



**YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ**

Bilgisayar Mühendisliği Bölümü

YAZILIM KALİTE ve TEST SÜRECİ Proje Raporu

Ders Yürütücüsü:

Prof. Dr. Oya KALIPSIZ

Rapor Hazırlayan:

Harun OKTAY

Öğrenci Numarası:

20011080

İletişim

Harun.oktay@std.yildiz.edu.tr

YAZILIM KALİTE ve TEST RAPORU

1.) *TANITIM – GEREKSİNİM ANALİZİ*

1) Giriş

1.1.a) Döküman Düzeni

1.1.b) Önsöz ve Teşekkür

2) Problem Tanımı

1.2.a) Özet Problem tanımı

1.2.b) Detaylı problem tanımı

3) Hedef Kitle

4) Gereksinim Analizi

1.4.a) Temel gereksinim görüşmeleri sonucu

1.4.b) Tüm gereksinimlerin detaylı çıkarımı

1.4.c) Donanım Konfigürasyonu

2.) *KOD İNCELEME ve KALİTE SPESİFİKASYONU*

1) Sınıflar ve Methodlar

2) Kullanılan Kütüphaneler

3) Kalite Perspektifinden Kod İncelemesi

3.) *YAZILIM TEST SÜRECİ*

1) Yazılım Test Planı ve Stratejisi

2) Test Amaç ve Kapsamı

3) Birim Test Aşamaları ve Test Senaryoları

4) Birim Test Sonuçları

5) Performans Testi

6) Kullanıcı Arayüz testi (UI Test)

7) Tüm Test Sonuçları özeti

8) Öneriler ve İyileştirmeler

4.) *SONUÇ*

5.) *REFERANSLAR*

1.1) Giriş

1.1.a) Döküman Düzeni:

- Tüm döküman hizalı yazılmıştır
- Ana başlıklarda kullanılan yazı düzeni
Font: Times New Roman
Font Stili: Kalın ve Altı çizili
Font Boyutu: 24
- Alt başlık yazı düzeni
Font: Times New Roman
Font Stili: Bold
Font Boyutu: 20
- Metin yazı düzeni
Font: Times New Roman
Font Stili: -
Font Boyutu: 18 (bazen 16)
(Önemli kısımlar italikleştirilebilir ya da kalın stil yapılabilir.)

1.1.b) Önsöz ve Teşekkür

Bu döküman 20011080 numaralı bilgisayar mühendisliği 3. sınıf öğrencisi Harun OKTAY tarafından, bilgisayar mühendisliği mesleki seçmeli dersi olan "Yazılım Kalite ve Test Süreci(BLM4770)" dersi dönem projesi olarak hazırlanmıştır.

Dersin yürütücüsü olan ve yazılım kalite ve test süreci hakkında öğrendiklerimi borçlu olduğum Oya KALIPSIZ hocama sonsuz teşekkürlerimi sunarım. Sayesinde yazılım kalite ve test sürecinin nasıl ilerlediğini kavradım ve bu kapsamlı dökümanı ortaya çıkarabilmiş bulundum.

Bu Dökümanda nesneye yönelik programlama dersinin dönem projesi olarak yapılan "Çarpım Tablosu Öğretme Programı" üzerinde yazılım kalite ve test incelemesi yapılmıştır. Kendi yazdığım kodu test etmektense arkadaşımın kodunu test edip, incelemenin yazılım kalite ve test sürecini daha iyi anlamamı sağlayacağını düşündüm. Bu nedenle yazılım seçimi olarak arkadaşım Saadettin BERBER'in yazdığı programı seçtim. Yazdığı kodda anlamadığım kısımlarda oldukça yardım dokundu, ona da katkıları için teşekkürü borç bilirim.

1.2) Problem Tanımı

1.2.a) Özet Problem Tanımı:

Teknolojinin gelişmesiyle neredeyse her alanda bilgisayar aracılığıyla üretilen çözümlerle karşılaşyoruz, ancak hala okullarda çarpım tablosu öğretilme biçiminin kağıt üzerinde olduğu belirtilerek müşteri tarafından bir çarpım tablosu programı yapılması istendi.

1.2.b) Detaylı Problem Tanımı:

Çarpım tablosu gibi temel yöntemler öğrencilerin matematik anlayışını derinleştirebilir ve kavramsal temellerini güçlendirebilir. Teknolojinin eğitimdeki rolü giderek artıyorken, bu yöntemin Yazılımla yapılabilecek bir program aracılığıyla gerçekleşmesi de oldukça efektif olabilir. Bu tür bir program, öğrencilere interaktif ve görsel bir öğrenme deneyimi sunabilir, öğrenme sürecini eğlenceli hale getirebilir ve motivasyonlarını artırabilir. Bu uygulamanın yazılmasındaki temel amacı bu şekilde ifade edebiliriz.

Programın kullanıcı dostu olması ve öğrencilerin ihtiyaçlarına yönelik esneklik sağlaması önemlidir. Örneğin, interaktif oyunlar, çarpım tablosunu öğretmeyi daha ilgi çekici hale getirebilir ve öğrencilerin matematikle olan ilişkilerini olumlu yönde etkileyebilir. Ayrıca, programın öğrencilerin ilerlemesini takip etmelerine ve kişiselleştirilmiş geri bildirim alabilmelerine olanak tanınması da değerlidir. Bu sayede, öğrencilerin güçlü yönleri vurgulanabilir ve zayıf noktaları üzerinde daha fazla çalışma yapmaları teşvik edilebilir.

