

Beykent Üniversitesi
Yazılım Mühendisliği Bölümü
Yazılım Mühendisliği Tasarım Projesi

Rapor 1
2022 – GÜZ

Proje İsmi (kısaltması ile birlikte)

Grup Elemanları
No, Adı Soyadı (Alfabetik Sırada)

I Proje Tanımı (16 font)

1 Projeye Genel Bakış (14 font)

Ayrıntılara girmeden geliştirilecek ürün ile ilgili kısa bir tanımlama yapılır (12 font).

2 Projenin Amacı (14 font)

Takım bu projeyi niçin seçmiştir? Projenin kapsamı nedir?

Proje çalışması ile yapılması hedeflenenlerin özetlenmesi istenir. Bu bölümdeki açıklamalar projenin gerekçesini oluşturacaktır. Böylece problemin proje olarak seçiminin önemi açık olarak ifade edilecektir.

Özetlenecek olursa bu bölümde, çalışma konusunun niçin seçildiği açıklanır. Çalışmanın sonraki aşamalarında burada açıklanan ifadelerin sağlanıp sağlanmadığının kontrol edilmesi gerekecektir (12 font).

3 Projenin Kapsamı (14 font)

3 a. Mevcut Durum

Geliştirilecek ürün ile ilgili olarak proje takımının ilk defa çalışmadığı kabul edilmektedir. Piyasada güncel olan yazılım ürünlerinin pek çoğu da zaten önceden geliştirilmiş ve baştan sonra yeniden düzenlenmiş (refactored) ürünlerdir. Yeni bir ürün geliştirirken iş analistleri eski ürün yerine yenisinin niçin geliştirilmesi ya da önceden elle yapılan işlerin neden otomatikleşmesi ya da değiştirilmesi gerektiğini araştırırlar.

3 b. Projenin İçeriği

Proje çalışmasına ait “component diagram” çizilmesi ve böylece çalışmanın görsel olarak üst düzeyde açıklanması istenmektedir. Bu bölümde sadece diyagram verilmesi yeterli değildir, görsel mutlaka açıklayıcı olarak yazılı metne dönüştürülmelidir.

Ders içeriğindeki tüm UML diyagramlarının “visual paradigm community edition” kullanılarak tasarımı önerilmektedir (<https://www.uml-diagrams.org/component-diagrams.html>). Çevrimiçi erişilebilen UML linkleri tasarım projenizin diyagramlarını çizmek için yeterli olmayacaktır. UML geliştirme araçları kullanılmadan yazılan görsel çözümler geçerli değildir.

Component diyagramı ile projenin sınırları belirlenmiş olacak ve bu çalışmanın niçin yapıldığı görsel olarak betimlenecektir. Özetle geliştirilecek ürün tüm alt bileşenleri ile birlikte açıklanmış olacaktır.

Çizdiğiniz tüm diyagramlar şeklinizin altında açıklanmalıdır.

Örneğin:

Şekil 3.1:projesine ait “component” diyagramı gibi...

Anlamı tam olarak vermeyen Türkçe ifadeler nedeni ile diyagram isimlerinin orijinal olarak kullanılması önerilir.

3 c. Rakip Ürün Karşılaştırması

Geliştirilecek ürün mevcut iken niçin yeniden geliştirilmesine gerek duyulmuştur? Diğer bir ifade ile mevcut sistemde ne gibi eksiklikler görüldü ki ürünün yeniden geliştirilmesi istenmiştir?

4 Ürün Senaryoları

Senaryolar, kullanıcıların ürünün geliştirilmesi sonunda nasıl kullanacağını açıklayan formal olmayan hikâyelerdir ve ürün senaryoları listesi şeklinde (1, 2, 3, 4,... olarak) sözel olarak açıklanacaktır.

