Bahar yarıyılı Yazılım Gereksinimler ve Analizi dersi ile ilgili paylaşımlar bu kısımdan yapılacaktır. Bu dersi alan öğrenciler önerilen dönem projelerinden birini seçip belirtilen tarihe kadar tamamlamak zorundadırlar.
Raporlanan ödevler Final Sınav haftasında teslim edilecektir.
Raporun değerlendirilmesinde rapor içeriğinin kaliteli ve yeterli olması ve içeriğinin benzerliğine göre puan verilecektir.
Yazılım Gereksinimleri ve Analizi dersi 3+2 lik bir derstir. Yani dersin laboratuvarı da bulunmaktadır. Bu sebeple laboratuvar ortamında belirli küçük çalışmalar yapılıp laboratuvar sorumlusu hocaya teslim edilecektir. Projelerin teslimi de benzer şekilde yapılacaktır. Dersle ilgili duyuruların izlenmesi önemlidir.
YAZILIM GEREKSİNİMLERİ VE ANALİZİ DERSİ DERS İZLENCESİ

Bölüm			**Öğretim Yılı**	**Tarih**	
Yazılım Mühendisliği (Teknoloji Fak.)			2024-2025	28.02.2025	
Ders Kodu	**Ders Adı**		**Dönem/Yıl**	**AKTS Kredisi**	
YMH212	Yazılım Gereksinimleri ve Analizi		Bahar / 2.Sınıf	4	
Ders Dili	Türkçe				
Durumu	Zorunlu				
Ön şartlar	Yok				
Dersin Adresi					
Kredi	**Teori**	**Uygulama**	**Laboratuar**	**Sunum**	**Proje/Alan Çalışması**
4	3	2	-	-	-
Öğretim Üyesi	Doç. Dr. Muhammet BAYKARA				
Ders Yardımcısı	Arş. Gör. Simge YILDIRIM				

Ders İçeriği	Yazılım Yaşam Döngüsünün kavranması. 12207 Standartı ve bağlantılı yorumlar. Yazılım yaşam döngüsü içerisinde gereksinim mühendisliği. Gereksinim çıkartımı ve modellemesi: sorunlar ve teknikler. Gereksinimlerin dökümanlaştırılması ve yönetimi. Standartlar ve CASE araçları. Bilişsel ve sosyo-organizasyonel konular. Farklı konsept projelerde Gereksinim Mühendisliği Aktivitelerinin Gerçekleştirilmesi. Use case'ler, User Storyler, Kısıtlar, Arayüzler ile Gereksinim Analizi Dökümanlarının

oluşturulması. Beyin Fırtınası ve Fikir Çimlenmesi ile, Gereksinimlerin belirlenmesi ve Doğrulanması.

Ders Planı	
Hafta	**Konular**
1	Giriş
2	Temel Kavramlar. Yazılım Mühendisliği ve Yazılım Yaşam Döngüsü ile ilgili Gereksinim Mühendisliği
3	Gereksinim Mühendisliği Temeller
4	Gereksinim Çıkarımı; Sorunlar
5	Gereksinim Çıkarımı; Teknikler
6	Gereksinim Değerlendirilmesi
7	Gereksinim Tanımlaması ve Dokümantasyonu
8	ARA SINAV
9	Gereksinimler Kalite Güvencesi - I (Kullanılabilirlik, Güvenlik)
10	Gereksinimler Kalite Güvencesi - II (Performans, Sürdürülebilirlik)
11	Gereksinim Devamlılığı
12	Gereksinim Yönetimi: Bakım, Kontrol ve Doğrulama
13	Sosyo Örgütsel ve Bilişsel Faktörlerin Gözden Geçirilmesi
14	Öğrenci Sunumları
15	FİNAL SINAVI

Değerlendirme Ölçütleri	**Adet**	Adet	**Yüzde (%)**
	Ara Sınavlar	1	40
	Kısa Sınavlar	-	-
	Ödevler	-	-
	Projeler	1	
	Dönem Ödevi	-	-
	Laboratuar	-	-
	Diğer	-	-
	Dönem Sonu Sınavı	1	60

	Arasınav:Sınav
Değerlendirme Ölçütleri Hakkında	Final :Sınav
	Sınıf İçi Yazılım Analizleri: Anlatım

İçerik Ağırlıkları Yüzdesi (%)	**Matematik ve Temel Bilimler**	20
	Mühendislik Bilimleri	60
	Mühendislik Tasarımı	20
	Sosyal Bilimler	-

Ders Çıktıları (Kazanımlar)	· Formal ve informal notasyonlar çerçevesinde fonksiyonel ve fonksiyonel olmayan gereksinimlerin açıklamalarını hazırlamak. SRS, BRD kavramlarını yorumlayabilmek. · Yazılım sistemlerinin analizlerini ortaya çıkartıp mevcut sistemleri analiz edebilmek. Önerilen sistemi tasarlayabilmek. Yapay zeka araçları ile projelerin front end backend geliştirmelerini yapıp test edebilmek. • Paydaşların ihtiyaçlarını ortaya çıkarmak için uygun yöntemleri kullanmak. • İhtiyaçlar, Kısıtlar, Eksiklikler, Gereksinimler, Arayüzler bağlamında ihtiyaçların analiz edilmesi.
Dersin Hedefleri	Bu dersin amacı öğrenciye yazılım mühendisliği süreçleri doğrultusunda gereksinim mühendisliği anlayışını ve bu hususdaki rolünü kazandırmaktır.
Dersin İşleniş Biçimi	Anlatım, tartışma, soru-cevap, örnek çalışmaları inceleme, mevcut sistem analizi ve önerilen sistem tasarımı, problem çözme, gereksinim analizi dökümanı yazma.