Bir çarpım tablosu programı geliştirirken aynı zamanda geleneksel öğrenme yöntemlerinin önemini de anlamak gerekiyor. Teknoloji, eğitimdeki araçlardan biri olabilir bununla beraber öğrencilerin öğrenme deneyimlerini zenginleştirmek için geleneksel ve modern yaklaşımların dengeli bir şekilde bir araya getirilmesi önemlidir. Yapılacak olan bir çarpım tablosu programı, öğrencilerin farklı öğrenme stillerine ve ihtiyaçlarına daha iyi cevap verebilir ve onların matematik gibi temel konulardaki başarılarını artırabilir. Bütün bunların ışığında müşteri tarafından bir çarpım tablosu programı istenmiştir.

1.3) Hedef Kitle

Öğrenciler:

Programımız, 7 ila 12 yaş arasındaki çocuklar için öğrenme sürecini desteklemeyi hedefler. Çarpım tablosunu anlamamanın ve öğrenmenin temel olduğu bu yaş aralığındaki öğrenciler için interaktif ve eğlenceli bir öğrenme deneyimi sunmayı amaçlıyoruz. Çocukların dikkatlerini çekecek görsel öğeler, oyunlar ve interaktif aktivitelerle çarpım tablosu kavramını anlaşılır ve eğlenceli hale getirmeyi planlıyoruz.

Veli Kitlesi:

Programımız aynı zamanda çocuklarının eğitime destek olmak isteyen veliler için de tasarlanmıştır. Velilerin çocuklarının öğrenme sürecini takip etmelerine yardımcı olacak kaynaklar, rehberlik materyalleri ve ebeveynlere yönelik pratik ipuçları içerecek şekilde hazırlanmıştır. Bu sayede, veliler çocuklarının çarpım tablosu konusundaki ilerlemesini destekleyebilir ve evde daha etkili bir öğrenme ortamı oluşturabilirler.

Öğretmenler:

Programımız, çarpım tablosu öğretimi için sınıf içi eğitimde öğretmenlerin kullanımı için de uygundur. Öğretmenler için özel olarak hazırlanan materyaller ve ders planları, çarpım tablosu konusunu sınıf ortamında daha etkin bir şekilde aktarmalarına yardımcı olacak. Ayrıca, öğretmenlere yönelik profesyonel gelişim kaynakları da sunarak, çarpım tablosu öğretiminde yenilikçi yaklaşımların benimsenmesini ve öğrencilerin daha iyi anlamalarını sağlamayı hedefliyoruz. Öğretmenlerin programımızı kullanarak öğrencilerin ilerlemesini izlemeleri ve desteklemeleri için araçlar sunuyoruz.

1.4) Gereksinim Analizi

1.4.a) Temel gereksinim görüşmeleri sonucu

Hedef kitle ile yapılan anket çalışmaları ve müşteri ile yapılan görüşmeler aracılığı ile üründen beklenen özellikler raporlanmıştır. Bu veriler analiz edilerek bir çalışma planı ortaya çıkarılmıştır. Beklentiler şu şekilde sıralanabilir:

- Bu projeyi isteyen müşteri velilerin sisteme ayrı ayrı kaydolabilmesi gerektiğini düşündüğünü söyledi.
- Her kaydolan kişiye bir ID verilmesi gerektiği kararlaştırıldı
- Veli sisteme kaydolduktan sonra ID ve kendi seçtiği şifre ile giriş yapacak
- Giriş yaptıktan sonra sistemden çocuklarını ekleyebilecek.
- Veliler eklediği çocuklarına ödev tanımlayabilmek istiyor
- Öğrenciler velilerinin ve velilerine ulaştırılmış olan kendi ID'si ile giriş yapacak
- Giriş yaptıktan sonra kendine tanımlanmış alıştırma çözebilecek.
- Çocuklar soruların tek tek karşısına çıkmasının dikkatini tek bir soruya odaklanmasını sağladığı için, soruların tek tek çıkmasını istediğini ifade ettiler. Alıştırmaların formatı bu ifadeye göre tek tek sırasıyla çıkacak sorular şeklinde tanımlanacak.
- Öğrenciler alıştırma skorlarına kendileri bakabilecek.
- Aynı skorlar velinin panelinde de çocuğunun alt sekmesinde gözükecek.
- Veli her alıştırmanın skorunu grafik halinde görebilecek.
- Her alıştırmanın sonucunu rapor olarak excel formatında oluşturabilecek.

Yukarıdaki maddelerden yola çıkılarak bir proje yapılacaktır projede yapılacak gereksinimlerin tek tek maddeler halinde ifadesi bir sonraki bölümde sunulacak. Ardından oluşturulması gereken sınıflar ifade edilecektir.

