

Yazılım Mühendisliği Proje Raporu

Rapor-1: SİSTEM ÖZELLİKLERİ – İterasyon 1(a) –

Bu rapor üç adımda sunulacaktır:

1. Bölüm 1 (1.Kısım Müşteri Sorun Bildirimi ve 2. Kısım Sistem Gereksinimleri)
 2. Bölüm 2 (3.Bölüm İşlev Gereksinimleri Özelliği ve 4.Bölüm Kullanıcı Ara birimi Özelliği)
 3. Rapor-1'in tamamı (tüm Bölümler dâhil)
-

Bu belge Gereksinim Analizi Belgesi (GAB) olarak bilinir. Ayrıca bakınız : Rapor-2: SİSTEM TASARIMI.

1. Rapor Formatı
2. Katkı Dağılımı
3. Rapor Hazırlığı
4. Kodlamaya Ne Zaman Başlanmalı?
5. Rapor Değerlendirmesi
6. Raporu Teslimi

NOT: Bu raporu ileride kullanmak üzere saklayın çünkü son raporunuzun bir parçası olacaktır (Rapor-3). Bu rapor sürekli olarak gözden geçirilmeli, güncellenmeli ve / veya düzeltilmelidir, böylece iyileştirilmiş bir sürüm Rapor-3'ün bir parçası haline gelmelidir.

1.Amaç

Proje raporlarının amacı, tasarım fikirlerini ve uygulama seçeneklerini diğer üyelere iletmektir; amaç “doküman yazmak” değildir.

Öğrenci projeleri durumunda, iletişim öncelikle eğitmen ve sınıf öğrencileriyle kurulur. Etkili iletişim olmadan, çalışmanın özgünlüğünü, kalitesini ve kapsamını değerlendirmek mümkün değildir.

Bu nedenle, rapor yazarken kendinize şunları sorun: “Çalışmamın temel fikirlerini ve önemli yönlerini etkili bir şekilde yansıtıyor muyum?”, "doküman yazmakla" uğraşmayın.

Formalitelere değil, içeriğe odaklanın. En önemli şey, projenizin yeni, yaratıcı, profesyonelce yürütülen ve diğer takım arkadaşlarıyla iletişim içinde hazırlanmış

olmasıdır. Kullanılan örneklerin sayısını veya raporunuzun sayfa sayısını saymaya odaklanmayın.

Kendinize sorun: iyi bir yazılım geliştiriyor muyuz? Bu rapor, not verecek kişilerin bunu anlayabilmeleri ve takdir edebilmeleri için çalışmalarımızı ve katkımızı net bir şekilde açıklıyor mu?

“İletişim kirliliği” yaratan ve etkili iletişimi engelleyen tipik hatalardan kaçınmaya çalışın.

2. Rapor Formatı

Aşağıdaki genel biçimin tüm proje türlerine uygun olmayabileceğini unutmayın. Farklı projeler, 3.Bölüm İşlev Gereksinimleri Belirtimi ile 4.Bölüm Kullanıcı Ara birimi Spesifikasyonu için farklı çaba dağılımı gerektirecektir. Tüm projeler her iki bölümü de gerektirir, ancak ilgili tasarım değiştirilebilir. Bazı projeler sistemlerinin işlevsel davranışlarına odaklanmalıdır (3.Bölüm). Bu tür projeler arasında park yeri otomasyonu, restoran otomasyonu, trafik izleme ve yatırım oyunu bulunmaktadır.

Diğer projeler kullanıcı ara yüzü spesifikasyonuna (4.Bölüm), özellikle video oyunu veya sanal laboratuvarlar geliştiren projelere odaklanmalıdır. Kullanım durumları nispeten daha basit olabilir.

Son olarak, bazı projeler nispeten basit kullanım durumlarına ve kullanıcı ara yüzüne sahip olacak, ancak karmaşık simülasyon algoritmalarına odaklanacaktır. Bakınız; Azınlık Oyunu(Minority Game). Algoritmalarının ayrıntılı açıklamaları Raporlarının Matematiksel Modeli Bölüm 5.c de bulunmalıdır.

Rapor, aşağıda listelenen bölümleri içermelidir. Her Bölüm kendi başlığı ve sayfalandırması ile açıkça tanımlanmalıdır. Sınıflandırma işlemini kolaylaştırmak için bölümler aşağıdaki gibi numaralandırılmalıdır.

- Yazılım mühendisliği 5. Bölümde belirtildiği gibi Kapak Sayfası ve Bireysel Katkı Dağılımı.

Katkı dağılımı, aşağıdaki Bölüm 2'de açıklanan sorumluluk matrisini ve sorumluluk tahsis tablosunu içermelidir.

Tüm ekip üyeleri eşit katkıda bulunsa bile, bunun Katkı Dağılımı sayfasında açıkça belirtilmesi gerekir.

- **İçindekiler**

Burada listelenen bölüm başlıklarının ve numaralarının ve sayfa numaralarının doğru olup olmadığını kontrol edin.

Rapor bölümleri aşağıdaki gibidir:

Bölüm 1:

1. Müşteri Sorun Bildirimi

a. Sorun bildirimi

Projeniz hakkında minimum 3 sayfalık üst düzey bir anlatım. Anlatım, planlanan sistemin özelliklerini açıklayan geliştiricinin bakış açısıyla yazılmamalıdır.

Bunun yerine, kendinizi müşterinin yerine koyun ve sorun ifadenizi hayal ettiğiniz müşterinin yazdığı gibi yazın! – Müşterinizin karşılaştığı sorunu ve bir yazılım sisteminin nasıl yardımcı olabileceği konusundaki önerilerini açıklayın. Sorununuz, projenize dayanmalı, gerektiğinde revize edilmeli ve geliştirilmelidir.

b. Terimler Sözlüğü

Sistem şartnamesinde tutarlılığı sağlamak ve belirsizliği önlemek için önemli terimleri ve tanımlarını listeleyin. Uygulama etki alanının dilini kullanın ve nadir olmayan terimlerden kaçının veya bunları tanımlayın.

Okuyucunun anlamasına yardımcı olacak görüntüler ve grafikler sağlayarak karmaşık terimleri göstermek işinize yarayabilir. (internetten görseller bulun).

“Ağ protokolü”, “istemci / sunucu”, “kullanıcı ara birimi” gibi genel bilgi işlem terimlerini tanımlamayın.

Yalnızca sisteminiz için tanıttığınız yeni terimleri veya sisteminizin bağlamında özel bir anlamı olan ortak kelimeleri tanımlayın.