Dersin program çıktıları ile olan ilişkisi

Program çıktıları		1	2	3
1	Matematik, Fen bilimleri ve mühendislik temel bilgilerinin Yazılım			X

	Mühendisliği problemlerine uygulayabilme becerisi			
2	Deney tasarlama, deney yapma ve deney sonuçlarını analiz etme ve yorumlama becerisi			X
3	İstenen gereksinimleri karşılayacak biçimde bir sistemi, parçayı ya da süreci tasarlama becerisi			X
4	Tek ve çok disiplinli takım çalışması yürütme becerisi			X
5	Mühendislik problemlerini belirleme, tanımlama, tasarlama, analiz etme ve çözme becerisi			X
6	Mesleki ve etik sorumluluk bilinci		X	
7	Sözlü ve yazılı etkin iletişim kurma becerisi		X	
8	Mühendislik çözümlerinin, evrensel ve toplumsal boyutlardaki etkilerini anlamak için gerekli genişlikte eğitim			X
9	Öğrenmenin sürekliliğinin önemini benimsemiş olarak, bilim ve teknolojideki gelişmeleri izleyerek kendisini sürekli yenileme becerisi		X	
10	Mühendislik uygulamaları için gerekli teknikleri, teknolojileri ve modern araçları kullanma becerisi		X	
11	Mühendislik problemlerinin çözümlenmesinde sağlık, güvenlik ve çevre üzerinde yaratacağı ulusal ve uluslar arası etkilere duyarlılık			X
12	Özgüveni yüksek ve kendi kendine öğrenebilme becerisi		X	
Dersin Katkısı: 1:Hiç 2:Kısmi 3:Tümüyle				

Düzenleyen Kişi(ler): Doç Dr. Muhammet BAYKARA
Bu ders yazılım yaşam döngüsündeki aşağıda verilen beşli çekirdek süreçten ilk iki fazla daha fazla ilişkilidir. Bu sebeple bu derste öğrencilerden beklenen şey "iş analistliği", "yerinde inceleme ve analiz" ile farklı türden yazılımların gereksinimlerini ortaya çıkarabilmektir.
planlama çözümleme tasarım gerçekleştirme test&bakım
Bu dersi amacına uygun olarak yazılım mühendisliğinin temelleri dersinden ayırabilmek için sadece gereksinimlerin belirlenmesi, en çok kullanılan yazılımların analiz edilmesi, çeşitli tekniklerle gereksinim tablolarının çıkarılması gibi konular üzerinde durulacaktır. Ama bu ders

kapsamında tekrar anlatılmamış olsa dahi yazılım mühendisliğinin temelleri dersinde görülen konuları öğrencilerin bildiği kabul edilecektir. Bu konuda herhangi bir eksiği olan "Yazılım Mühendisliğinin Temelleri" dersi ile ilgili paylaşımlarıma bakabilir.

Dersimiz kapsamında bu dönem öğrencilerimizden ekipler halinde incelemesini beklediğimiz, öğrencilerimizin ileride çalışabilecekleri bazı proje başlıkları aşağıda verilmiştir. Bu başlıklarda çalışacak ekiplerimizi sınıf mevcutlarımıza göre, derste konuşulduğu gibi belirmemiz gerekmektedir. Bu hususta dersimizin laboratuvar sorumlusuna başvurup belirleme yapmanız gerekmektedir.

Yazılım Gereksinimleri ve Analizi Dersi Proje Başlıkları: (2025 yılı için gerçekleştirilmesi önerilen projeler ayrıca duyurulmuştur. Burada bu isimler önceki yıllardaki önerilen konseptler olarak bilgi amaçlı verilmiştir.)

1-ERP (Kurumsal Kaynak Planlama)
2-CRM (Müşteri İlişkileri Yönetimi&Esnek Talep Yönetimi)
3-OBS (Öğrenci Bilgi Sistemi & Eğitim Kurumu Bilgi Sistemi)
4-IoT (Akıllı Şehir & Giyilebilir Teknolojiler)
5-SIEM (Security Information Event Management)
6-Mobil Uygulamalar (Getir & Yemek Sepeti & Beyaz Masa)
7-Bulut Bilişim (IaaS, PaaS, SaaS), Büyük Veri,
8-EBYS / EDYS (Elektronik Belge Yönetim Sistemi)
9-E Ticaret / Finans
10-Sosyal Medya
11- E devlet?

Derste de bahsedildiği gibi burada verilen proje başlıkları laboratuvar sorumlusu ile görüşülüp somutlaştırılacak ve ekiplerimizin çalışacakları konu net olarak belirlenmiş olacaktır. Ekiplerimiz nisan ayında belirlenen hafta ile birlikte yaptıkları analizleri yine verilecek başlıklar ile sunmaya başlayacaklardır.

Bu konulara **dersin amaç ve içeriğine uygun**, yazılım yaşam döngüsündeki her fazındaki aktivitelerinin belirgin/tanımlı olduğu proje önerileri ders sorumlularına danışarak ek yapılabilir. Ancak dersteki anlatımlarda da görüldüğü üzere bu proje alanları uygun olarak belirlenmiştir.