1.4) Gereksinim Analizi

1.4.b) Detaylı gereksinim analizi

Bir önceki kısımda ele aldığımız gereksinimlerin detaylı bir şekilde yazılım ekibine sunulabilmesi için her işlem maddeler halinde hazırlandı. 52 adet gereksinimden oluşan liste aşağıdaki gibidir:

1. Program İşlevleri

1.1) Ebeveny Kayıt Ol Ekranı

1. İsim girdi isteği
2. Soyad girdi isteği
3. Şifre girdi isteği
4. Güvenlik sorusu seç
5. Cevap seç
6. Kayıt ol
7. Kayıt sonucu ebeveyne ID ver
8. Ebeveny oluştur
9. Giriş paneline yönlendirme yap

1.2) Ebeveyn Giriş Ekranı

1. ID girdi isteği
2. Şifre girdi isteği
3. Güvenlik sorusu seç
4. Cevap seç
5. Eğer girdi girilmemişse girdi girmediğini belirt
6. Yanlış girilmişse yanlış girdiyi belirt
7. Doğru girdi ise
8. Ebeveny paneline yönlendir

1.4) Gereksinim Analizi

1.4.b)Detaylı gereksinim analizi (devamı)

1.3)Ebeveny Paneli

1.3.1)Ana Sayfa

1. Üst sol bölmede "Hoşgeldiniz <Veli Adı>" yazdır
2. Çocuk ismi girdi isteği
3. Çocuk şifre girdi isteği
4. Çocuk Ekle
5. Eklenen Çocuk için ID ata
6. Çocukları listeleyen bir tablo
7. Listedden çocuk seçme özelliği
8. Listedden seçilen çocuğu sil
9. Listedden seçilen çocuğa ait atanan alıştırmaları listele
10. Seçilen çocuğa ait seçilen alıştırmanın doğru yanlış sayısını ve çözülme süresini gösteren kısım
11. Seçilen alıştırmaya ait pasta dilimi formatında "egzersiz grafiği" oluştur
12. Seçilen alıştırmaya ait sütun grafiği formatında "soru grafiği" oluştur
13. Rapor Elde et
14. Excel formatında raporu dışarı aktar
15. Egzersiz grafiği oluştur (Daire grafik)
16. Soru grafiği oluştur (Sütun grafiği)

1.4) Gereksinim Analizi

1.4.b)Detaylı gereksinim analizi (devamı)

1.4) Çocuk Paneli

1.4.1) Alıştırmalarım

1. Sol üstte "hoşgeldiniz, <çocuk adı> "yazdır.
2. Alıştırmaların listesini, aralıklarını ve soru sayılarını gösteren alıştırma listeleme ekranı.
3. Alıştırma seç
4. Seçilen alıştırma için alıştırma başlat
5. Alıştırma soru paneline yönlendirme yap

1.4.2) Soru paneli

1. Panel açıldığı andan itibaren bir sayaç başlat
2. Her soru açıldığından itibaren ileri tuşuna basana kadarki aralığı kaydet
3. Tüm test bitene kadar geçen süreyi kaydet
4. Sorunun cevabını aldığımız girdi bölmesi
5. İlk sorudan itibaren soruları ileri tuşuna basılarak ilerlet
6. Sorular bitince çocuk paneline dönüş yap

1.4.3) Alıştırma skorum

1. Alıştırmalarım kısmında seçilen alıştırmmanın sonuçlarını doğrulanış sayılarını, çözme süresini ve alınan puanı gösteren bir kısmı.

1.4) Gereksinim Analizi

1.4.b) Detaylı gereksinim analizi (devamı)

1.5) Çocuk Giriş Ekranı

1. Ebeveyninin ID girdi isteği
2. Çocuk ID girdi isteği
3. Şifre girdi isteği
4. Eğer girdi girilmemişse girdi girmediğini belirt
5. Yanlış girilmişse yanlış girdiyi belirt
6. Doğru girdi ise
7. Çocuk paneline yönlendir

1.4.c) Donanım Konfigürasyonu: (minimum)

Operating System: Windows 7/8/8.1/10

Memory: 2GB RAM or more

Processor Speed: 1.2GHz processor

Processor: Intel i3 or more

Hard Disk Space: Min 3GB for Database usage.

2.1) Sınıflar ve Methodlar

Program "Java" dilinde yazılacağı için nesneye yönelik programlama paradigması kullanılmıştır. Bu nedenler sınıflarını ve methodlarını açıklamak büyük önem arz eder.

1. Child.java

- | | |
|--------------------|-----------------------|
| 1. addExercise | 7. setScores |
| 2. printExercises | 8. findExercise |
| 3. addDoneExercise | 9. getTreeSetExercise |
| 4. getExercises | 10. getParentID |
| 5. setExercises | 11. setParentID |
| 6. getScores | |

2. Exercise.java

- | | |
|------------------------|-------------------|
| 1. setSorulardanSoru | 17. getTrueNumber |
| 2. setStartSoruSuresi | 18. setTrueNumber |
| 3. setFinishSoruSuresi | 19. getN |
| 4. getSoruCarpan1 | 20. getA1 |
| 5. getSoruCarpan2 | 21. setA1 |
| 6. SoruOlustur | 22. getA2 |
| 7. setN | 23. setA2 |
| 8. getSolved | 24. getB1 |
| 9. setSolved | 25. setB1 |
| 10. getSorular | 26. getB2 |
| 11. printExercise | 27. setB2 |
| 12. getExerciseName | 28. getIdExercise |
| 13. setExerciseName | 29. setIdExercise |
| 14. getStart | 30. setFinish |
| 15. setStart | 31. getDuration |
| 16. getFinish | 32. setDuration |