Örneğin:

Senaryo 1: Başlangıç Menüsü Kullanımı

Senaryo 2: Oyuncu karakterin davranışı ve temel mekanikleri

Senaryo 3: Düşman karakterinin davranışı ve temel mekanikleri

Senaryo 4: Non-Player karakterle Player karakterinin etkileşimi

Senaryo 5: Kapı objesi ile oyuncu karakterinin etkileşimi

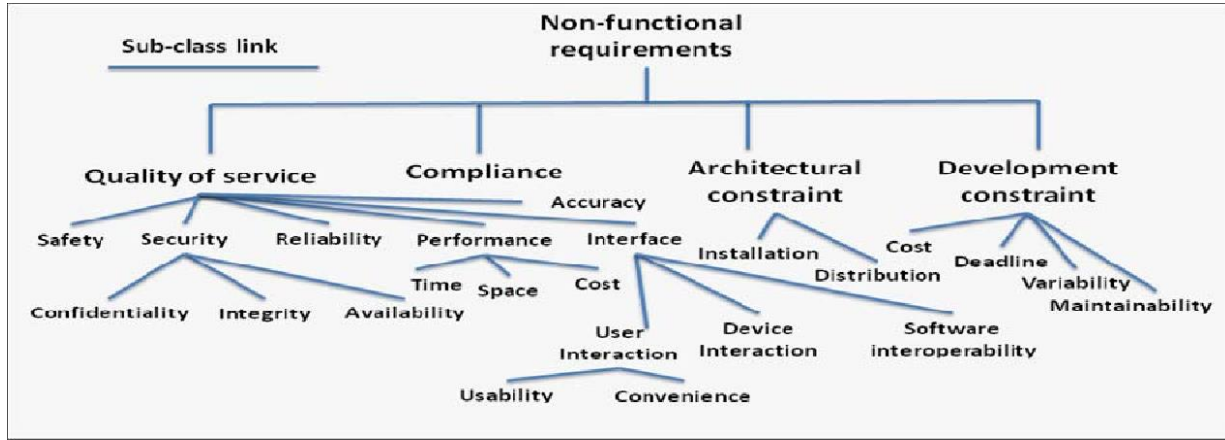
gibi

Projenizin eylemlerini sözlü açıkladığınız bu bölümde senaryolarınızı alt bölümlere ayırarak ta açıklama yapabilirsiniz. Bu bölümün çalışmanın başında belirlenmiş olmasının önemini fark ediniz.

Bu bölüm özetlenecek olursa II Gereksinimlerin Analizi Bölümünde, kullanılacak olan her bir *use case* diyagramına ait ürün senaryoları (aslında diyagram isimleri) listelenmiş olacaktır. Bu liste her bir senaryo için numaralandırılır. Böylece çalışma grubu ürün geliştirilirken kaç tane *use case* diyagramı kullanılacağına da karar vermiştir. Bu bölümde ürün senaryolarını açıklamak üzere SADECE “*use case*” diyagramları listelenir, kesinlikle çizilmez.

5 Kısıtlar

Çözülecek probleme özgü fonksiyonel olmayan gereksinimlerin tanımı, gerekçeleri ile birlikte, yapılmalıdır. Aşağıdaki sınıflandırmadan probleminiz için tanımlanması gerekli fonksiyonel olmayan gereksinimler seçilecek problem alanına göre seçilecek ve açıklaması yapılacaktır.



Non-Functional Requirements on Software Services (Axel Van Lamsweerde)

6 Tanımlar

Proje çalışması içerisinde kullanılacak simgeler ve kısaltmalar belirlenecek ve listelenecektir.

Örneğin

Non-Player Character (NPC): Kullanıcının kontrol etmediği karakter

Top-Down: Yukarıdan bakış

Texturing & Effects: Dokular ve efektler

Retro: Geriye dönük

Retro-Popüler: Geçmişte belirli bir dönem kendini göstermiş belirli şeylerin

günümüzde popüler, sevilen ve tercih edilen bir konumda olması hali

gibi

II Gereksinimlerin Analizi (16 font) (sayfa başı olmalıdır)

7 Use Cases ve Tablo Açıklamaları (14 font)

Uses cases (kullanım durumları), hem geliştireceğiniz yazılım sistemini ve sistemin sınırlarını ayrıntılı olarak tanımlamanızı, hem de fonksiyonel gereksinimleri tanımlamanızı sağlar.