2. Sistem Gereksinimleri

Not: Sistem gereksinimleri yerine, Kullanıcı Öyküleri yazmak isteyebilirsiniz (birini ya da diğerini yazın, ancak ikisini birden yazmayın).

a. Numaralandırılmış İşlevsel Gereksinimler

Gereksinimleri müşterinin anlatımından çıkarın ve gereksinimler başına bir satır olacak şekilde bir tabloda listeleyin. İlk sütun benzersiz bir “REQ-x” etiketi gösterir. İkinci sütun, bu gereksinimin atanmış Öncelik Ağırlığı gösterir. Üçüncü sütun kısaca gereksinimi açıklar.

b. Numaralandırılmış İşlevsel Olmayan Gereksinimler

FURPS + gereksinimlerini listeleyin, önceliklendirin ve açıklayın. İşlevsel olmayan gereksinimlerin numaralandırılması, işlevsel gereksinimler listesine devam etmelidir.

c. Kullanıcı Ara yüzü Gereksinimleri

Grafikleri ağır olan projeler için (sanal laboratuvarlar veya video oyunu gibi) kullanıcı ara yüzü, gereksinimlerin çoğunu oluşturur. Yine kullanıcı ara yüzü gereksinimlerini listeleyin, önceliklendirin ve açıklayın, ancak her gereksinimi gösteren bir grafik de ekleyin. İnternet'te görüntüler bulabilir veya kâğıt üzerine elle çizilmiş eskizler yapabilir, sonra tarayabilir ve raporunuza resim olarak ekleyebilirsiniz. Bu grafikleri mükemmelleştirmek için zaman harcamayın, geliştirme Bölüm 4 Kullanıcı Arayüzü Spesifikasyonunun bir parçasıdır (aşağıda)

- **Proje Yönetimi** (aşağıdaki Bölüm 7'de açıklanmıştır) ve **Referanslar** (aşağıdaki Bölüm 8'de açıklanmıştır)

Bölüm 2:

3. Fonksiyonel Gereksinimleri Tanımlama

Kullanım durumlarını yukarıdaki Bölüm 1 ve Bölüm 2'deki gerekliliklere göre türetiniz.

Not: Sistem gereksinimleri yerine kullanıcı hikâyeleri yazdıysanız, yine de kullanım örnekleri yazmanız gerekir.

Paydaşlar

Bu sisteme ilgi duyan herkesi (kullanıcılar, yöneticiler, sponsorlar, vb.) belirleyin. Paydaşlar insan veya insan topluluğu olmalıdır.

a. Aktörler ve Hedefler

Sistemle doğrudan etkileşime girecek kişilerin veya cihazların rollerini, türlerini (başlatma veya katılma) ve başlatan aktörlerin hedeflerini belirleyin.

b. Kullanım Örnekleri

i. Sıradan Açıklama

Aklınıza gelebilecek tüm kullanım durumları için (Sistem Gereksinimlerinize göre), kısa veya sıradan bir metin açıklaması yazın. Her kullanım durumunun yanıtladığı gereksinimleri açıkça listeleyin.

ii. Kullanım Durumu Şeması(Use Case)

Kullanım senaryolarını tüm kullanım senaryolarıyla birlikte çizin. <<include>> ve <<extend>> gibi ilişkileri belirtin.

iii. İzlenebilirlik Matrisi

Sistem gereksinimlerinin kullanım durumlarınızla nasıl eşleştiğini gösterin. Kullanım durumlarınızın önceliklerini gözden geçirin. En yüksek önceliğe sahip kullanım örnekleri ilk tanıtım(demo-1) için hazırlanmalı ve planlanmalıdır.

iv. Mükemmelleştirilmiş Açıklama

En önemli birkaç kullanım durumunu seçin ve ayrıntılı ("mükemmelleştirilmiş") açıklama sağlayın. "En önemli" kullanım durumları izlenebilirlik matrisinizle belirtilir. Etkinlik akışlarınız, belirtilen kullanım senaryosunu başlatırken başlatıcı aktörün ("kullanıcı") gerçekleştirebileceği her eylemi adım adım göstermelidir.

c. Sistem Dizisi Diyagramları

Yukarıda seçilen birkaç önemli kullanım durumu için sistem sırası diyagramlarını çizin.

4. Kullanıcı Ara yüzü Özellikleri

(Not: Sisteminiz bazı formlar yazdırıyorsa veya periyodik raporlar oluşturunca, bu da kullanıcı ara yüzünün bir parçası olarak kabul edilir ve form / rapor formatı bu bölümde belirtilmelidir.)Kullanıcı ara yüzü yalnızca önceki bölümde açıklanan kullanım durumları için belirtilmelidir ("mükemmelleştirilmiş" kullanım durumları).

a. Ön Tasarım

Belirli bir kullanım durumunda, kullanıcının bilgileri nasıl girdiğini ve sonuçların ekranda nasıl görüldüğünü adım adım gösterin.

[Görüntü maketlerini](#) kullanın ve kullanıcının hangi alanlara girdiği ve hangi düğmelere bastığını tam olarak açıklayın. Kullanıcının izleyeceği gezinme yollarını açıklayın.

Video oyunu gibi grafik ağırlıklı bir uygulama geliştiriyorsanız, bu raporunuzun en önemli bölümlerinden biridir.

b. Kullanıcı Çaba Tahmini

Birkaç tipik kullanım senaryosu seçin ve etkinlik akışında ilerlerken, görevi gerçekleştirmek için gereken fare tıklaması ve / veya tuş vuruşu sayısını sayın ve rapor edin. Bunların hangi kısmı kullanıcı ara yüzü navigasyonuna ve büro veri girişine gidiyor?

Ek A: Kullanıcı Çaba Tahmini içindeki örneğe bakın.

- Proje Yönetimi (aşağıdaki Bölüm 7'de açıklanmıştır) ve Referanslar (aşağıdaki Bölüm 8'de açıklanmıştır)

Bölüm 3:

5. Domain Analizi

a. Domain Modeli

Domain modelini türetme işlemini gösterin ve ardından diyagramı çizin. Aşağıdakiler için metin açıklaması sağlayın:

- i. Kavram tanımları
- ii. İlişkilendirme tanımları
- iii. Özellik tanımları
- iv. İzlenebilirlik matrisi - kullanım durumlarınızın Domain konseptlerinizle nasıl eşlendiğini gösterin.

b. Sistem İşletim Sözleşmeleri

Sadece Bölüm 3.c 'de açıklanan mükemmelleştirilmiş kullanım durumlarının operasyonları için, Bölüm 3.d' de tanımlanan sistem operasyonları için sağlanmalıdır.

c. Matematiksel Model

Herhangi bir matematiksel model kullanıyor musunuz? Örneğin, hisse senedi fiyatı tahmini için istatistiksel bir model veya bir video oyununda rakamları canlandırmak için yörüngeleri hesaplayan geometrik bir model kullanabilirsiniz.