2.1) Sınıflar ve Methodlar

3. FileTransaction.java

- | | |
|-----------------------|-----------------|
| 1. getSonID | 6. dosyaOku |
| 2. setSonID | 7. start |
| 3. dosyaYaz | 8. dosyaSil |
| 4. dosyaGuncelle | 9. getExcelFile |
| 5. dosyaGuncelleChild | 10. yazdirSatir |

4. JChildPanel.java

- | | |
|-------------|----------------------|
| 1. getChild | 3. exerciseGoruntule |
| 2. setChild | 4. scoresGoruntule |

5. JExam.java

1. goruntuleCarpanlar

6. JParentPanell.java

- | | |
|---------------------------|----------------------|
| 1. childGoruntule | 4. exerciseGoruntule |
| 2. childExerciseGoruntule | 5. scoresGoruntule |
| 3. isEmpty | |

7. Parent.java

- | | |
|----------------|----------------------|
| 1. getChildren | 5. getSecurityAnswer |
| 2. removeChild | 6. childGuncelle |
| 3. addChild | 7. addExercise |
| 4. findChild | |

2.1) Sınıflar ve Methodlar

8. Person.java

- | | |
|------------|----------------|
| 1. getName | 5. getSonID |
| 2. setName | 6. setSonID |
| 3. getID | 7. toString |
| 4. setID | 8. getPassword |

9. Soru.java

- | | |
|-------------------|------------------|
| 1. getCarpan1 | 17. getFinish |
| 2. setCarpan1 | 18. setFinish |
| 3. getCarpan2 | 19. getDuration |
| 4. setCarpan2 | 20. setDuration |
| 5. getChildAnswer | 21. getResult |
| 6. setChildAnswer | 22. setResult |
| 7. soruDogruMu | 23. getStart |
| 8. isSoruDogru | 24. setSoruDogru |

2.2) Eklenmesi Gereken Kütüphaneler

JFreeChart (jfreechart-1.5.4):

Veri görselleştirme için kullanılan bir kütüphane. Grafikler oluşturmak ve çizelgeler oluşturmak için sıklıkla kullanılır.

JUnit (junit-4.12):

Java tabanlı yazılım testleri için birim testlerini otomatikleştirmek için kullanılan bir framework. Yazılımın belirli parçalarının doğru çalışıp çalışmadığını kontrol etmek için kullanılır.

Log4j:

Loglama (log) için kullanılan bir Java kütüphanesi. Uygulamanın çalışması sırasında bilgi, hata veya hata ayıklama amacıyla log mesajları oluşturur.

Commons-Codec, Commons-Collections, Commons-Compress, Commons-IO, Commons-Math:

Apache Commons projesinin bir parçası olan bu kütüphaneler, geniş bir işlevsellik yelpazesi sunar. Encoding/decoding, veri yapıları, dosya sıkıştırma, girdi/çıkı işlemleri, matematiksel işlemler gibi çeşitli alanlarda kullanılırlar.

2.2) Eklenmesi Gereken Kütüphaneler

CurvesAPI:

Matematiksel eğrilerle çalışmayı sağlayan bir kütüphane. Eğri çizme, manipülasyon ve analiz için kullanılır.

Hamcrest-Core:

Unit testlerinde kullanılan bir eşleştirme (matching) kütüphanesi. Test sonuçlarını belirli koşullara göre doğrulamak için kullanılır.

Apache POI, POI-OOXML, POI-OOXML-Lite, XMLBeans:

Apache POI, Microsoft Office dosyalarını (Excel, Word, PowerPoint) oluşturmak, okumak ve değiştirmek için kullanılan bir kütüphane setidir. XMLBeans, XML dosyalarını Java nesnelere dönüştürmeye ve Java nesnelerini XML'e dönüştürmeye olanak tanır.

SparseBitSet:

Verimli bellek kullanımı sağlayan seyrek bit kümeleri üzerinde işlem yapmayı sağlayan bir kütüphanedir. Büyük bit kümeleri üzerinde işlem yaparken kullanışlıdır.

2.3) Kalite Perspektifinden Kod İncelemesi

Programı yazan arkadaşıma sorduğum üzere, bu programı yazarken herhangi bir yazılım standardına uyma endişesi duymamış. Yine de buna rağmen yazdığı kodu temiz yazmaya ve gelecekte kendinin anlayabileceği bir biçimde yazmaya çalıştığını ifade etti. Ben yazdığı kodu kalite perspektifinden incelediğimde ise şu eksiklikleri tespit ettim:

1. Açıklamalar ve Dokümantasyon:

Sınıfın ve metodların amacını açıklayan yeterli açıklama bulunuyor. Ancak açıklamaların bir kısmı daha ayrıntılı olabilir ve belirli metodların girdi ve çıktıları hakkında daha fazla bilgi içerebilir.

