

**YAZILIM TEST PLANLAMA VE RAPORLAMA
ARACI'NIN TASARLANMASI VE KODLANMASI**

**DESIGNING AND DEVELOPING SOFTWARE TESTING
PLANNING AND REPORTING TOOL**

MUSTAFA KASAPOĞLU

DR. ÖĞR. ÜYESİ İBRAHİM ZOR

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

İstatistik Anabilim Dalı için Öngördüğü

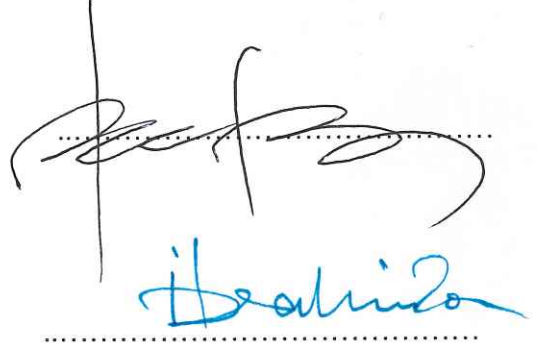
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2019

MUSTAFA KASAPÖĐLU'nun hazırladıđı "Yazılım Test Planlama ve Raporlama Aracı'nın Tasarlanması ve Kodlanması" adlı bu alıřma ařađıdaki jüri tarafından İSTATİSTİK ANABİLİM DALI'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Prof. Dr. Turhan MENTEŐ

Başkan



Dr. Öğr. Üyesi İbrahim ZOR

Danışman

Prof. Dr. Serpil CULA

Üye



Doç. Dr. ađdař Hakan ALADAĐ

Üye



Doç. Dr. Duygu İÇEN

Üye



Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından YÜKSEK LİSANS TEZİ olarak / /..... tarihinde onaylanmıştır.

Prof. Dr. Menemře GÜMÜŐDERELİÖĐLU

Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

09/09/2019

MUSTAFA KASAPOĞLU

YAYINLANMA FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan “*Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge*” kapsamında tezim aşağıda belirtilen koşullar haricince YÖK Ulusal Tez Merkezi / H. Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.
- Enstitü / Fakülte yönetim kurulu gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren ay ertelenmiştir.
- Tezim ile ilgili gizlilik kararı verilmiştir.

09/09/2019

MUSTAFA KASAPOĞLU

ÖZET

YAZILIM TEST PLANLAMA VE RAPORLAMA ARACI'NIN TASARLANMASI VE KODLANMASI

Mustafa KASAPOĞLU

Yüksek Lisans, İstatistik Bölümü

Tez Danışmanı: Dr. Öğr. Üyesi İbrahim ZOR

Eylül 2019, 68 sayfa

Günümüz yazılım dünyasında, kaliteli yazılım beklentisinin artmasının yanı sıra, yetkin personel, zaman, maliyet kısıtları nedeniyle yazılım test faaliyetlerinin planlanması ve kaynakların etkin bir şekilde kullanılması gittikçe önem kazanmaktadır. Ayrıca artan kalite ihtiyaçlarına karşılık verebilmek için yazılım testlerinin mümkün olan en kısa sürede ve detaylı şekilde yapılması beklenmektedir.

Tüm bu beklentilere cevap verebilmek için yazılım test faaliyetlerine başlamadan önce geçmişte toplanan verilerin analiz edilmesi ve planlamaların ihtiyaçları karşılayacak şekilde yapılması büyük bir önem arz etmektedir.

Sunulan tez kapsamında tasarlanan araç ile yazılım test süreçlerinde;

- yazılım testlerinde kullanılan kaynakların ve yazılım test stratejisinin planlanması,

- yazılım testlerinin kořturulması ve testler ile ilgili kayıtların tutulması,
- yazılım test sonuç raporlarının oluşturulması,
- süreçlerdeki insan hatalarının en aza indirilmesi ve tekrarlı işlerin araç tarafından yapılması,
- süreçlerin standartlaştırılması ve dokümantasyon sürecinin kolaylaştırılması

hedeflenmiştir.

Bu tez çalışmasında, belirtilen hedeflere ulaşabilmek amacıyla; hâlihazırda yazılım geliştirme süreçlerinde kullanılan diğer araçlar (IBM Rational ClearQuest, Atlassian JIRA, IBM Rational DOORS, Git vb.) ile çift yönlü (Okuma ve Yazma) iletişimi (arayüzü) olan, kullanıcı grafik arayüzü sade ve kullanımı kolay, anlaşılabilir modern kullanıcı arayüzü tasarım prensiplerini benimseyen yeni bir test yönetim aracı geliştirilmiştir.

Tasarlanan ve geliştirilen bu araç ile yazılım geliştirme süreçlerinde kayıt altına alınan tüm veriler tek bir merkezde toplanarak basit istatistiksel analizlere ek olarak ileri seviye istatistiksel analizler yapılarak farklı disiplinlerde görev alan paydaşlara süreç içindeki görev tanımına uygun yazılım test raporları üretilebilecektir.

Anahtar Kelimeler: yazılım test planlama, yazılım test yönetimi, yazılım test raporlama, CLEARQUEST, OSLC, JIRA, DOORS, DXL, DJANGO

ABSTRACT

DESIGNING AND DEVELOPING SOFTWARE TESTING PLANNING AND REPORTING TOOL

Mustafa KASAPOĞLU

Master of Science, Department of Statistics

Supervisor: Dr. Öğr. Üyesi İbrahim ZOR

September 2019, 68 pages

In today's software world, besides increasing the expectation of quality software, planning of software testing activities and efficient use of resources is gaining importance due to constraints of qualified personnel, time and cost. In addition, software tests are expected to be performed as soon as possible and in detail in order to meet the increasing quality requirements.