Ürünün gereksinimlerinin belirlenmesi ve her birinin tablo üzerinde açıklanmasından sonra, sisteminizin sınıflarını tanımlayacağınız ve aralarındaki ilişkileri tasarlayacağınız tasarım aşamasına geçilecektir.

Bu bölümde tanımlanmış ve açıklanmış olan *use case* diyagramları çözümlemelerinin tasarım aşamasındaki tüm diyagramlara değiştirilmeden yansıtılması zorunludur. Rapor 2 içerisinde bu rapordaki *use case* diyagramları ile örtüşmeyen çözümler değerlendirilmeye alınmaz (<https://www.uml-diagrams.org/use-case-diagrams.html>).

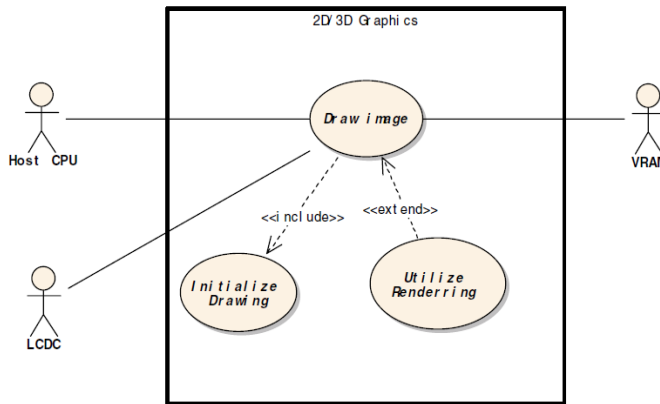
Bir *use case* diyagramı ve bu diyagramın açıklaması için aşağıda bir örnek verilmiştir.

Not 1 : Bilimsel çalışmalarda verilen bir şeklin açıklaması, *Şekil 1*:, şeklin altına, tablonun açıklaması ise, *Tablo 1*:, tablonun üstüne yapılır.

Not 2: *Use case* diyagramları ve tablolar sıralı olarak numaralandırılmalıdır.

Şekil 7.1, Tablo 7.1,

Şekil 7.2, Tablo 7.2 gibi....



Şekil 1: Use case diyagramı örneği (hani çözüm olduğu şeklin altında açıklanır)

Tablo1 : Use case diyagramı açıklaması örneği (tablonun üstüne yazılır)

Use Case	Draw Image
Pre-condition	<ul style="list-style-type: none">• Frame buffer must be cleared• Reset must switch off
Post-condition	<ul style="list-style-type: none">• Image data output to VRAM
Basic path	<ol style="list-style-type: none">1. This use case starts while Host CPU has already prepared the display list for drawing image.2. Host CPU writes display list system's FIFO.3. 2D/3D Graphics reads display list from FIFO and starts drawing image according to commands of display list.4. 2D/3D Graphics outputs image data to VRAM.
Alternative path	<p>At step 2 of the basic path, Host CPU writes display list that includes SYNC command to system's FIFO.</p> <p>At step 3 of the basic path, 2D/3D Graphics stops after reading SYNC command, and then 2D/3D graphics restarts drawing image after receiving a blank pulse from LCDC.</p>
Exceptional path	<p>At step 2 of the basic path, Host CPU writes display list includes undefined commands.</p> <p>At step 3 of basic path, 2D/3D graphics raises an error interrupt to Host CPU and clears system's FIFO.</p>

use cases sayısı, grup elemanlarının sayısına ve bu bağlamda gerçekleştirilecek projenin kapsamına göre 15-20 olmalıdır ve de giriş /çıkış yapılması gibi klasik basit diyagramlar olmamalıdır. Mutlaka bitirme proje olarak çalışacağınız karmaşık probleminizi açıklamalıdır.