2. Değişken ve Metod İsimlendirmesi:

Değişkenler ve metodlar anlamlı isimlerle adlandırılmış, bu sayede neyi temsil ettikleri anlaşılabilir. Ancak, bazı değişken veya metod isimleri daha açıklayıcı olabilir.

3. Görünürlük ve Erişim Kontrolü:

Sınıf alanları ve metodlarına erişim kontrolü sağlanmış. Ancak, bazı alanların veya metodların gereğinden fazla geniş erişim (örneğin, public) sağladığı görülüyor.

2.3) Kalite Perspektifinden Kod İncelemesi

4. Kod Yapısı ve Düzeni:

Kod akışı genellikle anlaşılır ve düzenli bir şekilde yazılmış. Bununla birlikte bazı metodların içeriği biraz daha modüler hale getirilebilir.

5. Exception Handling (İstisnai Durumlar):

İstisnai durumları ele alma konusunda herhangi bir yapı bulunmuyor. Özellikle, olası hata durumlarında kodun nasıl davranacağı belirtilmemiş.

6. Veri Yapıları ve Koleksiyonlar:

Veri yapıları doğru bir şekilde kullanılmış gibi görünüyor, ancak veri koleksiyonlarına daha spesifik ve anlamlı isimler verilebilir.

Özetle bu program, amacı o yönde olmasa dahi bazı kalite standartlarını takip etmiş. Yine de bir standarda uygun yazılsa ve yazılırken dökümantasyon tutulsa ürün anlamında oldukça faydalı olacağını düşünüyorum. Örneğin, daha ayrıntılı açıklamalar, daha açıklayıcı değişken isimlendirmesi ve daha kapsamlı bir hata yönetimi eklenmesi, kodun okunabilirliğini ve bakımını artırabilir. Ayrıca, belirli bir kodlama standardı (örneğin, Java Coding Conventions) kullanılabilir ve bu standartlara daha sıkı bir şekilde uyulabilir. fazla geniş erişim (örneğin, public) sağladığı görülüyor.

3.1) Yazılım Test Planı ve Stratejisi

Proje Adı: Çarpım Tablosu Öğrenme Uygulaması

Test Planı Amaçları:

- Uygulamanın doğruluğunu, güvenilirliğini ve kullanılabilirliğini sağlamak.
- Veli ve çocuk giriş ekranlarının doğru işlevselliğini test etmek.
- Ödev tanımlama ve çözüm süreçlerini doğrulamak.
- Çözülen alıştırmaların raporlanması ve doğruluğunun kontrol edilmesi.

Test Stratejisi:

1.Veli Giriş Ekranı Testleri:

1. Veli Kaydının doğru çalışıp çalışmadığını kontrol etmek
2. Veli girişinin doğru çalışıp çalışmadığını kontrol etmek.
3. Velinin çocuk ekleme işlevselliğini test etmek.
4. Velinin ödev tanımlama işlevselliğini test etmek.

2.Çocuk Giriş Ekranı Testleri:

1. Çocuk kaydının doğru çalışıp çalışmadığını test etmek.
2. Çocuk girişinin doğru çalışıp çalışmadığını kontrol etmek.
3. Çocuğun ödevi görebilmesi ve çözüm yapabilmesini test etmek.

3.Ödev Tanımlama ve Çözüm Testleri:

1. Veli tarafından ödev tanımlama sürecini test etmek.
2. Çocuğun ödevi görebilmesi ve çözüm yapabilmesini test etmek.
3. Çözülen alıştırmaların doğruluğunu kontrol etmek.

4.Raporlama ve Sonuç Testleri:

1. Çocuğun çözdüğü alıştırmaların sonuçlarını doğru bir şekilde raporlayıp raporlamadığını kontrol etmek.
2. Raporlama sürecinin doğruluğunu ve hassasiyetini test etmek.

3.2) Test Amaç ve Kapsamı

Test Amaçları:

1.Uygulamanın Doğruluğu:

1. Uygulamanın çarpım tablosu öğrenme sürecinde doğru ve eksiksiz bilgi sunup sunmadığını kontrol etmek.
2. Matematiksel doğruluk ve alıştırmaların içeriğinin uygunluğunu doğrulamak.

2.Güvenilirlik ve Kullanılabilirlik:

1. Uygulamanın güvenilirliğini test ederek, istikrarlı bir şekilde çalışıp çalışmadığını kontrol etmek.
2. Kullanıcıların uygulamayı kolayca kullanabilir olmasını ve işlevselliğin sorunsuz olduğunu kontrol etmek.

3.Veli ve Çocuk Giriş Ekranları:

1. Veli ve çocuk giriş ekranlarının doğru şekilde çalışmasını, kullanıcı adı ve şifre doğrulamasını test etmek.
2. Veli ve çocuk arasında doğru veri akışını sağlamak.

4.Ödev Tanımlama ve Çözüm Süreçleri:

1. Veli tarafından ödev tanımlama sürecinin işlevselliğini ve doğruluğunu kontrol etmek.
2. Çocuğun ödevleri görüp çözme süreçlerini doğrulamak.

5.Çözülen Alıştırmaların Raporlanması:

1. Çocuğun çözdüğü alıştırmaların doğru bir şekilde raporlanıp raporlanmadığını kontrol etmek.
2. Raporlama sürecinin doğruluğunu ve hassasiyetini test etmek.