In order to meet all these expectations, it is very important to plan software tests effectively by analyzing the data collected in the previous tests.

With the tool designed within the scope of the thesis, the following items are aimed in the software test processes:

- Planing the resources that used in tests and deciding the software test strategy.
- Running software tests and log change requests.
- Creating software test reports automatically.

- Minimizing human errors in progresses and doing repetitive work by the tool.
- Standardization of processes and simplification of documentation process.

In order to achieve these goals; a test management tool has been developed that has two-way (Read and Write) communication with other tools currently used in software development processes (IBM Rational ClearQuest, Atlassian JIRA, IBM Rational DOORS, Git etc.), adopts intuitive graphical user interface design principles, is simple and easy to use.

On the other hand, with this tool, all data recorded in the software development processes can be collected in a single center, and in addition to simple statistical analyzes, advanced statistical analyzes can be performed with all data recorded. Also software test reports in accordance with the job description can be created for the stakeholders in different disciplines.

Keywords: software test planning, software test management, software test reporting, CLEARQUEST, OSLC, JIRA, DOORS, DXL, DJANGO

TEŐEKKÜR

Lisansüstü eğitimim boyunca engin bilgi ve tecrübelerinden yararlandığım, desteğini benden esirgemeyerek her zaman yanımda olduğunu hissettiren değerli hocam Sayın Dr. Öğr. Üyesi İbrahim ZOR'a,

Lisans ve yüksek lisans eğitimim boyunca yardım, bilgi ve tecrübeleri ile sürekli destek olan İstatistik bölümündeki tüm hocalarıma,

Bu tezin tamamlanmasında her aşamada katkı sağlayan ve destek olan Sayın Ergün DOĞAN'a,

Her koşulda sabırla yanımda olan, koşulsuz desteğini hiç esirgemeyen hayat arkadaşım Yasemin KASAPOĞLU'na,

Sonsuz teşekkürler...

Mustafa KASAPOĞLU

Eylül 2019, Ankara

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER.....	vi
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ	xii
KISALTMALAR	xiii
1. GİRİŞ	14
2. GENEL BİLGİLER.....	16
2.1. Genel Kavramlar	16
2.1.1. Yazılım Testinin Temel Amaçları.....	16
2.1.2. Yazılım Kalite Güvencesi ve Testi.....	16
2.1.3. Beyaz Kutu, Kara Kutu ve Gri Kutu Test Teknikleri	17
2.1.4. Yazılım Test Seviyeleri.....	17
2.1.5. Yazılım Test Tipleri	18
2.1.5.1. Fonksiyonel Testler	18
2.1.5.2. Fonksiyonel Olmayan Testler	19
2.1.5.3. Beyaz Kutu Testleri.....	19
2.1.5.4. Değişiklikle İlgili Testler	19
2.1.6. Yazılım Test Metrikleri.....	20
2.2. Yazılım Test Planlama ve Raporlama Araçları.....	21
2.2.3. Benzer Yazılım Test Planlama ve Raporlama Araçları	22
2.3. Geliştirme Ortamı.....	24
2.3.1. Donanım İhtiyaçları.....	24
2.3.1. Yazılım İhtiyaçları.....	24
2.4. Kullanılan Yapı ve Kütüphaneler.....	24
2.4.1. Django Web Yapısı	24
2.4.2. Bootstrap Kütüphanesi	25

2.4.3. JQuery Kütüphanesi.....	25
2.4.4. MySQL Veri Tabanı	25
2.4.5. LDAP Servisi	25
2.4.6. FullCalender.io Kütüphanesi	26
2.4.7. DataTables.net Kütüphanesi	26
2.4.8. Highcharts Kütüphanesi.....	26
3. Yazılım Test ve Raporlama Aracı Yetenekleri.....	27
3.1. Ana Sayfa.....	27
3.1.1. YKB Çerçevesi	28
3.1.1.1. Proje	28
3.1.1.2. Altsistem	30
3.1.1.3. Konfigürasyon Birimi	31
3.1.1.4. Sürüm.....	33
3.1.2. Teste Hazırla	35
3.1.2.1. Kriter	36
3.1.2.2. Kriter Açıklaması.....	37
3.1.2.3. Teste Hazır Olma Gözden Geçirme.....	39
3.1.3. Test Planla.....	39
3.1.3.1. Test Grubu	39
3.1.3.2. Test Durumu	41
3.1.3.3. Test Döngüsü	43
3.1.4. Test Koştur.....	50
3.1.4.1. Test Koştur.....	50
3.1.4.2. Bulgu.....	52
3.1.5. Test Günlüğü.....	54
3.1.5.1. Konfigürasyon Birimi Test Kaydı	54
3.1.5.2. Tümlşik Birimler Test Kaydı	57
3.1.6. Diğer İşlemler	60
3.1.6.1. Test Grupları ve Durumlarını Kopyala	60
3.1.6.2. Altsistem ve Konfigürasyon Birimlerini İçeri Al	60
3.1.6.3. Bulguları Güncelle.....	61
4. SONUÇLAR VE TARTIŞMA	63