HAYIR, ise, bir sonraki öğeye atlayın;

EVET, ise, modelinizi tam olarak tanımlayın.

6. Kullanım senaryo noktalarına göre proje büyüklüğü tahmini.

Rapor-1 için, yalnızca bu raporda açıklanan kullanım örnekleri için projenizin kullanım senaryo puanlarına göre büyüklüğünü tahmin edin.

7. Çalışma Planı

Dönem sonuna kadar, 1 numaralı raporu gönderdikten sonra grubunuzun ne yapmayı planladığını açıklayın. Yol haritasını öngörülen dönüm noktaları ve bunları gerçekleştirmeyi planladığınız tarihlerle gösterin. Tabii ki, kısa vadeli (önümüzdeki birkaç hafta) için planlarınız gelecekte daha fazla ayrıntılı olmalıdır.

Tercihen, projenizi planlamak ve zamana bölmek için [Gantt şemalarını](#) kullanmalısınız (ayrıca kontrol edin: [Gantt şeması temel özellikleri: Nedir, faydaları ve alternatifleri](#)). ([Ayrıca ürün yol haritanızı](#) oluşturmayı düşünün.

Proje teklifinize ürün sahipliği açıklamasını ekleyin ve sorumlulukların dökümünü sağlayın: her ekip üyesinin şu anda yaptıklarını, yönetim ve koordinasyon faaliyetleri de dâhil olmak üzere gelecekte yapacaklarını belirtin.

8. Referanslar

Kaynaklar listesi, projede kullanılan ve ders kitabından gelmeyen herhangi bir materyalin kesin referanslarını ve URL'lerini içermelidir. Bir referans listelenmiş ancak ana metinde belirtilmemişse, kısaca ne şekilde kullanıldığını açıklayın.

Minimum 3 sayfalık **müşteri problem bildirimi** gereklidir (Bölüm 1.a). Sorun bildiriminizde, sistem geliştiricilerinin düşüncelerini ve analizlerini yansıtan özellikleri listelemek yerine müşterinin sisteminizden gelen gereksinimleri veya beklentileri açıklanmalıdır. Müşteri Sorun Bildirimi, geliştiriciden ziyade müşterinin bakış açısından yazılmalıdır. Kullanım örneği jargonu olmadan, projeniz için motivasyonu, ne tür bir problemi çözdüğünüzü ve bu sorunun mevcut uygulamada (önerilen sisteminiz devreye alınmadan önce) nasıl çözüldüğünü gayri resmi olarak açıklamalısınız. Müşterinin dilini ve terminolojisini, yani hedef kullanıcılarınıza tanıdık gelen dili kullanın ve teknik jargondan kaçın. Okuyucunun sorunu daha kolay anlamasını sağlamak için grafikleri, çizimleri ve ekran maketlerini kullanın. Daha fazla bilgi bulabileceğiniz veya mevcut çözümleri görebileceğiniz referanslar (kitaplar, makaleler, web siteleri) sağlayın.

Bölüm 2'deki numaralandırılmış gereksinimlerin, Müşteri Sorun Bildirimi'nin (Bölüm 1.a) açıklamasında belirtilen sistemin tüm özelliklerini yakaladığından emin olun. Eksik veya tamamlanmamış, numaralandırılmış gereksinimler listesi, gereksinimler ve kullanım senaryoları (yani [izlenebilirlik](#)) arasında eşleme yapılmasını zorlaştırır. Bu ihmal, sırayla, projenin gerçek kapsamını anlamayı zorlaştırır. Mükemmelleştirilmiş kullanım durumları ve müşteri gereksinimleri arasında açık bir ilişki göstermelisiniz. İlişkili kullanım örneklerinde belirli gereksinimlerin nasıl ele alındığını gösterin ve tartışın.

Gereksinimleriniz test edilebilir olmalıdır - gereksinimlerinizin müşteri tarafından kabul edilmek üzere nasıl test edileceğini göstermek için [kabul testi vakalarını](#) yazın.

Gereksinimler için öncelik ağırlıkları atarken, bir özelliğin algılanan öneminin, ona büyük önem verilmesi ve ilk olarak uygulanması gerektiği anlamına gelmediğini unutmayın. Örneğin, kullanıcı kimlik doğrulaması herhangi bir gerçek sistemde çok yüksek bir önceliğe sahiptir - genellikle sahip olması gereken bir özelliktir. Ancak, bu derste, kullanıcı kimlik doğrulaması için yeni bir yol bulmadıkça, sisteminizin kullanıcıyı doğru bir şekilde oturum açabileceğini ve kullanıcının sisteme kaydolmasına izin verebileceğini gösterdiğiniz için pek fazla puan alamazsınız. Başka bir deyişle, kullanıcı kimlik doğrulaması gerekli bir özellik olarak kabul edilir ve internette kolayca kopyalanabilen milyonlarca uygulama vardır. Bu yüzden kullandığınız kaynağı belirtin, ancak rapor sayfalarını muhtemelen bir yerden basitçe kopyaladığınız kullanıcı kimlik doğrulama

mekanini açıklamak için boşa harcamayın. (Elbette, sisteminizin kullanıcı kimlik doğrulaması konusunda yeni bir şey varsa, bu yeniliği raporunuzda tanımlamanız gerekir.)

Bu nedenle, **projenizin temel işlevsel özelliklerine en yüksek önceliği atamalısınız**. Destekleyici faaliyetler (oturum açma ve kullanıcı kaydı gibi), kendilerine hangi önceliğin verilmesi gerektiğini çok fazla düşünmeden uygulanmalıdır. **Sınıfınızdaki en etkileyici ve en rekabetçi projeyi gösterebilmeniz için daima en önemli, projenizle ilgili benzersiz ve yeni olan şeylere odaklanın!**

Bölüm 2.b) İşlevsel Olmayan Gereksinimler bölümünde, gereksinimlerinizin test edilebildiğinden emin olmalısınız. Restoran otomasyon projesi için iyi işlevsel olmayan gereksinimlere bazı örnekler:

- "Aşçının kullanıcı arayüzü, ana pencereden başlayarak 5'ten az tıklamayla bir sonraki yemek siparişini görüntülemeye izin vermelidir."
- "Aşçının kullanıcı arayüzü, garsona, ana pencereden başlayarak 3'ten tıklamayla yiyeceklerin hazır olduğunu bildirmeye izin vermelidir."
- "Sunucu programı o kadar güvenilir olmalı ki ayda bir kereden fazla çökmemelidir."
- "Aşçının kullanıcı arayüzü metin ve grafikleri göstermelidir, böylece ortalama görme yeteneği olan birisi 1 metreden okuyabilmelidir."