Test Kapsamı:

- Veli ve çocuk giriş ekranlarının doğruluğu ve işlevselliği.
- Ödev tanımlama süreçleri ve çözüm süreçlerinin doğruluğu.
- Çözülen alıştırmaların doğrulanması ve raporlanması.
- Kullanıcı arayüzü, veri akışı ve uygulama istikrarı.

3.3) Birim Test Aşamaları ve Test Senaryoları

Test Planı Aşamaları:

1.Veli Giriş Ekranı Testleri:

1. Veli giriş ekranının kullanıcı adı ve şifre doğrulamasını test etmek.
2. Veliye ödev tanımlama işlevselliğini kontrol etmek.

2.Çocuk Giriş Ekranı Testleri:

1. Çocuk giriş ekranının doğru kullanıcı adı ve şifre doğrulamasını yapabilmesini test etmek.
2. Çocuğun ödevleri görebilmesini ve çözebilmesini kontrol etmek.

3.Ödev Tanımlama ve Çözüm Testleri:

1. Veli tarafından ödev tanımlama sürecinin doğruluğunu kontrol etmek.
2. Çocuğun ödevleri doğru bir şekilde görebilmesini ve çözebilmesini test etmek.
3. Çocuğun çözülen alıştırmaların doğruluğunu kontrol etmek.

4.Raporlama ve Sonuç Testleri:

1. Çocuğun çözümlerini doğru bir şekilde raporlayıp raporlamadığını test etmek.
2. Raporlama sürecinin doğruluğunu ve hassasiyetini kontrol etmek.

Test Senaryoları:

•Veli Giriş Senaryosu:

- Veli girişi yapıldığında doğru ekranın açılması ve ödev tanımlama işlevselliğinin kontrolü.

•Çocuk Giriş Senaryosu:

- Çocuk girişi yapıldığında doğru ekranın açılması ve ödevleri görüp çözebilme işlevselliğinin kontrolü.

•Ödev Tanımlama Senaryosu:

- Veli tarafından ödev tanımlama sürecinin doğruluğunun kontrolü.

•Çözüm ve Raporlama Senaryosu:

- Çocuğun çözümlerinin doğru bir şekilde raporlanıp raporlanmadığının kontrolü.

3.3) Birim Test Aşamaları ve Test Senaryoları

Test Ortamı ve Kullanılan Araçlar:

1.İşletim Sistemi ve Bilgisayar Özellikleri:

- 1. İşletim Sistemi:** Windows 10 Education (64-bit)
- 2. İşlemci:** AMD Ryzen 5 2500U with Radeon Vega Mobile Graphics
- 3. Bellek:** 8 GB RAM

2.Geliştirme ve Test Ortamı:

- 1. IDE (Entegre Geliştirme Ortamı):** NetBeans IDE
- 2. Programlama Dili:** Java
- 3. Test Framework:** JUnit

3.Test Aracı ve Yardımcı Araçlar:

- 1. Test Otomasyonu:** JUnit (Entegre NetBeans IDE ile)
- 2. Hata İzleme ve Raporlama:** NetBeans IDE Konsol Çıktıları

4.Test Verileri:

- 1. Örnek giriş bilgileri** (Veli ve Çocuk girişi için farklı rollerde kullanıcılar)
- 2. Farklı çarpım tabloları ve çözümleri için örnek veri setleri**

3.4) Birim Test Sonuçları

1. Veli Giriş Ekranı Testleri:

- Kullanıcı adı ve şifre doğrulama testi: **BAŞARILI**
- Ödev tanımlama işlevselliği testi: **BAŞARILI**

2. Çocuk Giriş Ekranı Testleri:

- Kullanıcı adı ve şifre doğrulama testi: **BAŞARILI**
- Ödevleri görüp çözebilme testi: **BAŞARILI**

3. Ödev Tanımlama ve Çözüm Testleri:

- Veli tarafından ödev tanımlama testi: **BAŞARILI**
- Çocuğun ödevleri görüp çözebilme testi: **BAŞARILI**
- Çocuğun çözülen alıştırmaların doğruluğu testi: **BAŞARILI**

4. Raporlama ve Sonuç Testleri:

- Çocuğun çözümlerinin doğru raporlanması testi: **BAŞARILI**
- Raporlama sürecinin doğruluğu ve hassasiyeti testi: **BAŞARILI**

Test Senaryoları:

1. Veli Giriş Senaryosu: **BAŞARILI**
2. Çocuk Giriş Senaryosu: **BAŞARILI**
3. Ödev Tanımlama Senaryosu: **BAŞARILI**
4. Çözüm ve Raporlama Senaryosu: **BAŞARILI**

Tüm testler ve senaryolar başarıyla geçmiştir. Sistemde fonksiyonel anlamda herhangi bir hata veya kusur tespit edilmemiştir.