5. KAYNAKLAR.....66

ŞEKİLLER DİZİNİ

Şekil 3.1. Ana Sayfa Genel Görünüm.....	28
Şekil 3.2. Proje Genel Görünüm.....	29
Şekil 3.3. Proje Ekleme Ekranı.....	29
Şekil 3.4. Proje Güncelleme Ekranı.....	29
Şekil 3.5. Proje Silme Ekranı.....	30
Şekil 3.6. Altsistem Genel Görünüm.....	30
Şekil 3.7. Alt Sistem Ekleme Ekranı.....	30
Şekil 3.8. Altsistem Güncelleme Ekranı.....	31
Şekil 3.9. Altsistem Silme Ekranı.....	31
Şekil 3.10. Konfigürasyon Birimi Genel Görünüm.....	31
Şekil 3.11. Konfigürasyon Birimi Ekleme Ekranı.....	32
Şekil 3.12. Konfigürasyon Birimi Güncelleme Ekranı.....	33
Şekil 3.13. Konfigürasyon Birimi Silme Ekranı.....	33
Şekil 3.14. Sürüm Genel Görünüm.....	34
Şekil 3.15. Sürüm Ekleme Ekranı.....	34
Şekil 3.16. Sürüm Güncelleme Ekranı.....	35
Şekil 3.17. Versiyon Silme Ekranı.....	35
Şekil 3.18. Teste Hazır Olma Kriterleri Genel Görünüm.....	36
Şekil 3.19. Değerlendirme Kriteri Ekleme Ekranı.....	36
Şekil 3.20. Değerlendirme Kriteri Güncelleme Ekranı.....	36
Şekil 3.21. Değerlendirme Kriteri Silme Ekranı.....	37
Şekil 3.22. Teste Hazır Olma Kriter Detayları Genel Görünüm.....	37
Şekil 3.23. Değerlendirme Kriter Detayı Ekleme Ekranı.....	38
Şekil 3.24. Değerlendirme Kriter Detayı Güncelleme Ekranı.....	38
Şekil 3.25. Değerlendirme Kriter Detayı Silme Ekranı.....	39
Şekil 3.26. Yazılım Sürümü Teste Hazır Olma Gözden Geçirme Ekranı.....	39
Şekil 3.27. Test Grupları Genel Görünüm.....	40
Şekil 3.28. Test Grubu Ekleme Ekranı.....	40
Şekil 3.29. Test Grubu Güncelleme Ekranı.....	40
Şekil 3.30. Test Grubu Arşivleme Uyarı Ekranı.....	41

Şekil 3.31. Test Grubu Silme Ekranı.....	41
Şekil 3.32. Test Durumları Genel Görünüm.	41
Şekil 3.33. Test Durumu Ekleme Ekranı.....	42
Şekil 3.34. Test Durumu Güncelleme Ekranı.	42
Şekil 3.35. Test Durumu Arşivleme Uyarı Ekranı.	43
Şekil 3.36. Test Durumu Silme Ekranı.	43
Şekil 3.37. Test Döngüsü Genel Görünüm.	43
Şekil 3.38. Test Koşusu Ekleme Ekranı.	44
Şekil 3.39. Test Döngüsü Sürümler Ekranı.....	44
Şekil 3.40. Test Durumu Sonuçları Ekranı.	45
Şekil 3.41. Test Döngüsü Güncelleme Ekranı.	45
Şekil 3.42. Test Döngüsü Silme Ekranı.	46
Şekil 3.43. Test Döngüsüne Test Durumu Ekle/Sil Ekranı.....	46
Şekil 3.44. Test Raporu Oluşturma Ekranı.	47
Şekil 3.45. Test Raporları Listesi.	48
Şekil 3.46. Test Raporu Genel Görünüm.	48
Şekil 3.47. Test Sonuçları Genel Görünüm.....	49
Şekil 3.48. Test Döngüsüne Dâhil Edilmeyen Test Durumları Görünümü.	49
Şekil 3.49. Yeni Bulunan Bulguların Listesi.....	49
Şekil 3.50. Devam Eden Bulguların Listesi.	49
Şekil 3.51. Doğrulanın Bulguların Listesi.	50
Şekil 3.52. Grafikler Görünümü.....	50
Şekil 3.53. Daha Önce Oluşturulan Raporların Listesi.	50
Şekil 3.54. Test Koşturma Ekranı.	51
Şekil 3.55. Test Durumu Durum Güncelleme Ekranı.	51
Şekil 3.56. Değişiklik Kaydı Ekleme Ekranı.	51
Şekil 3.57. Bulgu Atama, Doğrulama Ekranı.....	52
Şekil 3.58. Değişiklik Kayıtları Listesi.	53
Şekil 3.59. Bulgu Güncelleme Ekranı.	53
Şekil 3.60. Bulgu Silme Ekranı.	54
Şekil 3.61. Konfigürasyon Birimi Test Kayıtları Genel Görünüm.	54
Şekil 3.62. Konfigürasyon Birimi Test Kaydı Ekleme Ekranı.....	55
Şekil 3.63. Konfigürasyon Birimi Test Kaydı Güncelleme Ekranı.....	56