Bu dersle ilgili temel anlatımlara gereksinimlerin belirlenmesi ile başlanacağından anlatımlardan önce aşağıda verilen sorularla ilgili herhangi bir problem kalmamalıdır.

Yazılım nedir? Yazılım yaşam döngüsü nedir?
Yazılım yaşam döngüsündeki aşamalar nelerdir?
Yazılım geliştirmede her bir faz ne anlama gelir?
Yazılım Mühendisliği nedir?
Yazılım Süreç Modelleri nelerdir?

Proje nedir? Proje fikir ilişkisi? Her fikir bir proje belirtir mi?

Bu sorularla alakalı olarak yararlanabileceğiniz temel kaynaklar şunlardır;

YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ GİRİŞ DÖKÜMANI

YAZILIM GEREKSİNİMLERİ VE ANALİZİ DERSİ KAPSAMINDA KULLANILABİLECEK PROJE DÖKÜMANTASYONU- ŞABLON VE İÇERİK TANIMLARI:

Yazılım Gereksinimleri Tanım Belgesi (YGTB/SRS)

Yazılım geliştirme süreci ;

- *Gereksinim Analizi / Requirements (Requirements Analysis)*
- *Spesifikasyon / Specification (Functional Specification)*
- *Mimari / Architecture (Software Architecture)*
- *Dizayn / Design (Software Design)*
- *İmplementasyon (Programlama) / Implementation (Computer Programming)*
- *Test / Testing (Software Testing)*
- *Sahaya Yerleştirme / Deployment (Software Deployment)*
- *Bakım / Maintenance (Software Maintenance)*

aşamalarından oluşmaktadır.

*YGTB, belgesinde geliştirilecek olan yazılımın işlevsel ve işlevsel olmayan gereksinimleri tanımlanır. **Use-Case**'ler, Kullanıcı **arayüz**leri, **User Story**'ler, **Kısıtlar**, **Mevcut Sistem**in Analizi, **Önerilen Sistem**in Tasarımı ve **ER** diyagramları verilir.*

Yazılım mühendisliği kapsamında YGTB için verilen verilen taslak şu şekildeydi:

Yazılım Gereksinimleri Tanımı

1. **Giriş**

1.1 Amaç

[Belgenin amacını ve hedef kitlesini tanımlayın.] Sorun Analizi ve Hedef Analizi mutlaka yazılmalıdır.

1.2 Kapsam

[Belgenin kapsamını tanımlayın.

Bu kısımda tanımlanan kapsam, varsa bir üst seviye gereksinimlerde yer alan benzer ifadelerle (örneğin, Vizyon belgesi) tutarlı olmalıdır.]

1.3 Tanımlar ve Kısaltmalar

[Belgeyi anlamaya yardımcı olacak ve alana özel terim ve kavramın tanımlarını verin ve belge içinde kullanılan kısaltmaları alfabetik olarak listeleyin.]

1.4 Referanslar

[Bu belgenin referans aldığı tüm kaynakları; referans numarası, referans kodu, başlığı, sürümü ve tarihi ile birlikte listeleyin.

Bu belge şablonu için, "IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications" referans alınmıştır.]

1.5 Dokümana Genel Bakış

[Belgenin sonraki bölümlerinde neler anlatıldığını özetleyin. Belgenin kullanımı ile ilgili (varsa) güvenlik ve gizlilik koşullarından bahsedebilirsiniz.]

2. Genel Tanım

[Yazılım ürünü ve gereksinimlerini etkileyen genel faktörleri, izleyen başlıklar altında tanımlayın.]

2.1 Ürüne Bakış

[Bu belge kapsamında tanımlanan yazılım ürününün (varsa) diğer ürünlerle ilişkisini anlatın.

Yazılım ürünü bağımsızsa ve sadece kendini içeriyorsa belirtin. Eğer bu belge, daha büyük bir sisteminin parçası olan yazılım ürününü tanımlıyorsa, üst seviye bir bakışla yazılım ile parçası olduğu sistem arasında ilişkiyi kurun ve sistem ile diğer yazılımlar arasındaki arayüzleri belirtin.

Sistemin ana parçalarını, birbirleriyle olan ilişkilerini ve dış arayüzleri gösteren bir blok diyagram çizebilirsiniz.]

[Mevcut Sistemin İncelenmesini bu kısımda verebilirsiniz]

2.1.1 Sistem Arayüzleri

[Yazılım ile diğer sistemler arasındaki arayüzleri ve bu arayüz gereksinimlerinin karşılanması için sağlanması gereken işlevleri belirtin.]

2.1.2 Kullanıcı Arayüzleri

[Yazılım gereksinimlerinin karşılayacak, yazılım ile kullanıcısı arasındaki arayüzleri belirtin. Bu arayüzler için taslak ekran yapılarını, sayfa/ekran yapılarını, menü veya raporların içeriklerini tanımlayın (eklere de referans edebilirsiniz)].

2.1.3 Donanım Arayüzleri

[Yazılım ile sistemin donanım öğeleri arasındaki arayüzlerin mantıksal özelliklerini belirtin. Bu özelliklerin konfigürasyon tanımlarını verin.]

2.1.4 Yazılım Arayüzleri

[Kullanılacak olan diğer yazılımın ürünlerinin (veri yönetimi sistemi, işletim sistemi, modelleme araçları, vs) ve diğer uygulama sistemleri ile (varsa) arayüzlerinin özelliklerini belirtin. Her bir uygulama yazılımı için, kod, isim, tanım/sürüm no.ve kaynak bilgilerini belirtin.]

