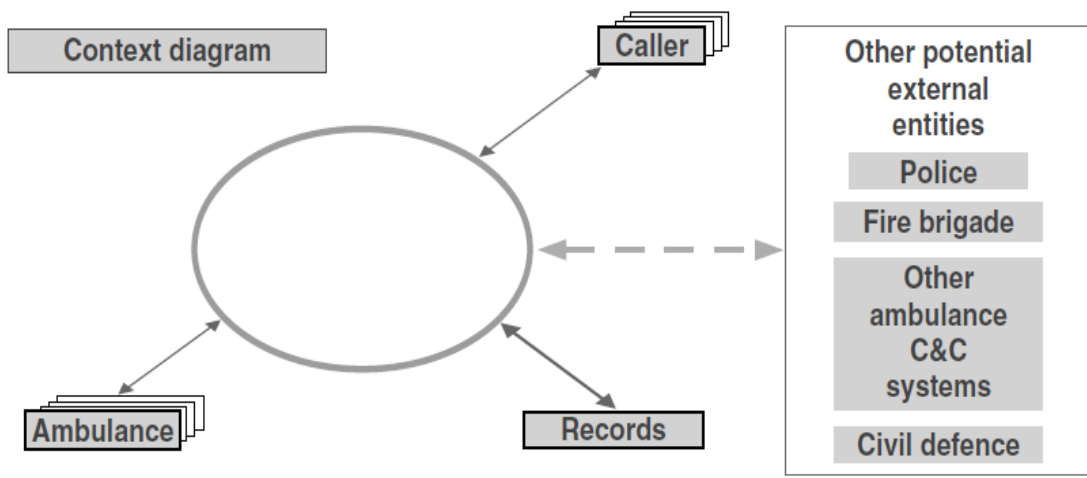


- DFD diyagramının temel elemanları:
- Veri akışları (yönlendirilmiş oklar)
 - Veri dönüşümleri (daire ya da elipsler)
 - Veri depoları (yatay paralel doğrular)
 - Dışsal (çevresel) varlıklar (dikdörtgenler)

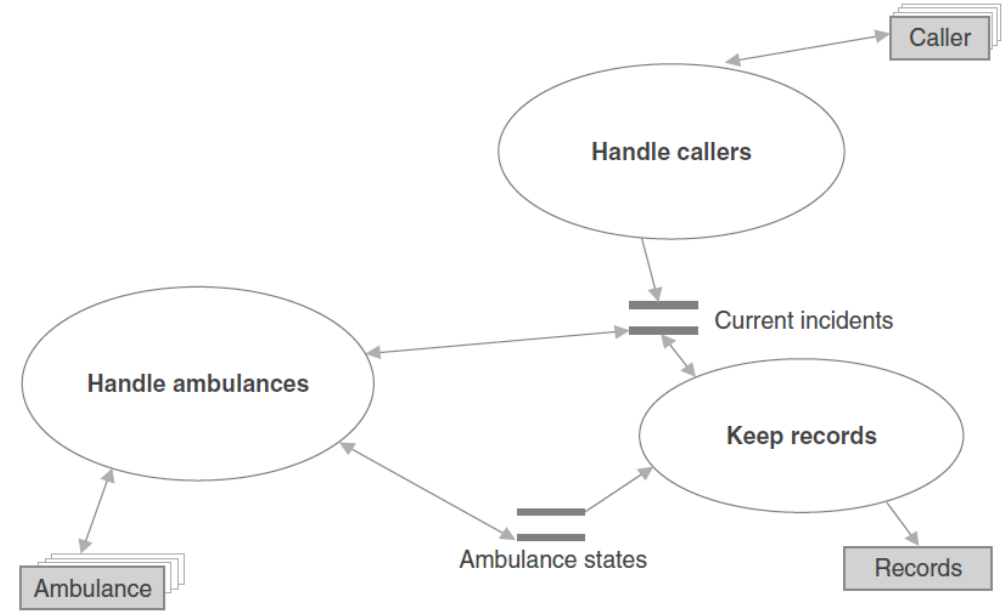
Context diyagram bir DFD 'nin en üst katmanıdır. Bir veri akışı analizine ait sistemin başlangıç noktasıdır.