Şekil 3.64. Konfigürasyon Birimi Test Kaydı Silme Ekranı.	57
Şekil 3.65. Tümleşik Birimler Test Kayıtları Genel Görünüm.....	57
Şekil 3.66. Tümleşik Birimler Test Kaydı Ekleme Ekranı.	58
Şekil 3.67. Tümleşik Birimler Test Kaydı Güncelleme Ekranı.	59
Şekil 3.68. Tümleşik Birimler Test Kaydı Silme Ekranı.	60
Şekil 3.69. Test Grubu ve Durumları Kopyalama Ekranı.....	60
Şekil 3.70. Altsistem ve Konfigürasyon Birimi İçeri Aktarma Ekranı.....	61
Şekil 3.71. Altsistem ve Konfigürasyon Birimi İçeri Aktarma Bilgilendirme Ekranı. ..	61
Şekil 3.72. Bulguları Güncelleme Ekranı.	61
Şekil 3.73. Bulguları Güncelleme Bilgilendirme Ekranı.	62

ÇİZELGELER DİZİNİ

Çizelge 2.1. Benzer Yazılım Test Yönetim Araçları.	23
---	----

KISALTMALAR

Kısaltmalar

RSS	Çok Basit Besleme (Really Simple Syndication)
SQL	Yapısal Sorgu Dili (Structured Query Language)
HTML	Hiper Metin İşaret Dili (Hyper Text Markup Language)
CSS	Basamaklı Stil Şablonları (Cascading Style Sheets)
JS	Java Betiği (Java Script)
API	Uygulama Programlama Arayüzü (Application Programming Interface)
SVG	Ölçeklendirilebilir Vektör Grafikler (Scalable Vector Graphics)
YKB	Yazılım Konfigürasyon Birimi

1. GİRİŞ

Modern yazılım projelerinde, müşterilerin isteklerinin tamamıyla karşılandığından emin olarak kısa süre içerisinde geliştirilmesi amaçlanır [1]. Örneğin, dünyanın en büyük yazılım şirketlerinden biri olan Google'da, geliştiricilerin günde ortalama 16.000 değişiklik yaptığı, otomasyon sistemlerinin ise bunlara ek 24.000 değişiklik yaptığı bilinmektedir [2]. Diğer taraftan, geliştiricilerin müşteri tarafından yapılan istekleri yazılımın kalitesinden ödün vermeden kısıtlı zaman içinde kodlaması gerekmektedir. Yeteneklerin, var olan yetenekleri bozmadan ve hatasız bir şekilde eklendiğinden emin olmak için yazılım testleri de titizlikle yapılmalıdır [3].

Yazılım testleri ile ilgili en büyük yanlış, yapılacak işlemlerin sadece testleri koşturup sonuçları kayıt altına almaktan oluştuğunu kabul etmektir. Yazılım test süreci birbirinden farklı birden çok aktiviteyi içerir ve testlerin koşturulması da bunlardan bir tanesidir. Yazılım test süreci; testlerin planlanması, analiz edilmesi, test tasarımının yapılması ve yazılması, koşturulması, sonuçların kayıt altına alınması, test sürecinin ve sonuçların raporlanması ve yazılım kalitesinin değerlendirmesi gibi birden çok aşamadan oluşur [4].

Yazılım testi, bir yazılımı manuel veya otomatik yollarla deneyerek veya değerlendirerek, belirlenmiş gereksinimleri karşıladığının doğrulanması veya beklenen ile gözlenen sonuçlar arasındaki farkların belirlenmesi sürecidir [5]. Yazılım regresyon (yineleme) testi ise bir yazılımın sonsuz sayıdaki çalışma alanından, sınırlı sayıda ve uygun şekilde seçilmiş testler ile beklenen davranışlarını karşılamaya yönelik, dinamik olarak yapılan doğrulama faaliyetlerini kapsamaktadır. Regresyon testleri; değişiklik yapılmış yazılım sürümünde yapılan değişikliklerin mevcut hataları düzelttiğini, yeni eklenen yeteneklerin hatasız olduğunu ve tüm bu değişikliklerin yazılımın geri kalanında herhangi bir bozulmaya neden olmadığını garanti altına almak için yapılan test faaliyetidir [6]. Yapılan değişikliklerin bir an önce devreye alınması için regresyon (yineleme) testlerinin kısa bir zaman içerisinde tamamlanması projeler için büyük önem taşımaktadır. Projedeki tüm test faaliyetlerinin de %80'inin regresyon testleri olduğunun tahmin edildiği [7] göz önünde bulundurulacak olursa regresyon testlerinin planlanması ve etkin bir şekilde koşturulmasının kaçınılmaz olduğu açıkça görülmektedir.

Tez çalışması beş bölümden oluşmaktadır. Birinci bölüm giriş bölümüdür. İkinci bölümde yazılım testinin amaçları, yazılım kalite güvencesi ve testi, yazılım test teknikleri, yazılım test seviyeleri, yazılım test tipleri ve yazılım test metrikleri gibi yazılım testi ile ilgili genel bilgilere değinilmiştir. Daha sonra, benzer yazılım test planlama ve raporlama araçları tanıtılmıştır. Bunun yanı sıra yazılım test planlama ve raporlama aracı geliştirme ortamı detayları verilmiştir. İkinci bölümde geliştirme esnasında kullanılan kütüphane ve çerçevelerin bilgileri verilmiştir. Üçüncü bölümde geliştirilen aracın yeteneklerinden detaylı bir şekilde bahsedilmiştir. Dördüncü bölümde tez çalışmasında geliştirilen aracın kullanılması ile elde edilen kazançların değerlendirmesi yapılmış geliştirilmeye açık yönleri verilmiştir.