Yazılım uygulandığında gereksinimlerin test edilebilmesini sağlayan gereksinimlerdeki nicel değerlere dikkat edin.

Gerçek sayılar sadece örnektir ve makul değerlerin ne olduğuna karar vermek sizin görevinizdir. Hatta müşterinizden bazı gerçekçi olmayan gereksinimleri de listeleyebilirsiniz (örneğin, aşçının garsona yiyeceklerin ekranın tek bir dokunuşuyla hazır olduğunu bildirmesini isteyebilir) ve daha sonra kullanıcı arayüzünün analizini yaptığınızda (bölümünüzün 4. Bölümü) raporunda) veya alan analizinde (Bölüm 5), analizinizde müşterinin taleplerinin gerçekçi olmadığını gösterirsiniz!

Bölüm 2.b) Kullanıcı Ara yüzü Gereksinimlerinde (hayal ettiğiniz) müşterinizden aldığınız şeyi (müşterinizin kullanıcı ara yüzünün nasıl olması gerektiğini düşündüğünü) gösterin. Çoğunlukla metin, internetten indirilen bazı resimler veya hatta elle çizilmiş eskizler olacaktır.

Oluşturmak için büyük çaba harcadığınız ekran görüntülerini dahil etmeyin. Müşteri tarafından oluşturulan bu gereksinimlere dayanarak, güzel bir şekilde oluşturulmuş kullanıcı ara yüzü tasarımı üretmeye devam edersiniz ve kullanıcı ara yüzü spesifikasyonu ile ilgili Bölüm 4'te güzel ekran görüntüleri rapor edersiniz.

Yine, Bölüm 4 müşterinizin talebine yanıt olarak neler yarattığınızı göstermektedir! Bir gereksinim olarak müşteriden gerekli kullanıcı ara yüzünün tasarlanmış ekran görüntüsü gerekmeyecektir. Yapmanız gereken, müşterinizin neler sağlayacağını hayal etmektir - muhtemelen bazı metin açıklamaları ve elle çizilmiş eskizler. Ardından, müşteri gereksinimlerinize yanıt olarak güzel grafikler oluşturmaya ve bunları raporunuzun 4. Bölümüne dahil etmeye devam edeceksiniz.

Özellikle Bölüm 3'te ve rapor boyunca, tekrardan kaçının çünkü tekrarlar sıkıcıdır. Ayrıntılı bir kullanım örneği zaten sunulan bir kullanım örneğine çok benziyorsa, rapordan çıkarınız. Örneğin, yeni bir kullanım durumu neredeyse aynı şemaya sahip olur, ancak kullanım örneği adı veya şemadaki diğer birkaç ad gibi birkaç küçük değişiklikte birlikte atladığınız ayrıntılı kullanım durumlarını kısaca listelemeniz ve sunulan kullanım durumlarından hangisinin atlananlara benzer olduğunu söylemeniz gerekir. Gerçek dünya problemlerinde tekrarlayan kullanım durumlarına ihtiyaç duyulabilir (pek böyle bir durum olmasa da), ancak tekrarlama bu dersin amaçları için yararsız ve gereksizdir.

Tüm uygulamalar için kullanıcı çabası en aza indirilmelidir (Bölüm 4.b). Ancak, bu sorun bazı uygulamalarda diğerlerinden daha önemlidir. Bir video oyunu veya sanal laboratuvarlar geliştiriyorsanız, öncelikle kullanıcı ara yüzünüzün görsel görünümüne odaklanmalısınız. Bu tür uygulamalarda kullanıcı çabasını tahmin etmek ve en aza indirmek önemlidir ve daha sonra yapılmalıdır. (Tabii ki, her zaman istisnalar vardır - bazı video oyunlarında önce onları yapmanız önemlidir...)

Bölüm 5 Etki Alanı Analizi, etki alanı modelinin nasıl türetildiğinin açıklamasını içermelidir. Etki alanı modelinizi türetmek için yaptığınız analiz işlemini göstermelidir. Yalnızca alan modeli diyagramı gösterirseniz alacağınız puan azalır.

Bölüm 8 Kaynaklar, projeniz için kullandığınız TÜM kaynakları listelemelidir. Referanslarınızı IEEE referans stilinde listelemeniz önerilir. Tüm ekip üyeleri bu bölümün hazırlanmasında yer almalı ve proje için kullandıkları URL'lerin, kitapların veya diğer kaynakların bir listesini sağlamalıdır. Bölüm eksikse veya eksik olarak değerlendirilirse, rapor notundan 5 puana kadar düşülecektir.

Seçtiğiniz son tasarım seçimini bildirmek yeterli değildir- ayrıca hangi alternatifleri düşündüğünüzü ve kendi tasarımınızı seçmek için yaptığınız argümanların neler olduğunu da tartışmalısınız. Değerlendirme sürecinde açıklamaları yazmak yerine, alternatifleri ve argümanları önceden tartışarak notlandırma sürecinde birçok yanlış anlama önlenabilir.

Rapor için sayfa sayısı sınırlı olmasa da, daha kısa raporların anlaşılması ve değerlendirilmesi daha kolaydır. Rapor **açıklayıcı** olmalıdır; hoca projeyi anlamak ve değerlendirmek için başka bir materyale ihtiyaç duymamalıdır. Kullanım senaryosu senaryolarında ve etki alanı modelinde, iş ayrıntılarını, yani kullanıcının hizmeti almak için genellikle sistemle nasıl etkileşimde bulunduğunu sağlayın. Bu aşamada, Rapor-2'de ele alınacak planlanan yazılımınızın ayrıntılarından kaçının.

Kısalık iyidir, çünkü profesyonel görünümlü bir teknik rapor oluşturmanız daha kolay olacaktır - daha kısa raporlar için tutarlılığı, netliği ve okunabilirliği korumak daha kolaydır. Ayrıca şunu da kontrol edin: [Yazınızdaki kelimeliği azaltmanın 10'dan fazla yolu \(TechRepublic\)](#) - ifadelerinizi düzene koyduğunuzda, mesajınız daha güçlü ve net hale gelir.