2.1.5 İletişim Arayüzleri

[İletişim ile ilgili arayüzler (varsa) belirtilir (protokoller vb.)]

2.1.6 Bellek Kısıtları

[Birincil ve ikincil bellek ile ilgili özellikleri ve kısıtları belirtin.]

2.1.7 Ürünün İşletimi

[Kullanıcı gereksinimi olarak yazılımın normal ve özel çalışma kiplerini, kullanım ile ilgili zamanlama kısıtlarını, yedekleme ve kurtarma kısıtlarını veya gereksinimlerini belirtin.]

2.1.8 Saha Uyumlama Gereksinimleri

[Ürünün alandaki işletimi için, yükleme sahasını uyumlama gereksinimlerini (örneğin, veri veya başlatma sırası gereksinimleri) belirtin.]

2.2 Ürün İşlevleri

[Yazılımın gerçekleştireceği ana işlevlerin, detaya girmeksizin bir özetini verin.

Bu bölüm için gereken ana işlevlerin özeti, daha üst seviye gereksinimlerin (örneğin, Vizyon belgesi) söz konusu yazılıma atanmış bölümlerinden alınabilir, ancak bu belgenin "3. Özel gereksinimler" kısmı ile de tutarlı olmalıdır.

İşlevleri, bu belgeyi ilk defa okuyacak herhangi birinin kolaylıkla anlayabilmesi için listelenmiş şekilde düzenleyebilirsiniz.

Farklı işlevleri ve aralarındaki ilişkileri gösterebilmek için metinsel ya da grafiksel gösterim kullanılabilir. Söz konusu diyagramlar ürünün tasarımını göstermez, sadece işlevler arasındaki mantıksal ilişkiyi ifade eder.]

2.3 Kullanıcı Özellikleri

[Kullanıcıların sahip olmaları gereken genel özellikleri (eğitim seviyesi, deneyim vb.) belirtin.]

2.4 Kısıtlar

[Geliştiriciyi kısıtlayabilecek herhangi bir öğe ile ilgili genel tanımları verin (örneğin; donanım kısıtları, uyulması beklenen yönergeler, geliştirme ortamı ve teknolojisi ile ilgili gereksinimler, güvenilirlik gereksinimleri, güvenlik ve gizlilik hususları vb.)]

2.5 Varsayımlar ve Bağımlılıklar

[Belgede belirtilen yazılım gereksinimlerini etkileyebilecek olan faktörleri listeleyin. Varsayım ve bağımlılıklar tasarım kısıtları değil, değişmeleri halinde gereksinimleri etkileyecek olan faktörlerdir.]

3. Özel Gereksinimler

[Bu kısımda,geliştirme ve test etkinliklerini mümkün kılacak detay seviyesinde, tüm yazılım gereksinimlerini tanımlayın. Tanımlanan gereksinimler, tüm paydaşlar tarafından algılanabilir olmalıdır.

Gereksinimler, bütün sistem girdilerinin ve çıktılarının tanımı ile girdileri çıktılara dönüştüren işlevlerin tanımını içermelidir. Bunlarla ilişkili ve bunlara ek olarak, işlevsel olmayan (kalite) gereksinimlerini de tanımlayın.

Gereksinimler doğru, kesin, tam, tutarlı, doğrulanabilir, değiştirilebilir, izlenebilir ve önceliklendirilmiş olmalıdır.

Bütün gereksinimleri biricik (“unique”) olarak numaralandırın. GEREKSİNİM TANIMLARI İÇİN AYRICA OLUŞTURULMUŞ BİR TABLO PAYLAŞILACAKTIR. VE TÜM EKİPLERİN O TABLOYU DOLDURMALARI İSTENECEKTİR.]

3.1 Harici Arayüz Gereksinimleri

[Bütün sistem girdilerinin ve çıktılarının detaylı tanımını verin. Bu tanımlar 2. kısımda sunulan bilgileri tamamlamalı / detaylandırmalı, orada verilen bilgilerin tekrarı olmamalıdır.

Bir sonraki bölümde tanımlanması istenen işlevsel gereksinimlerin işletilmesini destekleyecek kullanıcı arayüzlerine ilişkin detaylar bu bölümde tanımlanabilir. Kullanıcı arayüzü tanımlama için bir şablon Ek-A’da verilmiştir.]

3.2 İşlevsel gereksinimler

[Ürüne ait işlevsel gereksinimlerin detaylı tanımını verin.

Use-case esaslı gereksinim analizi yaptıysanız bu bölümde use-case modelini ve bu modeldeki öğeleri detaylı olarak tanımlayın. Seçtiğiniz use-case’lere ait senaryoları UML Etkinlik Diyagramları veya metin biçimde detaylandırabilirsiniz. Use-case’lerin metin biçimde detaylandırılması için bir şablon, Ek-B’de verilmiştir.]