3.5) Performans Testi

Test Hedefi: "NetBeans Profiler" test aracı kullanılarak programımızın performansı ölçüldü ve raporlandı

Test Aracı : NETBEANS PROFILER

Test Ortamı:

- İşletim Sistemi:** Windows 10 Education (64-bit)
- İşlemci:** AMD Ryzen 5 2500U with Radeon Vega Mobile Graphics
- Bellek:** 8 GB RAM

Performans Testi

1.) CPU Kullanım oranı

Tüm program ve tüm modüller test programı aracılığıyla çalıştırıldı ve kullanılan ortalama CPU miktarının %2.5 civarı seyrettiği tespit edildi. Bununla birlikte kullanılan CPU miktarının maksimum noktası %14.9 olarak ölçüldü. Bu seviyeye çıkarmak için programa oldukça fazla yüklenildi ve buna rağmen üst sınır %14.9 olduğu tespit edildi. Bu sonuca göre programın oldukça performanslı olduğu tespitini yapabiliriz.

2.Memory Kullanımı:

117MB ilk tahsisi yapılan heap memory'nin kullanım sonucu stabil durumdaki boş alan kısmı 78MB'a düşüyor ve stabil kullanımdan bu boş alanın 66MB kadar kısmı kullanılıyor olarak gösteriliyor. Performans testi sürecinde program üzerine yük bindirildiği zaman %97 kadar bir (dolu alan / tüm alan) oranına yükseldiği tespit edildi. Bu oranların programın çalışmasında herhangi bir zorluk çıkarmadığı için kabul edilebilir bulundu.

3.Thread Kullanımı

Stabil çalışma durumunda 13 thread üzerinde 4.200 civarı yüklenmiş class olduğu tespit edildi. Aynı zamanda maksimum kullanılan thread sayısının da aynı şekilde 13 olarak ölçüldü.

3.6) Kullanıcı Arayüz Testleri (UI Test)

1.) Kullanılabilirlik Testleri:

1. Navigasyon Testleri:

Programdaki menüler ve geçişler doğru çalışıyor. Ancak programda bir negatif yön bulunmakta. Veli ya da Öğrenci ekranındayken ana giriş ekranına geri dönüş özelliği eklenmemiş böylelikle veli panelindeyken başka bir veli giriş yapmak isterse ya da veli ekranından çocuk ekranına giriş yapılmak istenirse programın kapatılıp açılması gerekiyor.

1. Form ve Giriş Testleri:

Forma girilen bilgiler doğru bir şekilde kontrol ediliyor. Yani şifre yanlışsa ya da eksik bir bilgi varsa uygun uyarılar gönderilmiş bulunuyor.

1. Uyumluluk Testleri:

Responsive bir yapıya sahip olmadığı, ekran büyütülüp küçüldüğünde girdi alma ekranlarının kaybolduğu ya da kötü bir biçimde sağa sola kaydığı tespit edildi. Bu nedenle her cihaza uyumlu olmayacağını söyleyebiliriz.

3.6) Kullanıcı Arayüz Testleri (UI Test)

2.) Görünüm Testleri:

- 1. Renk ve Tasarım Testleri:** Renk kullanımı olmadığı ve herhangi bir estetik tasarım standardına uyulmadığı tespit edildi. Bunlara rağmen yine de kullanıcılar için sade ve anlaşılır bir arayüz sunduğu söylenebilir.
- 2. Doğruluk ve Uygunluk Testleri:** Yazım hataları ya da oluşturulan grafiklerde herhangi bir hata tespit edilmedi.

3.) İşlevsellik Testleri:

- 1. İşlevsellik Doğrulama:** Butonların ve menülerin doğru çalıştığı tespit edildi.
- 2. Hata Durumları Testleri:** Hatalı girişlerde ve eksik bilgi girişlerinde hata mesajı gönderilerek kullanıcı bilgilendiriliyor
- 3. Performans Testleri:** Arayüz oldukça hızlı bir şekilde tepki veriyor. Ölçülen performansı kabul edilebilir ölçüdedir.
- 4. Gerçek Kullanıcı Testleri:** Beta testlerinin yanında çocuklar ve veliler gibi gerçek kullanıcılarla yapılan görüşmelerde, programın beklentileri yerine getirdiği tespit edildi.

3.7)Tüm Test Sonuçları Özeti

1.) Junit Testi Sonuçları Özeti:

Netbeans IDE'sine eklenen "*Junit Kütüphanesi*" aracılığıyla aşağıdaki modüller üzerinde;

- Veli Giriş Ekranı Testleri
- Çocuk Giriş Ekranı Testleri
- Ödev Tanımlama ve Çözüm Testleri
- Raporlama ve Sonuç Testleri

yapılmış ve tüm testler başarıyla geçilmiştir.

2.) Performans Testi:

"*NetBeans Profiler*" aracı ile;

- CPU Kullanım Oranı
- Memory Kullanımı
- Thread Kullanımı

ölçüldü ve performans testi sonucunda bütün sonuçlar kabul edilebilir düzeyde bulundu.

2.) UI Test:

Keşif Testine benzer bir biçimde uygulama kullanılarak yapılarak;

- Kullanılabilirlik testleri
- Görünüm testleri
- İşlevsellik testleri

yapılarak programın arayüzündeki eksiklikler de incelenmiş oldu.

Değerlendirme:

- Junit testleri başarıyla tamamlandı ve programın öğrenci ve veli fonksiyonları doğru çalıştığı belirlendi.
- Performans test sonuçları, programın CPU, bellek ve thread kullanımında kabul edilebilir seviyelerde performans sergilediğini gösterdi.
- UI Testi tasarım anlamında yetersiz bulundu ancak görünüm ve işlevsellik açısından herhangi bir sorun tespit edilmedi.