DFD için Context Diyagram ve Fonksiyonel Ayırıştırması

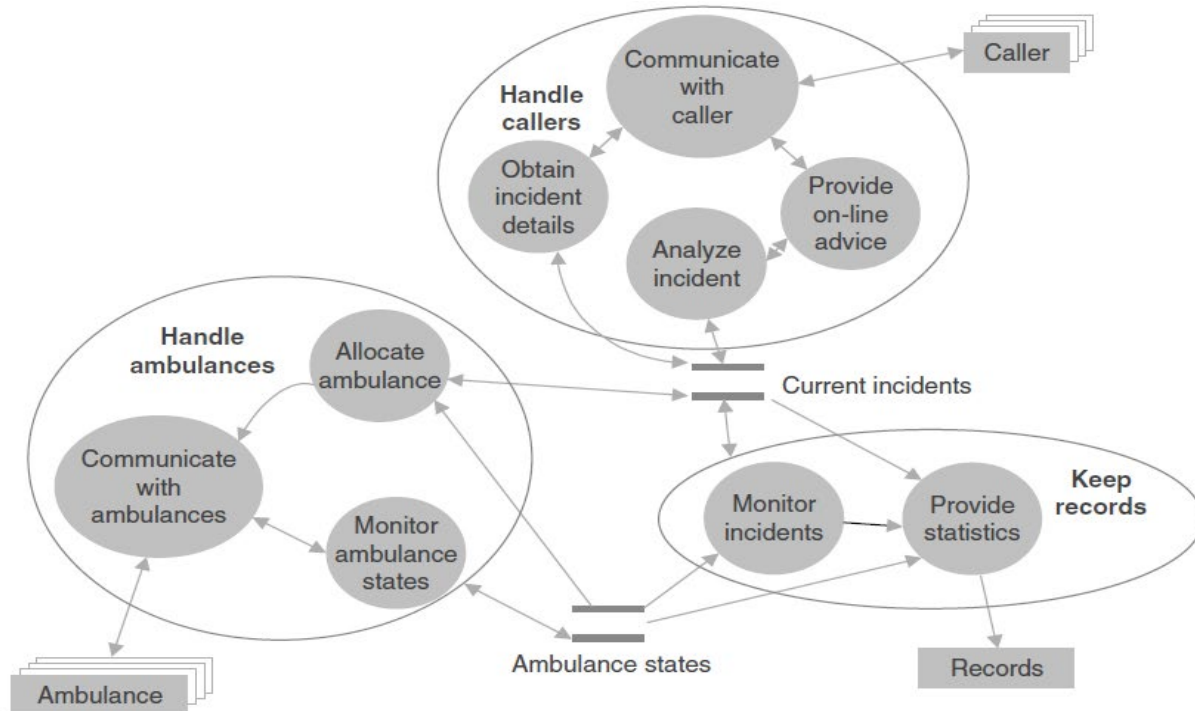




Ambulans Sistemine ait Context Diyagramı

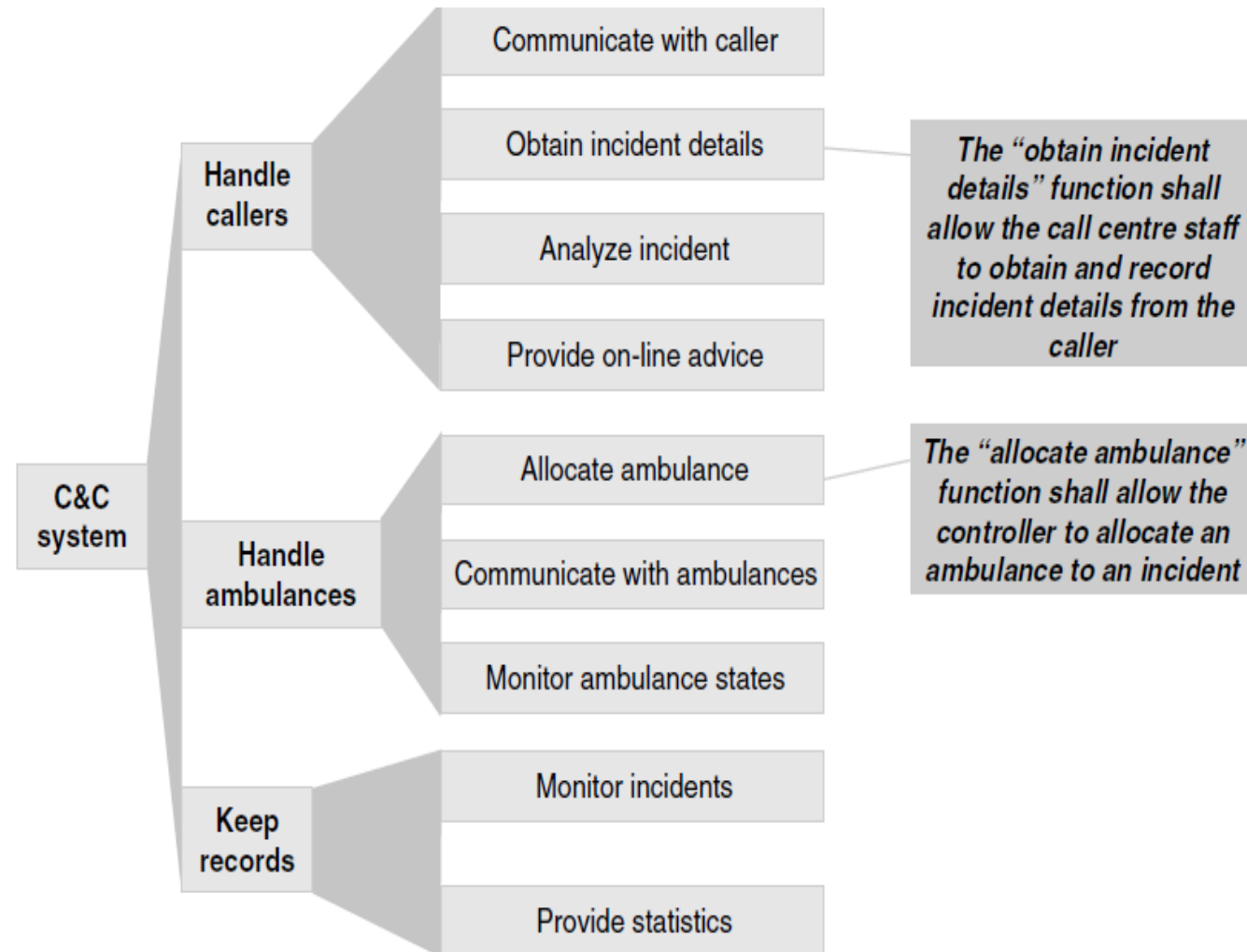


Ambulans Sistemi Modelinin Analizine ait üst düzeydeki DFD diyagramı

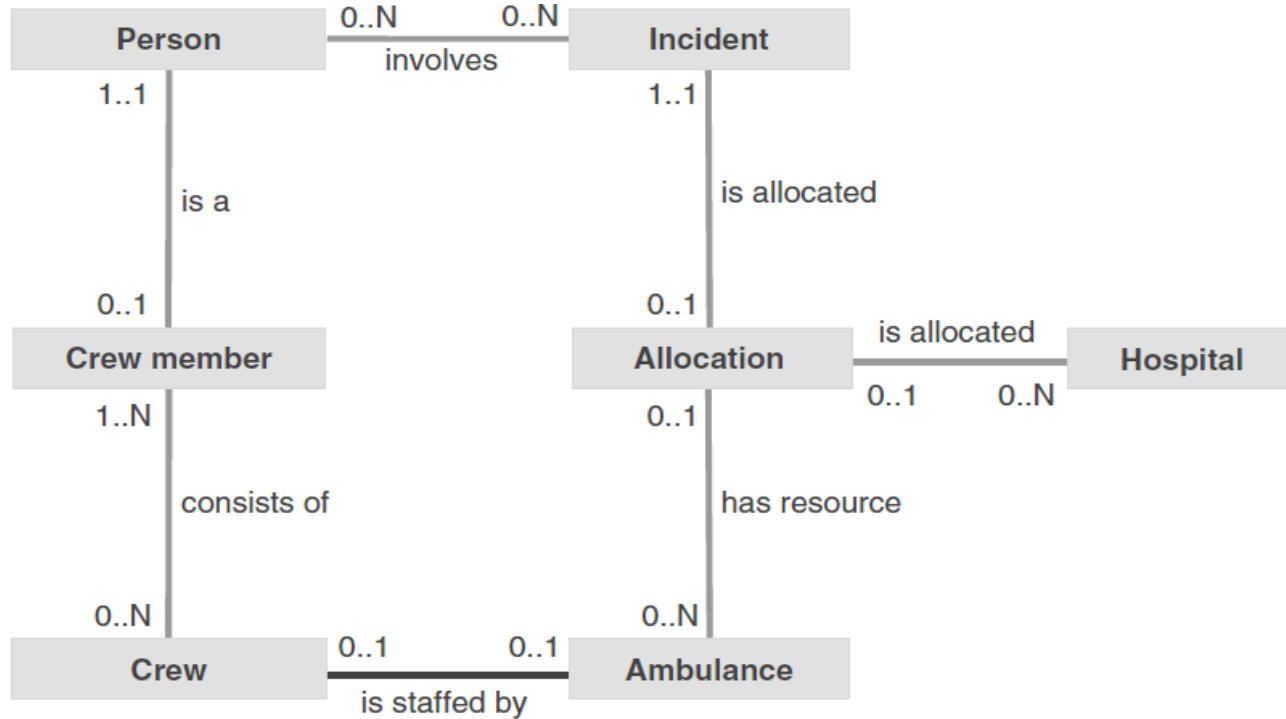


Ambulans Sistemine ait Ayrıntılı DFD Modeli

Ambulans Sisteminin Fonksiyonel Yapısı



Ambulans Sistemi için Entitiy –Relationship Diyagramı (ERD)



ERD diyagramı sistem içindeki varlıkları ve aralarındaki ilişkileri tanımlar. Bu da sistemin kısmi bir modelidir.

Bilginin (information) üretimi veya kullanılması için gerekli işlemlerden bağımsızdır.

ERD, sistem gereksinimleri aşamasındaki soyut modelleme çalışmalarında kullanılabilir.

Entity (varlık) bir nesnedir. Özellik (**property**) /**attribute** (öznitelik) varlığı betimleyen bilgidir. **Relationship** (ilişki) birleşimin kardinalitesini belirler. Birden-bire (one-to one), birden-çoğuna (one-to many), çoktan-çoğuna (may to many) gibi birliktelikler belirler.