Sadece UML diyagramlarını göstermeyin - **tüm şekiller, tablolar ve diyagramlar için açıklamalar yapın!** Ne yazık ki, diyagramlar, özellikle teknik diyagramlar, nadiren kendi kendini açıklar niteliktedir. Düşündüğünüz alternatif çözümleri ve son seçim argümanlarını belgelemelisiniz. Diyagramlar yalnızca nihai çözümünüzü temsil eder, ancak bu çözümlere neden karar verdiğinizi ve hangi alternatiflerin dikkate alındığını açıklamaz. Bu nedenle, tüm şemalara alternatiflerin ve değiş tokuşların açıklanması ve tartışılması eşlik etmelidir. İnceleyicinin kısmında belirsizlik veya yanlış anlaşılmaya yol açabilecek her şey

açıkça açıklanmalıdır. Açıklamalar önemli noktalarda vurgulanan düzyazı ve kilit argümanlarla yazılmalıdır.

Mümkün olduğunca çok grafik sağlayın- sadece UML diyagramları değil, aynı zamanda çabalarınızı anlamamızı ve değerlendirmemizi kolaylaştıracak her türlü diyagram ve çizimi kullanın. Grafikler her zaman yardımcı olur, muhtemelen “[bir resim bin kelimeye bedeldir](#)” demişsinizdir!

Sembollerinizin anlamını tanımladığınız sürece herhangi bir sembol gösterimi kullanabilirsiniz (UML gerekli değildir). UML kullanmanın avantajı, sembollerin zaten iyi tanımlanmış olması ve tanımların aranabileceği birçok kaynak olmasıdır.

Projeniz için bazı saha çalışması yaptıysanız, lütfen raporunuza Ek olarak ekleyin. Örneğin, garaj otomasyonu üzerinde çalışıyorsanız, yerel bir garaj operatörünü ziyaret etmiş ve karşılaştıkları sorunları ve projenizin uygulanabilirliğini tartışmış olabilirsiniz. Restoran otomasyonu üzerinde çalışıyorsanız, yerel restoran çalışanları ile görüşmeler yapmış olabilirsiniz. Sanal laboratuvarlar geliştiriyorsanız, öğrencilerle ve fakülte ile görüşmüş olabilirsiniz. Vb. Bu tartışmanın ayrıntılarını raporunuzun Ek kısmına ekleyin. (Röportajlar kesinlikle yazıya dökülmelidir, ancak bu gerekli değildir.)

Müşteri Sorun Bildiriminizde, “müşterinizden” toplanan bu bilgileri sisteminizin gereksinimlerine nasıl dahil ettiğinizi tartışın.

2.1 Bir şeyi elde etmenin en iyi yolu her şeyi denemek ve yapmaktır

Tipik bir acemi hatası, her şeyi aynı anda yapmaya çalışması ve hiçbir şey başaramamasıdır. Öğrenciler birçok şey belirler ve hepsini ayrıntılı olarak açıklamaya çalışırlar. Bu, çok deneyimli bir geliştirici için bile verilen zaman diliminde imkânsız olurdu! Projenizde odak noktası eksik olacak ve tüm kullanım örnekleri basit görünecektir.

Projenizin odağını tamamen daraltmanızı ve hırslarınızı azaltmanızı önermiyorum. Bu rapor iki bölümden oluşmaktadır:

- Mükemmelleştirilmiş Kullanım Durumlarının Tanımı: müşteri problem bildirimi, sistem gereksinimleri ve kullanım senaryolarının gündelik açıklaması

- Kullanım Durumlarının Mükemmelleştirilmiş Açıklaması ile başlayan ve kalan tüm bölümleri içeren “pratik” kısım

Raporunuzun “düşünsel” bölümünde, olabildiğince iddialı olmaya çalışmalısınız. Aklınıza gelebilecek tüm olası özellikleri ve uzantıları açıklayın.

Ardından, raporunuzun “pratik” bölümünde, pratik (!) Olmalı ve dar bir odak ile başlamalısınız. İşte tasarladığınız tüm seçeneklerden ne yapacağınız burada ortaya çıkacak.

İlk tanıtım için sunacağınız en önemli iki ila üç kullanım durumunu seçin ve yalnızca seçilen kullanım durumları için ayrıntılı (“mükemmelleştirilmiş”) açıklama sağlayın. Açıklamanıza özen gösterin! Tüm kullanım durumlarını ayrıntılı olarak belgelemekten kaçının. Tüm özet kullanım durumlarınızı ayrıntılı olarak açıklayabileceğinizi düşünseniz bile, bunu yapmaktan kaçının! Böyle bir güven genellikle probleminiz hakkında fazla bir şey bilmediğinizi gösterir. Kullanım durumlarınız sadece **yüzeysel** olarak ele alınacak ve hiçbirisi bir gerçeklik kontrolünden geçmeyecektir - hiçbirisi gerçek bir sistemde gerçek bir görevi yerine getiremez.

Sisteminizin birkaç temel özelliğine odaklanın ve inandırıcı ve gerçekçi olmalarına özen gösterin. Diğer kullanım durumlarınızı yalnız bırakın. Birçok özelliğe gelişigüzel değil, birkaç özelliğe dikkatlice odaklanırsanız, egzersizin amacını daha sıkı kavrayacaksınız.

ÖNEMLİ: Bu ders sırasında projenizin iki yinelemesi olduğunu unutmayın. İlk yinelemede her şeyi yapmaya çalışmayın - ikinci yinelemeye bir şeyler bırakın!

İlk tanıtımdan sonra, kalan kullanım örneklerinden bazılarını seçecek ve ikinci tanıtım için bunları ayrıntılı olarak açıklayacaksınız.

2.2 Ayrıca faydalı birkaç bilgiyi aşağıda bulabilirsiniz.

Yukarıdaki liste, bu proje raporu için sadece gerekli ana öğeleri vermektedir. İyi bir proje raporu hazırlayabilmek için sınıf ders notlarındaki ilgili materyalleri okumanız ve anlamanız kesinlikle gereklidir. Ayrıca kontrol edin [Gereksinim analizi @ Wikipedia](#).

Ayrıca, IEEE Yazılım Gereksinimleri Spesifikasyonları için Önerilen Uygulama (ekte) özel gereksinimlerin açıklaması için birkaç örnek anahat içermektedir: [IEEE Std 830-1993](#) (IEEE Std 828-1998 tarafından değiştirilmiştir).