3.8)Öneriler İyileştirmeler

1.) UI Tasarımında İyileştirme:

Kullanılabilirlik Odaklı Düzenleme:

Arayüzde kullanılabilirlik sorunları tespit edildi. Daha net ve kullanıcı dostu bir deneyim sunmak adına kullanıcı arayüzü tasarımı gözden geçirilmeli. Özellikle, kullanıcı işlemlerinin daha akıcı ve sezgisel olması için düzenlemeler yapılabilir.

Tasarım Kılavuzlarına Uyum:

Tasarım kılavuzları kullanılarak, programın daha tutarlı bir görünüm ve hisset sunması sağlanabilir.

2.) Performans İyileştirmeleri:

Optimizasyon Fırsatları:

Performans testlerinde belirlenen noktalarda kod optimizasyonları yapılabilir. Özellikle, bellek kullanımı veya CPU yoğun işlemlerde performans artışı sağlanabilir.

Performansı Etkileyen Modüllerin İncelenmesi:

Performans testlerindeki bulgulara dayanarak, özellikle yavaş çalışan veya kaynak tüketimi yüksek olan modüller gözden geçirilmelidir.

3.8)Öneriler İyileştirmeler

3.) Dökümantasyon ve Kalite Standartları:

1. Javadoc Uyumlu Belgelendirme:

Kodun anlaşılabilirliğini artırmak için Javadoc standartlarına uygun belgeler oluşturulabilir. Özellikle, metodların ve sınıfların detaylı açıklamaları eklenerek kodun okunabilirliği artırılabilir.

2. Kod Temizliği ve Standartlar:

Kodun genel temizliği için standartlar belirlenmeli ve bu standartlara uygun olarak kod yazılmalıdır.

4.) Süreç İyileştirmeleri:

1. Kod İncelemelerinin Düzenli Hale Getirilmesi:

Takım içi düzenli kod incelemeleri ve geri bildirimlerin değerlendirilmesi sağlanabilir. Bu sayede hataların erken tespiti ve düzeltilmesi kolaylaşabilir.

2. Sürekli Entegrasyon ve Test Otomasyonu:

Otomatik testlerin ve sürekli entegrasyonun sağlanmasıyla, hataların erken aşamalarda fark edilmesi mümkün olabilir.

Bu öneriler, programın gelişimi ve kalitesi için belirlenen eksiklikleri ele alarak daha sağlam bir temel oluşturmayı hedefler. Bu iyileştirmeler, programın performansını artırabilir ve kullanıcı deneyimini iyileştirebilir.

4.) Sonuç

Dökümantasyon Özeti:

Hiç bir dökümantasyonu bulunmayan bir program olan "çarpım tablosu öğretme" java programının;

- Programın çözüm bulduğu problem tanıtıldı.
- Ürünün hedef kitlesinin analizi yapıldı.
- Detaylı bir gereksinim analizi yapıldı.
- Kodlar incelendi, sınıfların detaylı incelemesi yapıldı.
- Ürünün kalite standartlarına uygunluğu analiz edildi.
- Yazılım test planı ve stratejisi oluşturuldu.
- Yazılım üzerinde Junit kütüphanesi ile birim testler uygulandı.
- Programın performans testi NetbeansProfiles aracı ile gerçekleştirildi.
- UI testi yapıldı, kullanılabilirlik, görünüm ve işlevsellik testleri yapıldı.
- Yapılan testler raporlandı.
- Bu rapor ışığında yapılabilecek iyileştirmeler öneri olarak sunuldu.

Bu şekilde ürünün detaylı bir analizi gerçekleştirilmiş oldu ve ürün üzerinde testler gerçekleştirilerek, ürünün kalitesi ve çalışabilir olduğu ölçülmüş oldu. Yapılan testlerin çoğunu geçerek, kullanılabilir bir program olduğunu kanıtlamış bulundu. Program fonksiyonel ya da performans açısından başarılı bulundu. Bunun yanında UI testlerinde düzeltilebilecek oldukça fazla hata da bulundu. Son kullanıcı açısından görsel ve işlevsellik oldukça önemli olduğu için tespiti yapılan hataların düzeltilmesi büyük önem taşımaktadır.

5.) Referanslar

- I. Patton, R. (2016). *Software Testing* (Second Edition). Sams.
- II. Jeff Tian, “*Software quality engineering: testing, quality assurance, and quantifiable improvement*”, John Wiley Pub.
- III. <https://www.rexblack.com/templates-and-examples>
- IV. <https://github.com/nnakul/library-management>
- V. <https://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/>
- VI. [system/blob/main/TESTING/LIS TESTPLAN DOCUMENT.pdf](system/blob/main/TESTING/LIS_TESTPLAN_DOCUMENT.pdf)
- VII. <https://medium.com/@ozaneseriu/yazilim-kali%CC%87te-ve-teste-baki%C5%9F-1963c91e7942>
- VIII. [Java for Testers, Alan Richardson, Evil Tester](#)
- IX. [Open Lecture by James Bach on Software Testing](#)