2. GENEL BİLGİLER

2.1. Genel Kavramlar

2.1.1. Yazılım Testinin Temel Amaçları

Çalışılan herhangi bir yazılım projesi için yazılım testlerinin temel amaçları: [4]

- Gereksinimler, kullanıcı hikâyeleri, tasarım ve kod gibi iş ürünlerini değerlendirmek,
- Geliştirilen üründe önceden belirlenmiş tüm şartların/isteklerin yerine getirilip getirilmediğini doğrulamak,
- Geliştirilen ürünün sorunsuz çalıştığını görmek ve diğer paydaşların beklediği şekilde çalıştığını doğrulamak,
- Ürün kalitesini yükseltmek,
- Üründe ortaya çıkabilecek hataları önlemek,
- Geliştirilen ürünlerdeki hataları müşteriye ulaşmadan tespit etmek,
- Paydaşlara ürünün yazılım kalite seviyesi ile ilgili bilgi sağlamak,
- Yazılım risk seviyesini azaltmak,
- Sözleşmeye bağlı, yasal veya düzenleyici şartlara veya standartlara uymak ve / veya yazılım ürününün bu şartlara veya standartlara uygunluğunu doğrulamaktır.

2.1.2. Yazılım Kalite Güvencesi ve Testi

Yazılım testinden bahsederken genellikle kalite güvencesi ifadesi kullanılır, ancak yazılım kalite güvencesi ve testi aynı kavramlar değildir, fakat ilişkilidirler. Daha büyük bir konsept olan kalite yönetimi bu iki kavramı birbirine bağlar.

Kalite yönetimi, bir organizasyonu kalite konusunda yönlendiren ve kontrol eden tüm faaliyetleri içerir. Diğer faaliyetler arasında kalite yönetimi; hem kalite güvencesini hem de kalite kontrolünü sağlar. Kalite güvencesi genel olarak, uygun kalite seviyelerinin elde edileceğine dair güven sağlamak için uygun süreçlere bağlı kalmaya odaklanır. İşlemler uygun şekilde gerçekleştirildiğinde, bu işlemlerin yarattığı iş ürünleri genellikle daha yüksek kalitededir ve bu da hataların önlenmesine katkıda bulunur. Ek olarak, kusurların nedenlerini saptamak ve ortadan kaldırmak amacıyla kök neden analizinin kullanılması,

süreçleri iyileştirmek için geriye dönük toplantıların yapılması, yapılan tespitlere bağlı iyileştirilmelerin değerlendirilmesi, etkin kalite güvencesi için önemlidir.

Kalite kontrol ise, uygun kalite seviyelerinin elde edilmesini destekleyen test faaliyetleri de dâhil olmak üzere çeşitli faaliyetleri içerir. Test faaliyetleri, genel yazılım geliştirme veya bakım sürecinin bir parçasıdır. Kalite güvencesi, tüm sürecin doğru şekilde yürütülmesiyle ilgili olduğu için test süreçlerinin uygunluğunu denetler ve belirlenen kalite hedeflerine ulaşılmasını sağlar. Yazılım testi ise bölüm 2.1.1’de belirtilen amaçlara ulaşmayı hedeflemektedir [8].

2.1.3. Beyaz Kutu, Kara Kutu ve Gri Kutu Test Teknikleri

Yazılım testleri; kara kutu testleri ve beyaz kutu testleri olarak iki ana kategoriye ayrılmaktadır [10].

Kara kutu testleri yalnızca yazılım gereksinimlerine ve özelliklerine göre yapılan testlerdir. Tamamlayıcı olan beyaz kutu testlerinden farklı olarak testleri koştururken ilgili yazılımın iç tasarım, kodlama bilgilerine ihtiyaç duyulmadan yapılır.

Beyaz kutu testleri ise yazılımın daha çok iç yapısına yönelik yapılan testleri içermektedir. Testleri koşturmak için iyi bir yazılım ve kodlama bilgisine sahip olmak gerekmektedir.

Kara kutu ve beyaz kutu testlerine ek olarak Gri kutu testleri, yazılımın iç yapısını yeterli kadar inceleyip bilgi sahibi olduktan sonra edinilen bilgilerle kara kutu testlerinin koşturulmasıdır.

2.1.4. Yazılım Test Seviyeleri

Tipik olarak yazılım test seviyeleri dörde ayrılır [10].

- **Birim Testleri:** Geliştiricilerin ürettiği en küçük yazılım parçasına birim denir. Genellikle bir dosyada tutulan Java ve C++ kodları için class dosyaları, C için fonksiyon olabilir. Birim testler bu birimler için test verisi oluşturup gerçekleşen çıktılar ile beklenen çıktıları karşılaştırılarak yapılır.
- **Entegrasyon Testleri:** Entegrasyon testlerinde birimlerin bir araya gelmesi ile alt sistemler ve sistemler oluşur. Entegrasyon testlerinde hedef birim testlerinden başarıyla çıkmış birimlerin bir araya geldiklerinde hatasız çalışmaya devam ettiğini garanti altına almaktır. Birimlerin entegrasyonu yapılarak uygun test verisiyle işlevlerin beklenen çıktıları verip vermediği kontrol edilir.