Ayrıca yeni IEEE Yazılım Gereksinimleri için Önerilen Uygulamaya da bakın
Özellikler: [IEEE Std 830-1998](#) (IEEE Std 830-1993 Revizyonu)

3. Çaba Dağılımı

Tüm gruplar raporlarına çaba dökümünü gösteren bir tablo eklemelidir. Bu aktivitenin iki amacı vardır:

1. Ekip üyelerinin proje sorumluluklarını eşit olarak dağıtmalarına yardımcı olur,
2. Hocanın bireysel katkıları adil bir şekilde derecelendirmelerine yardımcı olur.

ÖNEMLİ NOT: Bu rapor yalnızca gereksinim analiziniz, kullanıcı arabirimi belirtimi ve etki alanı modellemesi hakkında bilgi içermelidir. Program kodunun yazılması ile ilgili herhangi bir faaliyeti rapor etmemelisiniz (yani sistem uygulaması).

Bu tür faaliyetler ilk tanıtımınızın bir parçası olarak rapor edilmelidir.

Uygulamaya odaklanan ve bu raporun hazırlanmasına daha az katkıda bulunan ekip üyeleri, çabalarının düzgün bir şekilde yansıtılması için bireysel katkıların yapay olarak değiştirilmesini istememelidir. Bunun yerine, proje tanıtımına daha fazla katkı talep etmelidirler.

ÖNEMLİ NOT: Çaba dökümü, raporun belirli bileşenine katkıda bulunan tüm faaliyetleri yansıtmalıdır. Bu tür faaliyetler şunları içerir:

- Raporun belirli bölümlerinin nasıl tasarlanacağıyla ilgili fikirleri beyin fırtınasıyla kararlaştırmak,
- Yerel restoranların ziyaret edilmesi ve restoran otomasyonunun nasıl tanıtılacağı, yerel park garajlarının nasıl ziyaret edileceği ve otopark garaj otomasyonunu tanıtırken karşılaşılan konular hakkında yönetim ile nasıl tartışıldığı, anlamak için finansal ders kitaplarının nasıl çalışılacağı hakkında personel ile röportaj gibi problem alanını anlamak için gerekli saha çalışması bir yatırım oyununun nasıl geliştirileceği, ders kitaplarının incelenmesi ve sanal öğretim laboratuvarlarının nasıl geliştirileceğini anlamak için öğrenciler ve fakülte ile konuşmak,
- Raporun belirli bir bölümünün metninin yazılması,

- Raporun belirli bölümlerinin taslağını okuma ve geri bildirim sağlama,
- Rapor yazılmasına katkıda bulunan diğer etkinlikler.

Başka bir deyişle, bu bölüm için tüm krediyi alması gereken kişi sadece metinleri yazan kişi değildir. Katkıda bulunan ve asıl yazıyı yapmayı mümkün kılan herkes, bir miktar kredi talep etmelidir.

Bu rapor için çaba dökümü tablosu formatı aşağıdaki gibidir. Ekibin beş üyesi olduğunu ve gösterim amacıyla bireysel sorumluluklarının bu sorumluluk matrisinde gösterildiği gibi kabul edildiğini varsayalım:

Not: Raporun tüm alt bölümleri, özellikle de en çok puanı taşıyan bölümler için sorumluluk matrisinde ek satırlar olmasını isteyebilirsiniz. Bu size sorumluluk tahsisi hakkında daha doğru bir fikir verecektir.

Her satırdaki değerler %100'e kadar eklenmelidir. Her sütundaki değerler aşağıdaki gibi bireysel ekip üyeleri için sorumluluk düzeylerini verecektir. Bireysel ekip üyelerinin yüzdelerini toplam puan sayısı ile çarparsak, her üyenin potansiyel olarak kaç puan kazanabileceğini bileceğiz. Örneğin, yukarıdaki örnekteki Üye-1 en fazla aşağıdakileri kazanabilir:

		Ekip Üyeleri				
Not: Puanlama durumları için buraya bakın.		Üye 1	Üye 2	Üye 3	Üye 4	Üye 5
Sorumluluk Seviyeleri	Proje Yönetimi (10 puan)	%70		%20		%10
	Bölüm 1: Müşteri Sorun Bildirimi (9 puan)	%50			%50	
	Bölüm 2: Sistem Gereksinimleri (6 puan)		%100			
	Bölüm 3: Fonksiyonel Gereksinimler (30 puan)		%67	%10	%23	
	Bölüm 4: Kullanıcı Arayüzü Özellikleri (15 puan)					%100
	Bölüm 5: Alan (Domain) Analizi (25 puan)			%25	%75	
	Bölüm 6: Çalışma Planı (5 puan)	%50				%50

$$0.7 \times 10 \text{ puan} + 0.5 \times 9 \text{ puan} + 0.5 \times \text{puan} = 14 \text{ puan}$$