3.3 Performans Gereksinimleri
[Yazılım ve/veya kullanıcı arayüzü ile ilgili statik ve dinamik, nicel performans gereksinimlerini tanımlayın. Statik gereksinimlere; desteklenecek eşzamanlı kullanıcı sayısı, desteklenecek istemci sayısı vb. örnek verilebilir. Bu tür gereksinimler ölçülebilir şekilde ifade edilmelidir.
Benzer özellikler: Hız (“speed”), saniyedeki işlem sayısı (“processed transactions/second”), ortalama maksimum cevap süresi (“average, maximum response time”), ekran tazeleme süresi (“screen refresh time”), işlem hacmi (“throughput”), kurtarma/kendine gelme süresi (“recovery time”), kaynak kullanımı (“resource usage”) – hafıza, disk vb., eş zamanlı desteklediği kullanıcı sayısı (“number of simultaneous users to be supported”), kapasite (“capacity”) – sistemin sağlayabileceği müşteri ya da işlem sayısı]
3.4 Mantıksal Veritabanı Gereksinimleri
<Veritabanında tutulacak veriler ile ilgili mantıksal gereksinimleri tanımlayın. Veri varlıklarını ve aralarındaki ilişki ve bütünlük kısıtlarını belirtin. Bu gereksinimleri ifade etmek için E/R Diyagramı kullanabilirsiniz.]
3.5 Tasarım Kısıtları
[Standartlar, kullanıcı istekleri veya donanım kısıtlarına dayanan ve tasarımı gerçekleştirirken uyulması gereken kararları tanımlayın.]
3.6 Kalite Özellikleri
[Yazılım kalite özelliklerini tanımlayın.
Aşağıda bazı özellikler için açıklamalar ve örnekler verilmiştir.]
3.6.1 Güvenilirlik (“Reliability”)
[Yazılımın güvenilirliği yani yazılımın sunduğu servisin devam etmesi ile ilgili özelliktir.
Sistemin çökme sıklığı ve şiddeti, iki çökme arasında geçen ortalama süre (“MTTF: mean time to failure”), onarım için geçen ortalama süre (“MTTR: mean time to repair”), hata ya da kusur oranı (“bugs or defect rate”), maksimum hata ya da kusur oranı (maximum bugs or defect rate”) gibi ölçevler tanımlayarak ifade edilebilir.]
3.6.2 Kullanılırlık (“Availability”)
[Yazılımın kullanıma hazır olması ile ilgili özelliğidir.
Yazılımın kullanım için uygun olma yüzdesi (“% of time available=MTTF/(MTTF+MTTR)”) ile ifade edilebilir.]

3.6.3 Güvenlik

[Yazılımın kazayla ya da kötü niyetle erişimine, kullanımına, değiştirilmesine, tahrip edilmesine ya da imhasına engel olmak için taşıması gereken özellikleridir.

Belirli şifreleme ve kriptografi ile ilgili tekniklerinin kullanımı, amaca özel kayıtların ("log") ya da geçmiş verilerin tutulması, farklı modüllere/kullanıcılara belirli fonksiyonların atanması, programın bazı bölümleri arasında iletişimin kısıtlanması ve kritik değişkenler için veri bütünlüğünün kontrolü ile ifade edilebilir.]

3.6.4 Bakım-yapılabilirlik ("Maintainability")

[Yazılımın kullanıma alındıktan sonra kolay bakım-yapılabilmesi için taşıması istenen özellikler belirtilir.]

3.6.5 Taşınabilirlik ("Portability")

[Yazılımın başka platformlara (işletim sistemi, vb.) taşınabilirliği ile ilgili özellikler belirtilir.]

3.6.6 Kullanılabilirlik ("Usability")

<Yazılımın kolay kullanılabilirliği ile ilgili özellikler belirtilir.

Uyulması gereken standartlar (kullanıcı arayüzü standartları, kullanıcı arayüzlerinin tutarlılığı, vb), çevrimiçi ("online") ve içerik duyarlı ("context-sensitive") yardım, sihirbazlar ("wizards"), kullanıcı dokümantasyonu, eğitim materyalleri ve eğitim süresi gerekleri tanımlanarak ifade edilebilir.]

4. Gereksinimlerin Önceliği ve Kritikliği

[Bu belgede tanımlanan gereksinimlerin göreli önemlerini, önceliklerini ve varsa tanımlanmış ağırlıklarını yazın.]

5. Gereksinimlerin İzlenebilirliği

[Bu belgenin "3. Özel Gereksinimler" başlığı altında tanımlanan yazılım gereksinimleri ile bir üst seviye gereksinimler (örneğin, Vizyon belgesi) arasında çift yönlü izlenebilirliği oluşturun.

İzlenebilirliği tablo yapısında oluşturabilirsiniz.]

6. Ekler

[Bu belgenin yapısını basitleştirmek ve anlaşılmasını kolaylaştırmak için,içerikteki bazı bilgileri eklerde verebilirsiniz. Ana döküman içerisinde bilginin normalde sunulduğu yerde verilemeyen tanımlar için bu bölüm kullanılır.

Ekler, alfabetik olarak ve belge içinde referans edildiği sırada tanımlanmalıdır.]

BONUS ÖDEV

Ödevler haftalık verilecektir. Web sayfamdan verilecek bu ödevlerin takibi ve zamanında yapılması kanaat değerlendirmesi açısından önemlidir.

BAZI UML VE DİYAGRAM ÇİZİM ARAÇLARI

Bu kısımda özellikle tümleşik modelleme dili, proje yönetim, çevik yazılım ve diğer metodolojilerde kullanılabilecek bazı çizim araçları ve bağlantıları verilmiştir.