- **Sistem Testleri:** Sistem testleri, tüm bileşenlerin bir arada bulunduğu, müşteriye teslim edilecek tüm ürünlerin (donanım birimleri, kullanıcı kitapları, eğitim dokümanları, vs.) testlere dâhil olduğu testlerdir. Sistem testlerinde en üst seviye entegrasyon hatalarının tespit edilmesi hedeflenir ve birden çok tipte testler (kullanılabilirlik, güvenlik, performans, yedekleme, fonksiyonel, vs.) koşturulur.
- **Kabul Testleri:** Kabul testleri, başarıyla tamamlandıktan sonra müşterinin geliştirilen ürünü kabul ettiği testler olarak tanımlanabilir. Kabul testleri müşteri tarafından ne kadar kapsamlı ve geniş tutulmak istense de geliştirici tarafından o kadar kısıtlı ve minimum tutulması istenmektedir. Kabul testlerinde hedef geliştirilen yazılımın müşteri taleplerini hatasız yerine getirdiğinin gösterilmesidir.

Bu tez kapsamında geliştirilen Yazılım Test Planlama ve Raporlama Aracı entegrasyon testlerinin planlanması, koşturulması, hataların takip edilmesi ve raporlanması işlevlerini yerine getirecektir.

2.1.5. Yazılım Test Tipleri

Yazılım test tipleri ilgili yazılımın belirli özelliklerini veya belirli bir test hedefine dayanarak sistemin bir bölümünü test etmeyi amaçlayan bir test etkinlikleri grubudur. Bu hedefler [4]:

- Bütünlük, doğruluk ve uygunluk gibi fonksiyonel kalite özelliklerini değerlendirilmesi,
- Güvenilirlik, performans verimliliği, güvenlik, uyumluluk ve kullanılabilirlik gibi işlevsel olmayan kalite özelliklerinin değerlendirilmesi,
- Bileşen ya da sistemin yapısının ya da mimarisinin doğru, eksiksiz ve belirtildiği gibi değerlendirilmesi,
- Daha önce tespit edilen hataların düzeltilmesi için yapılan değişikliklerin ve/veya geliştirme süreci içinde yapılan diğer değişikliklerin etkilerinin değerlendirilmesidir.

2.1.5.1. Fonksiyonel Testler

Bir sistemin fonksiyonel testleri, sistemin gerçekleştirilmesi gereken işlevleri değerlendiren testleri içerir. İşlevler; sistemin ne yapması gerektiğini tanımlayan

gereksinimler, kullanıcı hikâyeleri, işlevsel özellikler gibi ürün dokümanlarında tanımlanabileceği gibi dokümanate edilmemiş de olabilir. Fonksiyonel testler odak her seviyede farklı olsa da tüm test seviyelerinde (bkz. Bölüm 2.1.4) yapılmaz. Fonksiyonel testler yazılımın davranışını göz önünde bulundurduğu için test koşullarını ve durumlarını üretmek için kara kutu teknikleri kullanılabilir (bkz. Bölüm 2.1.3) [4].

2.1.5.2. Fonksiyonel Olmayan Testler

Bir sistemin fonksiyonel olmayan testleri, kullanılabilirlik, performans verimliliği veya güvenlik gibi sistemlerin ve yazılımların özelliklerini değerlendirir. Fonksiyonel olmayan testler, sistemin ne kadar iyi davrandığının testidir.

Yaygın yanlış algıların aksine, fonksiyonel olmayan testler tüm test seviyelerinde yapılabilir ve yapılmalıdır ve mümkün olduğunca erken yapılmalıdır. Fonksiyonel olmayan hataların geç keşfi bir projenin başarısı için son derece tehlikeli olabilir.

Fonksiyonel olmayan testler için test koşullarını ve test durumlarını üretmek için kara kutu teknikleri (bkz. Bölüm 2.1.3) kullanılabilir. Örneğin, performans testi için stres koşullarını tanımlamak için sınır değer analizi kullanılabilir [4].

2.1.5.3. Beyaz Kutu Testleri

Beyaz kutu testleri sistemin iç tasarım ya da uygulanmasına dayanan testlerden oluşur. İç tasarım; kod, mimari tasarım, iş akışları ve/veya sistem içindeki veri akış yapılarını içerebilir. Beyaz kutu testlerinin etkinliği kod kapsama oranı ile ölçülebilir.

Beyaz kutu testleri tüm test seviyelerinde yapılabilir. Birim düzeyinde yapılan küçük hataların erken aşamada tespit edilmesi ürünün başarısı için oldukça önemlidir.

Beyaz kutu testleri için test durumlarını oluşturabilmek için beyaz kutu test teknikleri kullanılabilir (bkz. Bölüm 2.1.3) [4].

2.1.5.4. Değişiklikle İlgili Testler

Bir sistemde, bir hatayı düzeltmek için veya yeni veya değişen işlevsellik nedeniyle değişiklikler yapıldığında, değişikliklerin hatayı düzelttiğini veya eklenen işlevin doğru bir şekilde uyguladığını ve öngörülemeyen herhangi bir olumsuz sonuçlara yol açmadığını doğrulamak için yazılım ürünü tekrar test edilmelidir [4].

- **Onay Testi:** Bir hata düzeltildikten sonra, yazılım, yeni yazılım sürümünde tekrar çalıştırılması gereken, hata nedeniyle başarısız olan tüm test durumlarıyla test

edilebilir. Hata, örneğin fonksiyonelliğin eksik olması durumunda, yeni testlerle de test edilebilir. En azından, hatadan kaynaklanan arızaları yeniden oluşturma adımları, yeni yazılım versiyonunda tekrar yürütülmelidir. Onay testinin amacı, orijinal hatanın başarıyla çözümlenmediğini onaylamaktır.