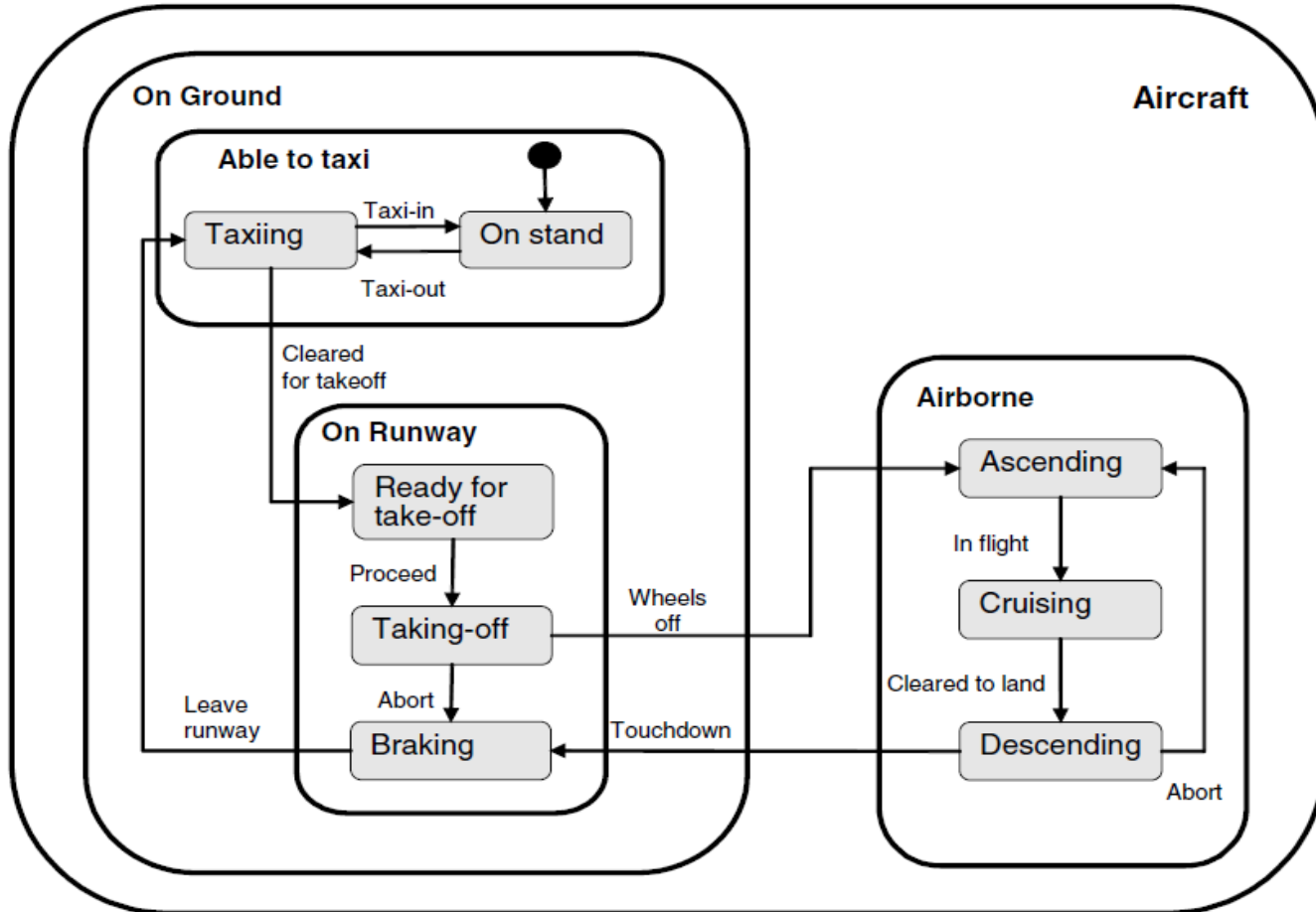
Durum Çizelge Diyagramları (Statecharts)

Gereksinimlerin tanımlanması için fonksiyonellik ve veri akışlarının belirlenmesi yeterli değildir.

Sistemin davranışını betimlemek ve bazı durumlarda sistemi, durumlar arasında geçişlere yol açan tetikleyiciler olarak hareket eden harici olaylarla birlikte, sabit sayıda olası "duruma" sahip bir yapıda çözümlenmek gereklidir.

Durum diyagramları (çizelgeleri) sistemin davranışsal yapısını belirler.

Durum Çizelgesi Diyagramı



Bu hiyerarşi **encapsulation** olarak ifade edilir.

Herhangi olayın (event) açıklamasıyla etiketlenen yönlendirilmiş oklar, durumlar arasındaki geçişi belirtir.

Durum, olay ve geçişlerin durum çizelgeleriyle tanımlanmasıyla, sistemin tümünün modellenmesi gerçekleşmiş olur.

Örnek bir Yazılım Projesinin Analiz ve Tasarımı: Oyun Uygulaması

Bu proje ile isimli bir korku-gerilim oyunu geliştirilmektedir.

Oyuncular, ana karakter 'e rehberlik etmektedir;virüsünden etkilenen tehlikeli hastane dünyasında hayatta kalmaya çalışmaktadır.