ARGO UML;

https://argouml-tigris-org.github.io/tigris/argouml/

ALTOVA U MODEL;

https://www.altova.com/umodel.html

VISUAL PARADIGM

https://www.visual-paradigm.com/download/community.jsp

IBM RATIONAL SOFTWARE ARCHITECT

https://www.ibm.com/developerworks/downloads/r/architect/

Online Diagram Software

GEN MY MODEL

https://www.genmymodel.com/

Microsoft Visio

https://indir.firat.edu.tr

BAZI PROJE YÖNETİM ARAÇLARI

REDMINE ;

http://www.redmine.org/

Microsoft Project Professional
https://indir.firat.edu.tr/
Taiga
https://taiga.io/
JIRA :
Jira Platformu: Jira Software, Bitpocket, Sourcetree, Bamboo, Clover, Fisheye, Crucible, Jira Service Dest, Jira Core, Confluence, Hipchat
https://www.atlassian.com/try
Versiyon Kontrol Sistemleri
Subversion
https://subversion.apache.org/
KULLANILABİLECEK BAZI ARAÇLAR

1. Gereksinim Toplama ve Dokümantasyon Araçları

- **Microsoft Word / Google Docs** – Dokümantasyon hazırlamak için.
- **Notion / Evernote** – Gereksinimleri düzenlemek ve not almak için.
- **Jira / Confluence** – Yazılım gereksinimlerini takip etmek ve işbirliği yapmak için.
- **Trello / Asana** – Proje yönetimi ve görev takibi için.

2. Gereksinim Modelleme ve Diyagram Araçları

- **Microsoft Visio** – UML diyagramları ve süreç akışlarını oluşturmak için.
- **Lucidchart / Draw.io (diagrams.net)** – Çevrimiçi diyagram oluşturma araçları.
- **StarUML** – UML modelleri çizmek için güçlü bir araç.
- **PlantUML** – Metin tabanlı UML diyagramları üretmek için.
- **Enterprise Architect** – Gereksinim modelleme ve sistem tasarımı için.

3. Prototipleme ve Wireframing Araçları

- **Figma / Adobe XD** – UI/UX tasarımı ve prototip oluşturma.
- **Balsamiq** – Düşük seviyeli prototipleme için basit bir araç.
- **Axure RP** – Etkileşimli prototipler oluşturmak için.

- **Sketch** – Mac kullanıcıları için güçlü bir tasarım aracı.

4. İşbirliği ve Paylaşım Araçları

- **GitHub / GitLab / Bitbucket** – Gereksinim dokümanlarını yönetmek ve sürüm kontrolü için.
- **Slack / Microsoft Teams** – Takım içi iletişim ve dosya paylaşımı için.
- **Miro / Mural** – Beyin fırtınası ve fikir haritaları oluşturmak için.

5. Gereksinim Yönetimi ve İzleme Araçları

- **IBM Engineering Requirements Management DOORS** – Büyük ölçekli projeler için gereksinim yönetimi.
- **Helix RM (Perforce)** – Gereksinim takibi ve izlenebilirlik için.
- **Polarion ALM** – Gereksinim mühendisliği ve yaşam döngüsü yönetimi için.
- **ReqView** – Basit ve etkili bir gereksinim yönetimi aracı.

6. Test ve Doğrulama Araçları

- **TestRail** – Test senaryoları oluşturmak ve yönetmek için.
- **JMeter** – Performans testi için.
- **Postman** – API gereksinimlerini doğrulamak için.
- **Selenium** – Otomatik test süreçleri için.

Frontend Geliştirme Araçları

Kod Editörleri ve AI Destekli Editörler

- **VS Code** – En popüler kod editörlerinden biri.
- **Cursor** – AI destekli kod yazma ve tamamlama özelliklerine sahip bir editör.
- **Codeium** – AI destekli otokompletör.
- **IntelliJ IDEA / WebStorm** – JavaScript ve TypeScript projeleri için güçlü IDE'ler.

Frameworkler ve Kütüphaneler

- **React.js / Next.js** – Modern frontend geliştirme için.
- **Vue.js / Nuxt.js** – Hafif ve esnek frontend framework'leri.
- **Angular** – Kurumsal projelerde tercih edilen güçlü bir framework.
- **Tailwind CSS / Bootstrap** – Hızlı ve şık tasarımlar için CSS framework'leri.

Prototipleme ve UI Tasarımı

- **Figma / Adobe XD / Sketch** – UI/UX tasarımı için.
- **Storybook** – UI bileşenlerini bağımsız olarak geliştirmek ve test etmek için.

Backend Geliştirme Araçları

Backend Frameworkler ve Platformlar

- **Node.js (Express, NestJS, Fastify)** – JavaScript/TypeScript backend geliştirme için.
- **Django / Flask** – Python tabanlı projeler için güçlü backend çözümleri.
- **Spring Boot** – Java ile ölçeklenebilir backend geliştirme.
- **ASP.NET Core** – Microsoft ekosistemi için güçlü backend framework'ü.
- **FastAPI** – Performans odaklı Python API geliştirme framework'ü.

Veritabanı Yönetim Sistemleri (DBMS)

- **PostgreSQL / MySQL / MariaDB** – Geleneksel SQL tabanlı veritabanları.
- **MongoDB / Firebase** – NoSQL tabanlı veritabanları.
- **Redis** – Caching ve hızlı veri erişimi için.

API ve Authentication Araçları

- **Postman / Insomnia** – API geliştirme ve test etme için.
- **OAuth / Auth0 / Firebase Auth** – Kullanıcı kimlik doğrulama çözümleri.