- **Regresyon (Yineleme) Testi:** Kodun bir bölümünde yapılan bir değişikliğin, bir düzeltme veya başka bir tasarım değişikliği, aynı bileşenlerin diğer bileşenlerinde ya da aynı bileşende, kodun diğer bölümlerinin davranışını yanlışlıkla etkilemesi olasıdır. Değişiklikler, işletim sisteminin yeni bir sürümü veya veri tabanı yönetim sistemi gibi çevrede yapılan değişiklikleri de içerebilir. Bu istenmeyen yan etkilere gerileme denir. Regresyon testi, istenmeyen bu yan etkileri tespit etmek için yapılan testleri içerir.

Onay testi ve regresyon testi tüm test seviyelerinde yapılır. Özellikle yinelemeli ve artımlı gelişim yaşam döngüleri (örneğin, Scrum, Kanban), yeni özellikler, mevcut özelliklerde yapılan değişiklikler ve kod yeniden düzenlenmesi, kodda sık sık değişikliklerle sonuçlanır ve bu değişiklikler testlerin oluşturulmasını gerektirir. Sistemin gelişen doğası gereği, onay ve regresyon testi çok önemlidir.

Bu tez kapsamında geliştirilen Yazılım Test Raporlama ve Analiz Aracı daha çok Regresyon ve Onay Testleri'nin etkinliğinin artırılmasını hedeflemektedir.

2.1.6. Yazılım Test Metrikleri

Yazılım test metriklerinin amacı bilgi toplamak ve test etkinlikleri hakkında geri bildirim ve görünürlük sağlamaktır. İzlenecek bilgiler manuel ya da otomatik olarak toplanabilir. Toplanan bu bilgiler testlerin ilerleme durumunu değerlendirmek, test çıkış kriterlerinin kontrolünü sağlamak ve özellikle çevik geliştirme yöntemleri uygulanan projelerde hedeflerin sağlanıp sağlanmadığını kontrol etmek amacıyla kullanılmaktadır. [11] Test faaliyetleri sırasında ve sonrasında metrikler toplanarak planlanan program ve bütçeye karşı ilerleme, testlerin mevcut kalitesi, test yaklaşımının yeterliliği, test faaliyetlerinin etkinliği alanlarda değerlendirmeler yapılır.

Genel test metrikleri aşağıdaki alanlarda ölçüm alınmasını sağlamaktadır: [11, 12]

- Test planında belirlenen testlerin gerçekleşme oranı

- Test durumu kořturulması (örneğin, kořturulan/kořturulmayan test durumlarının sayısı, başarılı/başarısız olan test durumları sayısı ve/veya test durumlarının geçme/kalma durumu)
- Hata bilgileri (örneğin, hata yoğunluğu, bulunan ve devam eden hatalar, hata oranı)
- Gereksinimlerin, kullanıcı hikâyelerinin, risklerin veya kodların test kapsamı
- Kaynak tahsisi ve kullanımı

Yazılım Test Planlama ve Raporlama Aracı; testlerin planlanması ve kořturulması esnasında kaynak kullanımı, testlerin gerçekleşme oranı, kořturulan test durumları, başarılı/başarısız test durumları, hata yoğunluğu, bulunan ve devam eden hatalar, testlerin kapsama oranları gibi yazılım geliştirme sürecinde yararlanılan temel metrikler için kayıtlar tutmakta ve her test kořusu sonunda üretilen raporda sunmaktadır.

2.2. Yazılım Test Planlama ve Raporlama Araçları

Yazılım Test Araçları test edilen yazılım ürününde var olan hataların doğrudan ya da dolaylı olarak bulunmasını ve raporlanmasını sağlamaktadır. Test otomasyonunun manuel testlerden daha çok kullanılır olması ile birlikte yazılım test araçları daha popüler olmuştur. Yazılım test araçları aşağıda belirtilen şekilde kategorize edilebilir [10].

- Test Tasarlama Araçları, kullanıcıya hangi testlerin kořturulması gerektiği konusunda bilgi verir ve karar vermesini kolaylaştırır. Test verisi ve test durumları oluşturan araçlar bu gruba girer [10].
- Grafik Kullanıcı Arayüzü Test Araçları, grafik kullanıcı arayüzü olan yazılımların test edilmesinde kullanılan araçlardır. Genelde web sitelerinin testlerinde kullanılan araçlar bu gruba girer [10].
- Yük ve Performans Test Araçları, geliştirilen yazılımların ulaşabileceği maksimum yükü sentetik olarak yaratıp test yazılımının bu tip durumlarda nasıl davrandığının test edilmesinde kullanılır. Ayrıca bu grupta yer alan araçların çoğu Grafik Kullanıcı Arayüz Test Araçları olarak da geliştirilir [10].
- Test Yönetim Araçları, testlerin kořturulmasını otomatikleştiren, büyük test setleri ile çalışmayı kolaylaştıran araçlardır [10].

- Test Geliştirme Araçları, testlerin otomatik yazılması ve geliştirilmesi hedeflenen araçlardır [10].
- Test Değerlendirme Araçları, testlerin kalitesini ölçen ve bu konuda rapor hazırlayan araçlardır. Özellikle testlerin kodların ne kadarının kapsadığını, yazılım gereksinimlerinin ne kadarının kapsadığı gibi bilgiler vermektedir [10].
- Statik Kod Analiz Araçları, geliştirilen yazılımın çalıştırılmadan kodun statik analiz edilmesi ve hatalarının tespit edilmesi için kullanılan araçlardır [10].