Oyun bulmaca /gizem çözme ve korku unsurlarını içerir ve oyuncuya zekasını kullanma yeteneği sunar.

Hikâye, ana karakterin hastaneden kaçması ve dünyayıvirüsünün etkilerinden korumasını hedeflemektedir.

Projenin yenilikçi hikâye anlatımı, heyecan verici deneyimleri ve bilimsel bağlantıları birleştirilerek oyun endüstrisine katkıda bulunmak hedeflemektedir.

Projenin Amaç ve Hedefleri

□ Projenin amacı: Oyun geliştirme süreçlerini uygulayabilme olanaklarının sağlanması ile oyun tasarımı, hikâye anlatımı, programlama ve oyun geliştirme becerilerini amaçlar.

Aynı zamandaamaçlanmaktadır.

Bu hedefler, projenin anlamlı deneyimler sunmasını ve oyun endüstrisine katkıda bulunmasını sağlar. Oyununu geliştirirken, oyuncularınve kendilerini hikâyenin bir parçası gibi hissetmelerini sağlamak amaçlanır.

❖ Yenilikçi oyun deneyimi :

❖ Gerilim ve Korku Deneyimi:

❖ Bulmaca Çözme.....

❖

❖ Çoklu Platform Erişim

❖ Bilimsel ve Tıbbi Farkındalık

❖

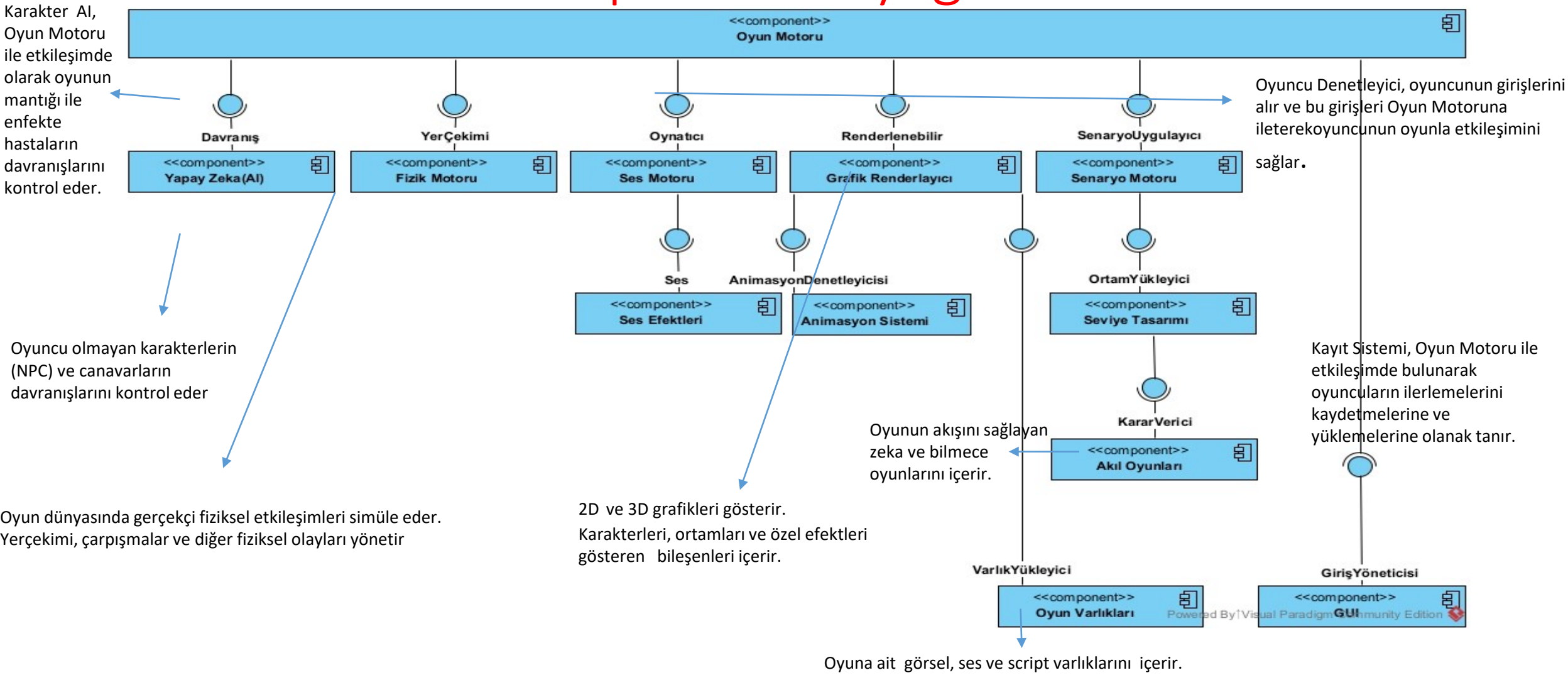
Projenin Kapsamı

Geliştirilecek ürün, oyun endüstrisinde yer alan korku ve gerilim türündeki oyunlara yenilikçi bir yaklaşım sunmayı hedeflemektedir. Oyunun niçin yeniden geliştirildiği, mevcut oyunlarda gözlemlenen eksikliklerin giderilmesidir. Mevcut oyunlarda oyun hikayesi, karakter gelişimi ve bilimsel farkındalık unsurlarını etkin ve etkileşimli bir araya getirme konusunda eksiklikler gözlemlendi . Oyun,oyunculara yanı sıraile daha fazla cezbetmeyi hedefler.

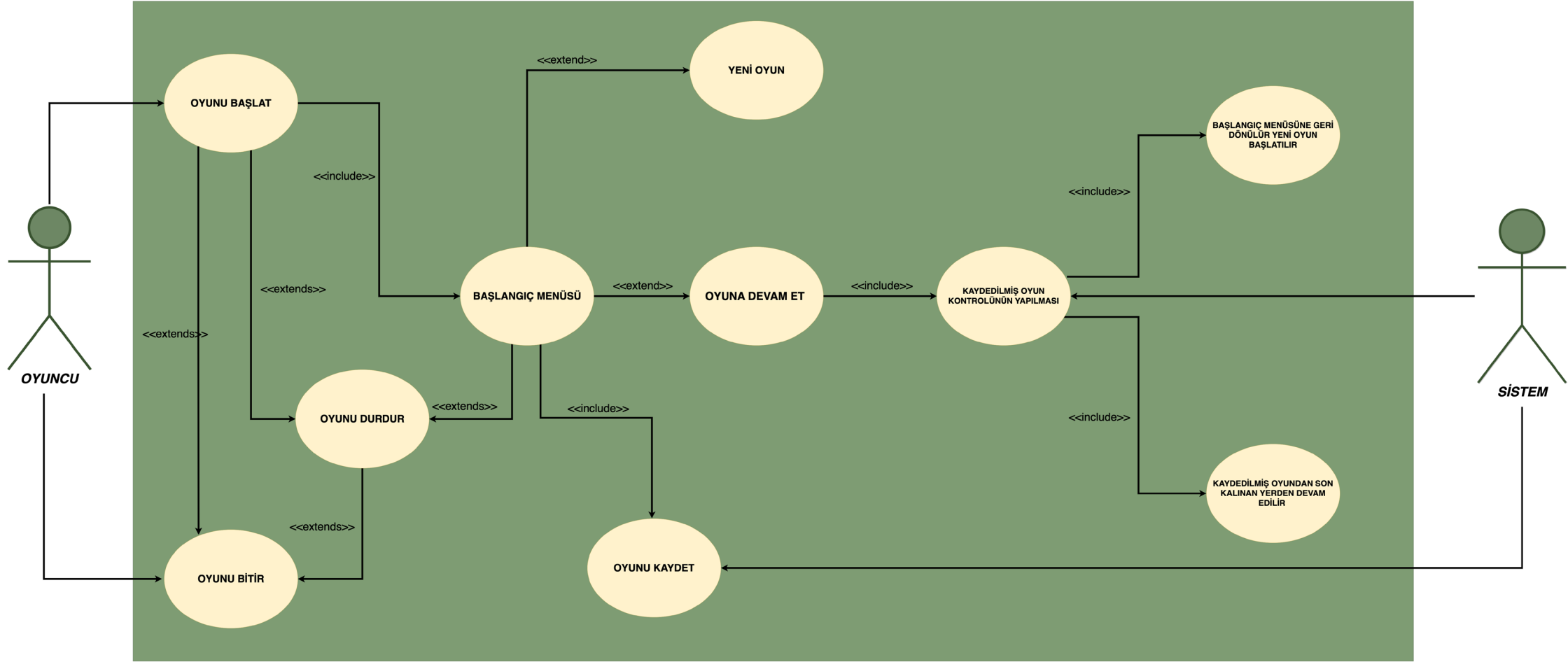
"" oyununun hikayesiamaçlar. Bu, oyuncuların oyun onanırkenkatkıda bulunur. Mevcut sistemlerde bu türeksik olduğu gözlemlenmiştir. Oyun ayrıca mevcut korku ve gerilim oyunlarının sınırlı kaldığı ve oyunculara daha fazla olanağı sunar.