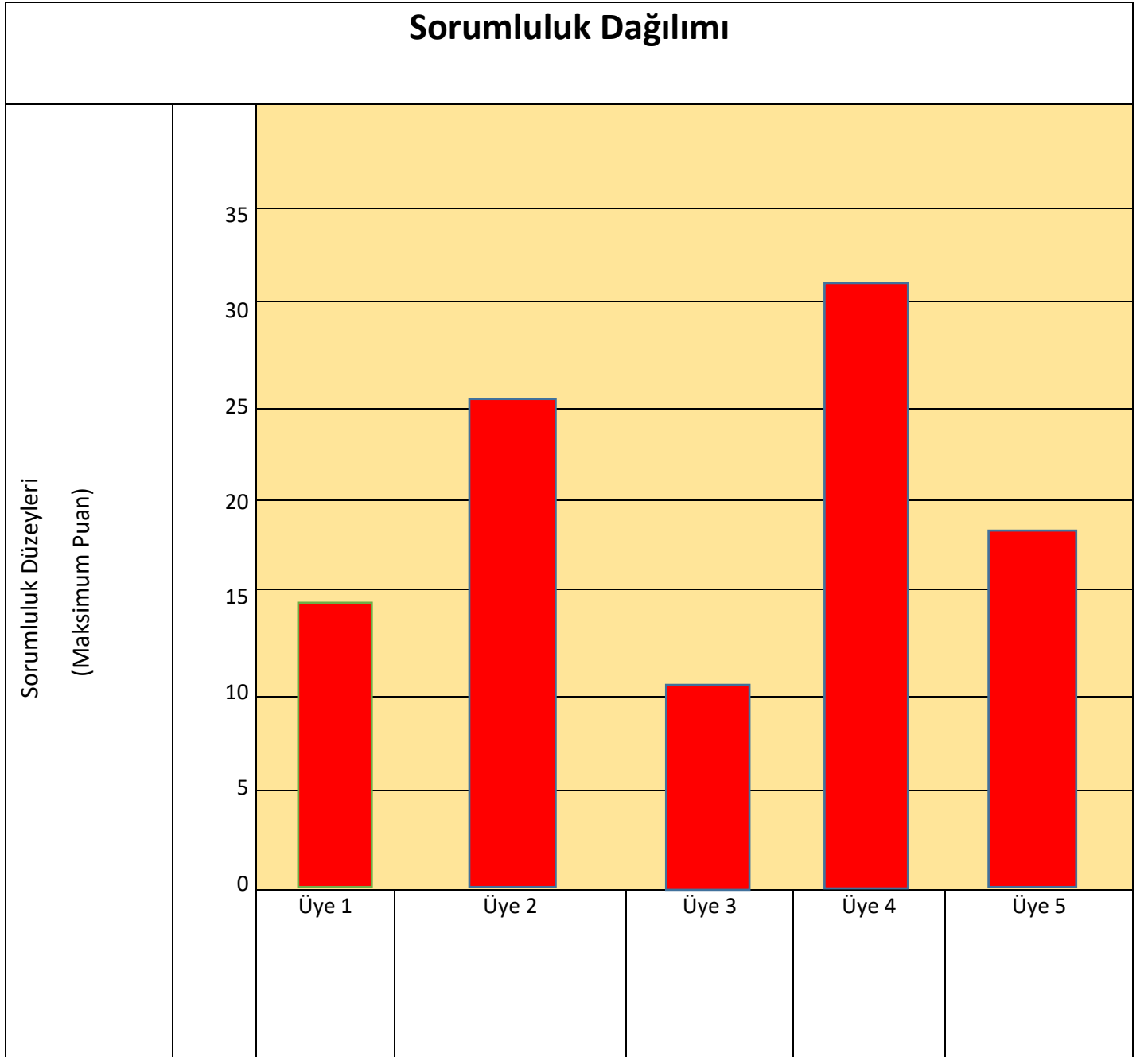
Tüm ekip üyeleri için aynı hesaplamayı tekrarlayarak, ekip boyunca aşağıdaki sorumluluk dağılımı çizelgesini elde edeceğiz:

$$(\text{Üye 2: } 1.0 \times 6 + 0.67 \times 30 = 26,1 \text{ puan})$$

$$\text{Üye 3: } 0.2 \times 10 + 0.1 \times 30 + 0.25 \times 25 = 11,25 \text{ puan}$$

$$\text{Üye 4: } 0.5 \times 9 + 0.23 \times 30 + 0.75 \times 25 = 30,15 \text{ puan}$$

$$\text{Üye 5: } 0.1 \times 10 + 1.0 \times 15 + 0.5 \times 5 = 18,5 \text{ puan})$$



Üyeler

Bu grafikten, bu örnek senaryoda sorumluluk dağılımının ekip genelinde iyi dengelenmediğini kolayca görebiliriz. İdeal olarak, tüm üyelerin yaklaşık eşit düzeyde sorumluluk alması gerekir. Bu nedenle, yukarıdaki senaryoda, Üye-1 ve 3 daha fazla sorumluluk üstlenmelidir. Bu bölümün başında açıklandığı gibi, “sorumluluk” sadece metnin gerçek yazımı ve diyagramların çizilmesi değil, rapor yazımına katkıda bulunan tüm faaliyetleri de içerir!

4. Rapor Hazırlığı

Rapor elektronik ortamda hazırlanmalıdır. Bu rapora UML diyagramları oluşturmak için bir yazılım aracı kullanmak **zorunludur**. Elle çizilmiş UML diyagramları kabul edilemez. UML diyagramı için yazılım araçları için internette arama yapabilir ve başka araçlar da bulabilirsiniz. UML sembollerini destekleyen herhangi bir araç kabul edilebilir.

Elle çizilmiş şekiller, kullanıcı ara yüzü gereksinimlerini göstermek için kabul edilebilir. Bu şekiller beyaz kâğıda siyah mürekkep kullanılarak çizilmeli, taranmalı ve bu belgeye görüntü olarak eklenmelidir. Diyagramlarınızın okunabilirliğini sağlamak sizin sorumluluğunuzdadır.

Raporun **tutarlılığı, anlaşılabilirliği ve doğruluğu** kritik öneme sahiptir. Rapor, yeterlilik ve okuyucunun anlaması için başka materyallere ihtiyaç duymaması açısından **eksiksiz** olmalıdır. Sınıftaki öğrencilerin kafasını karıştırabilecekleri ni tahmin edin ve olası her belirsizlik veya tutarsızlık kaynağını netleştirin. **Tüm ekip üyelerinin** tüm raporu dikkatlice okuduğundan ve içeriğini tamamen ve net bir şekilde anladığından emin olun.

Rapor herhangi bir okuyucuya tek bir kişi tarafından yazılmış gibi görünmelidir. Tabii ki, her ekip üyesi raporun farklı kısımlarına katkıda bulunacaktır, ancak herkes raporun tamamını okumalı ve açıklık ve bütünlük sağlamalı, tutarlı terminoloji kullanmalı ve farklı bölümlerdeki diyagramların birbiriyle tutarlı olmasını sağlamalıdır. Raporun tamamı, tek bir kişi tarafından yazılmış gibi görünmesi için aynı yazma stilini, dili vb. kullanmalıdır.

Rapor **profesyonel görünüme** sahip olmalıdır; açıkça etiketlenmiş bölüm başlıkları, şekil başlıkları, sayfalandırma ve dilbilgisi ve yazım hataları olmadan düzgün, okunması ve anlaşılması kolay olduğundan emin olun. Ayrıca, diyagramların ve görüntülerin basıldığında okunabilir olduğundan emin olun (yani harfler veya simgeler çok küçük ve okunaksız olmamalıdır). Görüntülerde ve diyagramlarda renkler kullanıyorsanız, siyah beyaz / gri tonlamalı yazdırıldıklarında bunların ayırt edilebilir olup olmadığını kontrol edin. Her şekle / tabloya metinde atıfta bulunulmalı ve uygun şekilde tanımlanmalıdır.

Raporunuzu, yalnızca sizin gerçek anlamını bildiğiniz kelimelerle yazmayın. Raporunuzu üçüncü bir kişinin bakış açısından yazın. Projenizle ilgili her şeyi biliyorsunuz, böylece içinde ne olduğunu anlamak için fazla bilgiye ihtiyacınız yok. Üçüncü bir kişinin yalnızca yazılım mühendisliği hakkında genel bilgisi vardır ve genel ilkelerin özel bağlamınızda nasıl uygulandığını anlamada yardıma ihtiyacı vardır.