Deploy ve CI/CD Araçları

Containerization ve Orkestrasyon

- **Docker** – Uygulamaları container içinde çalıştırmak için.
- **Kubernetes** – Büyük ölçekli container yönetimi.
- **Podman** – Docker alternatifi açık kaynak container yönetim aracı.

CI/CD (Continuous Integration / Deployment) Araçları

- **GitHub Actions** – Otomatik test ve deploy süreçleri için.
- **Jenkins** – Popüler CI/CD otomasyon aracı.
- **GitLab CI/CD** – GitLab entegrasyonu ile CI/CD yönetimi.
- **CircleCI / Travis CI** – Alternatif CI/CD araçları.

Bulut ve Sunucu Çözümleri

- **AWS (EC2, S3, Lambda, RDS)** – Bulut tabanlı deploy ve sunucu yönetimi.

- **Google Cloud (App Engine, Cloud Run, Firebase Hosting)** – Google ekosisteminde bulut çözümleri.
- **Microsoft Azure (Azure Functions, App Service)** – Microsoft'un bulut çözümleri.
- **Vercel / Netlify** – Serverless hosting ve kolay frontend deploy için.
- **Railway / Render / Fly.io** – Backend uygulamalarını kolayca deploy etmek için.

Test ve Debug Araçları

Otomatik ve Manuel Test Araçları

- **Jest / Mocha / Chai** – JavaScript test framework'leri.
- **Cypress / Playwright / Selenium** – UI test otomasyonu.
- **Postman / Newman** – API testleri.
- **JMeter** – Performans testi.

Hata Takip ve Debugging Araçları

- **Sentry / LogRocket** – Hata izleme ve log analizi.
- **OpenTelemetry** – Dağıtık sistemlerde izleme ve hata tespiti.
- **Wireshark** – Ağ trafiği analiz aracı.

AI Destekli Kod ve Test Araçları

- **GitHub Copilot** – AI destekli kod tamamlama.
- **Tabnine** – AI tabanlı otokompletör.
- **Cody (Sourcegraph)** – AI destekli kod analiz ve hata bulma aracı.
- **Mutable AI** – Test üretimi ve kod tamamlama için AI destekli bir platform.
- **CodeWhisperer (AWS)** – Amazon'un AI tabanlı kod yazma aracı.
- **Codium AI** – Test otomasyonu için AI destekli araç.

PROJELERİN DEĞERLENDİRME KRİTERLERİ NELERDİR?

Bu kısımda projelerin değerlendirme kriterlerinden bahsedilecektir. Projelerimiz için IEEE standartlarına göre belirlediğimiz dökümantasyon formatı yukarıda verilmiştir. Ancak yazılım yaşam döngüsündeki her fazda nelerin yapıldığının vurgulanması için "yazılım mühendisliğinin temelleri" dersinde verilen/önerilen proje şablonundaki isterler dikkate alınabilir. Bunun için hazırlanan değerlendirme kriterleri tablosu aşağıda verilmiştir.

YAZILIM GEREKSİNİMLERİ VE ANALİZİ DERSİ PROJE/ÖDEV DEĞERLENDİRME KRİTERLERİ

-Projeler, YMH114/YMH212 dersinde anlatılan kriterlere göre yapılmalı ve rapor düzenine uyulmalıdır.

-Proje ödevinde sayfa sınırı yoktur. İçeriğe bağlı olarak değerlendirilecektir.
-Teslim edilen Proje dosyası içerisine, yapılan tüm çalışma dosyalarının **Edit**'lenebilecek formatları dahil (**.doc, .vsd, .psd,** gibi formatları ayrıca tamamı **.pdf** olarak).

-Proje uygulamalarını yapanlar, geliştirdikleri yazılımları da teslim edecektir.
- Proje içerisindeki bölümlerde olması gereken zorunlu çizelgeler, şekiller, diyagramlar aşağıda belirtilmiştir.

- Genel olarak metinde Times New Roman, 12 punto, 1.5 satır aralığı, iki yana yaslı metin kullanılmalıdır.

PROJE DÖKÜMANTASYONUNDA BULUNMASI GEREKEN BÖLÜMLER VE AÇIKLAMALARI

1.Bölüm – Tanıtım Bu kısımda projenin amacı, kapsamı ve genel tanıtımı yapılmalıdır.

2.Bölüm – Planlama

- **Gant diyagramı** çizilmelidir (Maliyet kestirim dökümanı doğrultusunda oluşturulacaktır).
- **Ekip yapısı şematik olarak** verilmeli ayrıca bunların **zaman iş planları** gösterilmelidir.
- Proje planı içerisindeki alt planlama başlıkları (Şablondaki 2.8 ve 2.13 arası her bir planlama aşaması) **için gant diyagramları verilmeli** ve bu planlama aşamaları için yer alacak ekiplerin yapısı, zaman/iş planları ve her bir planlama aşamasında kullanılacak kaynaklar yazılmalıdır.