".....", oyuncularınbecerilerini geliştirmelerini hedefler. Sonuç olarak, "....." projesinin geliştirilmesinenedeniyle gerek duyulmaktadır.

Component Diyagram



Use Case Diagramı: Başlangıç Menusu Tasarımı



<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Use Case Diyagramı: Başlangıç Menüsü Tasarımı

Use Case Numara:	1
Use Case İsmi	<i>Başlangıç Menüsü Kullanımı</i>
Ön Koşul	<i>Oyuncunun oyunu başlattığı varsayılır</i>
Son Koşul	<i>Oyuncu Ana menü ekranını kullanarak oyununu başarılı bir şekilde başlatmıştır</i>
Temel yol	<ol style="list-style-type: none"><i>1. Kullanıcı Oyunu başlat butonuna tıklar</i><i>2. Kullanıcı karşısına çıkar ara yüzden yeni oyun seçeneğini seçer.</i><i>3. Oyun yazılımı oyunu başlatır.</i>
Alternatif Yol	<ol style="list-style-type: none"><i>1. Kullanıcı Oyunu Başlat butonuna tıklar</i><i>2. Oyuna devam et butonuna tıklar</i><i>3. Oyun yazılımı kaydedilmiş bir oyun var mı kontrolü yapar</i><i>4. Oyun yazılımı kaydedilmiş bir oyun dosyası bulur ise</i><i>5. Oyunu son kalan yerden devam ettirir</i>
Alternatif Yol:	<ol style="list-style-type: none"><i>1. Kullanıcı Oyunu Başlat butonuna tıklar</i><i>2. Oyuna devam et butonuna tıklar</i><i>3. Oyun yazılımı kaydedilmiş bir oyun var mı kontrolü yapar</i><i>4. Oyun yazılımı kaydedilmiş bir oyun dosyası bulamaz ise</i><i>5. Oyun yazılımı başlangıç menüsüne geri döndürür.</i>
Alternatif Yol:	<ol style="list-style-type: none"><i>1. Oyuncu kaydet butonuna tıklayarak oyunu sonlandırır</i>
Alternatif Yol:	<ol style="list-style-type: none"><i>1. Kullanıcı oyunu başlat butonuna tıklar</i><i>2. Kullanıcı oyunu durdur butonuna tıklar</i><i>3. Kullanıcı oyunu bitir butonuna tıklar</i>

Nonfunctional Requirements

Accessibility	Efficiency	Reliability
Availability	Extensibility	Responsiveness
Compatibility	Failure Transparency	Robustness
Certification	Fault Tolerance	Safety
Compliance	Flexibility	Scalability
Configurability	Interoperability	Securability
Dependability	Maintainability	Supportability
Deployability	Operability	Testability
Documentation	Performance	Traceability
Disaster Recovery	Recoverability	Usability