5. Kodlamaya Ne Zaman Başlamalı?

Sisteminizin en azından bazı işlevlerini hemen uygulamaya başlamalısınız. Prototipler oluşturmak ve onlarla oynamak sisteminizi daha iyi belirlemenize ve tasarlamaya yardımcı olacaktır. Kodlama ile başlamak için Rapor-2'yi göndermeyi beklememelisiniz- ilk demoya fazla zaman kalmamış olacak!

Kullanıcı ara yüzü için **model ekranları** hakkında bir not.

Sisteminize model ekranları oluşturmaya başlamak için bir programlama dilinde programlamanız gerekmediğini unutmayın. Model ekranları fikri, bunları programlama dilinde değil PowerPoint veya başka bir grafik düzenleyicide oluşturmanızdır.

Ancak, bu rapordaki başlıca uygulamalar hakkında rapor vermeyin. Uygulama hakkında ilk raporunuzu ilk tanıtımda bekliyoruz.

6. Rapor Değerlendirmesi

Bu rapor ancak raporun tamamı gönderildikten sonra notlandırılacaktır.

Raporun bireysel bölümleri sunumdan hemen sonra notlandırılmaz.

Ancak, **puanların %50'si zamanında teslim edilmeyen kısımlardan düşülecektir.**

Zaman izin verirse, raporun tamamının tamamlanmasından önce tek tek parçalar hakkında geri bildirim sağlayacağım.

Rapor derecelendirme iki adımdan oluşur:

1. Bireysel rapor değerlendirilmesi
2. Raporların karşılaştırılması

Her adım daha sonra ayrıntılı olarak açıklanmaktadır. Genel takım notu ve bireysel notlamalar hakkında daha fazla bilgi için değerlendirme politikasına göz atın.

Tüm raporlar, aşağıdaki tabloya göre birbirinden **bağımsız** olarak derecelendirilecektir. Tabloda her bölüm için mümkün olan maksimum puan gösterilmektedir. İşinizin kalitesine bağlı olarak gerçek puanınız daha düşük olabilir.

Asıl puanlar değerlendirme kontrol listesine göre atanacaktır.

Bölüm	Müşteri Sorunu Bildirimi		Sistem Gereksinimleri			Fonksiyonel Gereksinimler					Kullanıcı Arayüzü		Domain Modeli				Çalışma planı	Kaynakça
	Sorun bildirimi	Sözlük	Fonksiyonel Gereksinimler	Fonksiyonel Olmayan Gereksinimler	Arayüz Gereksinimleri	Aktörlerin Hedefleri	Kullanım Durumu Basit Tanımları	Kullanım Durumu Diyagramları	Kullanım Durumu Tüm Tanımları	Sistem sırası diyagramları	Ön Tasarım	Çaba Tahmini	Kavramlar	Bağılantılar	Özellikler	Geliştirmeler		
Puanlar	5	4	2	2	2	2	8	5	10	5	11	4	10	5	5	5	5	(-5)
Toplam Puanlar	<p><u>Proje Yönetimi</u> içi 90 + 10 = 100 puan * Eksik veya tamamlanmamış kaynakçalar için 5 puana kadar düşülecektir</p>																	

ÖNEMLİ NOT: Ne yazık ki, yukarıdaki tabloyu “her projeye uyan” şekilde tasarlamak imkansızdır. Bu nedenle, bu dağılımı belirli bir projenin özelliklerini yansıtabilecek şekilde uyarlanabilen genel bir başlangıç noktası olarak düşünün. Bazı bileşenler belirli bir proje için daha önemli görülürse, puanlar bölümler arasında kayabilir. Bu sayfanın başında belirtildiği gibi, projenin sadece basit kullanım durumlarına değil, karmaşık algoritmalara ihtiyacı varsa, Bölüm 3 mükemmel şekilde yapılabilir, ancak maksimum 30 puandan daha azını taşıyacaktır. Bunun yerine, puanlar, kullanıcı arabiriminde Bölüm 4 veya etki alanı modelinde Bölüm 5 gibi, söz konusu proje için daha kritik görünen diğer bölümlere kaydırılacaktır. Projeniz için Matematiksel Model (Bölüm 5.c) veya algoritmaların açıklaması, kullanım örnekleri veya kullanıcı arabiriminden daha önemliyse, matematik modeli bölümü en fazla 15 puan alabilir. Vb.

Her ekip üyesine, rapora bildirilen katkılarına dayanarak puanı verilecektir.

Bu adımdan sonra öğrenci notları ayrıntılı olarak hesaplanacaktır.

7. Rapor Teslimi

Raporunuzun eksiksiz olduğundan ve grafiklerinizin okunabilir olduğundan emin olmalısınız (özellikle elle çizilmiş çizimler ekliyorsanız). Bunu sizin için kontrol etmeyeceğiz ve yeniden göndermenizi isteyeceğiz. Diyagramların okunaklı olmadığını tespit edersek, eksik olduğunu düşüneceğiz. Hazırlanan ancak yanlışlıkla atlanan rapor bölümlerinin takip sunumları kabul edilmeyecektir.

Bölüm 1'i teslim ederken, belgenin aşağıdaki bölümleri içermesi gerekir:

- Kapak Sayfası ve Bireysel Katkı Dağılımı
- İçindekiler
- Bölüm 1: Müşteri Sorun Bildirimi
- Bölüm 2: Sistem Gereksinimleri
- Proje Yönetimi
- Kaynakça

Bölüm 2'yi teslim ederken, belgenin aşağıdaki bölümleri içermesi gerekir:

- Kapak Sayfası ve Bireysel Katkı Dağılımı
- İçindekiler
- Bölüm 1: Müşteri Sorun Bildirimi- gerektiği gibi düzenlenmiş halde
- Bölüm 2: Sistem Gereksinimleri- gerektiği gibi düzenlenmiş halde

- Bölüm 3: İşlevsel Gereksinimlerin Belirtimi
- Bölüm 4: Kullanıcı Ara yüzü Özellikleri
- Proje Yönetimi
- Kaynakça

Tam rapor, Rapor Formatında belirtilen **tüm bölümleri** içermelidir (daha önce kısmi raporlarda gönderilen bölümler gerektiğinde düzenlenebilir).

Her grup raporlarını (**her grup için tek bir PDF belgesi !!!**) teslim tarihinde veya öncesinde sunmalıdır.

Tüm grupların son teslim tarihine uymaları ve sunum formatı ile ilgili talimatları takip etmeleri gerekmektedir.

Son teslim tarihi, son gün gece yarısına kadardır.