3.Bölüm – Çözümleme

- Bu kısımda konu ile alakalı örnek bir mevcut sistemin **use case diyagramı** ve sistemin işleyiş senaryosu verilmelidir.
- Mevcut sistemin **eksik yönleri belirlenerek** önerdiğiniz sistemin bu eksikliklerden hangilerine nasıl çözüm getirdiği açıklanmalıdır.
- Önerilen sistemin İşlevsel Model başlığı altında **Use case diyagramları** çizilmeli ve **her bir senaryo için Text formatında** senaryolar yazılmalıdır.
- Bilgi sistemleri/Nesneler başlığı altında önerilen sistemin **Sınıf diyagramları** çizilecek her bir sınıfın kullanım amacı kısaca açıklanacak senaryolarla ilişkisi verilecektir.
- Sınıfların nasıl kullanılacağı veri modelinde açıklanacak gerekirse veri tabanı yapısı ile

ilişkisi verilecektir.
- **Önerilen sistemin arayüzleri** kısaca tanıtılacaktır. Verilen arayüzlerin maliyet kestirim dokümanında kullanılıp kullanılmadığı incelenecektir. **Kodlanan tüm arayüzler detaylı açıklamaları ile verilmelidir. Derste bahsedilen sistemsel arayüzler, ara katmanlar, servisler vb. de paylaşılmalıdır.**

4.Bölüm – Tasarım

- **Sistemin tasarım mimarisi akış diyagramı** olarak verilmelidir. Ancak diyagramın dışında verdiğiniz mimarinin açıklaması seçim nedenleri ile birlikte detaylı verilmelidir.
- Tasarım aşamasında her bir arabirim için kullanım amacı, üzerinde çalışacağı veri modeli, hangi testlerin uygulanacağı ve performans kriterlerinin ne olacağı verilmelidir.
- Arabirim tasarımların yapılmasının ardından fonksiyonel gereksinimleri sağlayacak modüllerin neler olacağı verilmelidir. Her bir modül için tasarımı, kullanıcı profilleri, entegrasyon ve test işlemlerinin **nasıl yapılacağı akış diyagramı şeklinde v**erilmelidir.
- Varsa ortak alt sistemler açıklanmalı, **modüller arası ortak veriler** belirtilmelidir.
- Ders kapsamında tanıtılan UML diyagramlarının tamamını projenize göre yapmanız gerekmektedir.
- Ayrıca ERD, rich picture gibi UML diyagramı olarak değerlendirilmeyen ama projelerde paydaşlar arası etkileşimde, pazarlama/tanıtımda/eğitimde kullanılabilecek her türlü içerik de paylaşılacaktır.

5.Bölüm – Gerçekleştirme
- Önerilen sistem gerçekleştirilirken **hangi programlama dilinin ve araçlarının, hangi teknolojilerin** neden seçildiği açıklanacaktır.
- **Veri tabanı yönetim sisteminin mimarisi** verilmelidir. Çözümleme aşamasında verilen veri modeliyle ilişkili olmasına dikkat edilecektir. (5.2.2 başlığı altındaki tüm başlıklar detaylı olarak irdelenmelidir.)
- Gerçekleştirme aşamasında **kullanılan standartlar** verilmelidir.
- Gerçekleştirim aşamasında meydana gelebilecek olağan dışı durumlar belirlenmeli **nasıl yönetileceği** kısaca açıklanmalıdır.
- **Kod gözden geçirme işleminin nasıl yapılacağı** verilmelidir.
- Git vb kullanılan platformlardaki çabalar yansıtılmalıdır.

6.Bölüm – Test
- **Doğrulama ve geçerleme işlemlerinin nasıl yapılacağına ilişkin iş zaman planı gant diyagramı** olarak verilmelidir. (planlama aşamasında verilen test planı ile uyumlu olmasına dikkat edilecektir).
- Test (doğrulama) planı için **hangi yöntemlerin neden ve nasıl kullanıldığı** detaylı olarak açıklanmalıdır.
- **Kullanılacak test araçları varsa işleyişi** kısaca açıklanmalıdır.
- **Önerilen sistem gerçekleştirilirken hangi test türü ve araçlarının neden seçildiği** açıklanacaktır.

7.Bölüm – Bakım
- Yazılım kurulum aktiviteleri hakkında bilgiler verilmelidir.
- Kurulum sonrasında yerinde destek organizasyonu tarif edilmelidir.
- Kurulum ve entegrasyon aşamalarında yapılacaklar hakkında bilgiler verilmelidir.

8.Bölüm – Sonuç
-Bu bölümde gerçekleştirilen uygulamanın detaylı bir biçimde değerlendirilmesi yapılmalıdır.

-Gerçekleştirilen sistemin mevcut sistemlerden **farkı** net bir biçimde ortaya konulmalıdır.
-Gerçekleştirilen sistemin hangi probleme çözüm ürettiği belirtilmelidir.
- Gerçekleştirilen sistemin mevcut sistemlere göre **avantajları** ve **dezavantajları** anlatılmalı, sistem benzer diğer sistemlerle tablolar ile karşılaştırılmalıdır.
-Gerçekleştirilen sistemin dezavantajları değerlendirilerek gelecekte nasıl iyileştirilebileceği detayları ile anlatılmalıdır.

9.Bölüm – Kaynaklar

- Geliştirme süreçlerinin her aşamasında yararlanılan kaynaklar (Kitap, Makale, Ders Notu, İnternet Sitesi vb) numaralandırılarak yazılmalıdır.
- Buradaki kaynaklar sadece yararlanıldığı anlamında olmayıp, doküman içerisinde kaç numaralı kaynaktan nerede yararlanıldığı kaynak göstererek verilmelidir.

muhammetbaykara.com

mbaykara@firat.edu.tr