Her bir ürün kullanım durumunun (*use case*) bir listesini yapmak, her birini ayrı ayrı modellemek veya tanımlamak proje çalışmasının analiz aşamasının izlenebilirliğini kolaylaştıracaktır. Bu aşamada çalışma grubunun her üyesinin aktif olması ve çözüm tartışmalarına katılması çalışmanızın kalitesi için önemlidir.

Kullanım durumu diyagramlarının geliştirilecek sistem üzerinde iki amacı vardır:

Her bir *use case* diyagramını listeleyen tablo, bu diyagramın sistemin bir parçası olarak hangi eylemleri içerdiğini ve de aynı zamanda sınırlamaları (kısıtları) ifade ederek sistemin neleri içermediğini gösterir.

Her bir *use case* diyagramının problemin ilgili parçasına ait kavramsal çözümde *include* ve/veya *extend* ilişkileri tanımlanmalıdır. Bu ilişkiler karmaşık bir problemin çözümüne doğal olarak bulunur.

Aktörlerin canlı ve/veya cansız bir varlık olması mümkündür.

Çözdüğünüz problemin karmaşıklığına bağlı olarak, bazı kullanım durumlarını (*use cases*) listelemek için birden fazla diyagram kullanılması mümkündür. Örneğin herhangi bir kullanım durumu (*use case*) için birden fazla şema gerekirse, kullanım alanları birkaç yola ayrılabilir: normal işlemlere karşılık göz önüne alınmayan durumlar veya günlük işlemlere karşılık aylık işlemler ya da kullanıcı görevlerine karşılık yönetim işleri gibi.

Proje grubu geliştireceği ürünün sınırının ne olması gerektiğine karar vererek ürünlerine kullanım durumlarını (*product use cases*), diğer bir ifade ile *use cases* tasarlayarak probleminin gereksinimler analizini tamamlayacaktır. Bu sınırın ne olacağı (ne büyüklükte olacağı, diyagramların sayısı ve birbirlerine bağlılığı) çalışma konusundaki bilgilerinize ve gereksinim kısıtlamalarınıza göre belirlenecektir.

8 Fonksiyonel Gereksinimler

Her bir olay (event) ya da diğer bir ifade ile *use case* kullanımı listesi için fonksiyonel gereksinimler aşağıdaki tabloya yazılacaktır. Her bir fonksiyonel gereksinime *unique* bir sayı verilecektir. Bu bölüm çalışmanın izlenebilirliğini (traceability) ve geliştirme aşamasını kolaylaştıracaktır.

Tablo 2: Problem Gereksinimleri Tanımı

Section/ Requirement ID	Requirement Definition (Gereksinim Tanımı)
FR1.0.	The system shall [parent requirement group 1].
FR1.1	The system shall [child/parent requirement].
FR1.1.1	The system shall [child requirement].
FR1.1.2	The system shall [child requirement].

Gruplar bu bölümü Türkçe gelecek zaman (shall) cümleleri ile tasarlayacaklardır.

Örneğin (Çalışma grubu shall cümleleri kullanmamıştır)

Section ID	Gereksinim Tanımı
FR1.0.	Kullanıcı oyun yazılımı başlatır
FR1.0.1	Kullanıcı Start game butonuna basarak yeni bir oyun başlatabilir ya da kaldığı oyuna devam edebilir
FR1.0.2
FR1.1.	Kullanıcı verilen durumlardan birini seçerek karakterini bir aksiyona sokar
FR1.1.1
FR1.1.2	Player karakteri, enemy karakteri ile etkileşime girip zarar verebilir ya da zarar görebilir
FR1.2.
FR1.2.1	Enemy karakteri komut grubundan aldığı bir komut durumu ile..... aksiyonlarından birini gerçekleştirebilir