Yazılım Test Planlama ve Raporlama Aracı; testlerin tasarlanabilmesi, koşturulması ve raporlanmasına imkân sağladığı için Test Tasarlama, Yönetme ve Raporlama Araçları kategorilerine girmektedir.

2.2.3. Benzer Yazılım Test Planlama ve Raporlama Araçları

Yazılım geliştirme süreçlerinde test ve bakım süreçlerinin gitgide önem kazanması ile birlikte yazılım test tasarlama ve yönetim araçlarının da önemi oldukça artmıştır. Bu durum birçok firmanın test yönetim aracı geliştirmesini ve kullanmasını zorunlu kılmıştır. Yazılım testine duyulan ihtiyacın artmasından sonra, farklı şirketler tarafından ilgili birçok çalışma yapılmıştır.

Ürün Adı	qTest Manager	PractiTest	Zephyr (Standalone Edition)
Geliştirici Firma	Tricentis	PractiTest	SmartBear
Test Durumu Hazırlama	Var	Var	Var
Manuel Test Koşurma	Var	Var	Var
Hata Yönetimi	Var	Var	Var
JIRA Entegrasyon	Var	Var	Var
IBM CQ Entegrasyon	Yok	Yok	Yok
Raporlama	Var	Var	Var

Çizelge 2.1. Benzer Yazılım Test Yönetim Araçları.

Yukarıda verilen test araçların her biri yazılım test süreçlerini kolaylaştırmak ve güvenilir hale getirmek için geliştirilmiştir. Ayrıca bu araçların tamamı tekrar kullanılabilirliğin kolaylaştırılmasını sağlamaktadır. Yazılım Test Planlama ve Raporlama aracı bunlara benzer avantajlar sağlayan özel amaçlı bir yazılımdır.

Ayrıca Yazılım Test Planlama ve Raporlama aracı yukarıda belirtilen araçlara amaçlar ve yetenekler açısından çok benzer ancak farklı platformları desteklemektedir. Tez kapsamında geliştirilen bu aracın diğer araçlardan farklı olarak IBM ClearQuest hata takip aracı ile entegrasyonu vardır. Web tarayıcısı üzerinden kullanıcı dostu grafik arayüzü ile tüm test süreçlerinin araç üzerinden kontrol edilmesi planlanmıştır.

2.3. Geliştirme Ortamı

2.3.1. Donanım İhtiyaçları

Yazılım Test Planlama ve Raporlama Aracı aşağıda özellikleri verilen bilgisayar üzerinde geliştirilmiştir.

- Intel(R) Core(TM) i5-3317U CPU
- 4 GB Ram
- 20 GB Hard-Disk
- 19 inch Monitör
- 1920x1080 Çözünürlüklü Monitör
- Klavye, Fare ve Yazıcı

2.3.1. Yazılım İhtiyaçları

Yazılım Test Planlama ve Raporlama Aracı geliştirilmesi PyCharm IDE'si kullanılarak aşağıda belirtilen yazılımların yüklü olduğu masaüstü bilgisayarda geliştirilmiştir.

- MS Windows 10 İşletim Sistemi
- Bitnami Django Stack (v2.2.1-0)
- IBM ClearQuest
- Herhangi bir web tarayıcısı (Explorer, Edge, Chrome, vs.)
- Herhangi bir PDF okuyucu araç

2.4. Kullanılan Yapı ve Kütüphaneler

2.4.1. Django Web Yapısı

Django, açık kaynak kodlu ücretsiz web uygulaması geliştirme yapısıdır. Django ile web uygulaması geliştirmenin zorluklarının aşılmasını ve geliştiricilerin enerjilerini, kaynaklarını, temel problemler ile vakit kaybetmeden geliştirdikleri uygulamaya harcamalarını sağlamaktadır. Ana hedeflerinden bir tanesi de geliştiricilerin konseptten uygulamanın tamamlanmasına en kısa sürede ve kolayca ulaşmasını sağlamaktır [13].

Django, yaygın web geliştirme görevlerini yerine getirmek için birçok ekstra özellik (kullanıcı kimlik doğrulaması, içerik yönetimi, site haritaları, RSS (Çok Basit Besleme (Really Simple Syndication)) yayınları, vb.) içerir. Güvenlik konusunda da geliştiricilerin SQL (Yapısal Sorgu Dili (Structured Query Language)) enjeksiyonu, siteler arası komut dosyası çalıştırma (cross-site scripting), siteler arası istek sahteciliği (cross-site request

forgery) ve tıklama ihlali (clickjacking) gibi birçok güvenlik hatasından kaçınmasına yardımcı olur. Kullanıcı doğrulama sistemi ile kullanıcıların hesaplarını ve şifrelerini yönetmek için güvenli bir yol sağlar [13].

Tüm bunlara ek olarak Django, en yoğun trafik taleplerinin hızlı ve esnek şekilde ölçeklendirme yeteneği sağlamaktadır [13].

2.4.2. Bootstrap Kütüphanesi

Bootstrap, dünyanın en popüler açık kaynak kodlu ön yüz (front-end) bileşen kütüphanesidir. HTML (Hiper Metin İşaret Dili (Hyper Text Markup Language)), CSS (Basamaklı Stil Şablonları (Cascading Style Sheets)) ve JS (Java Betiği (Java Script)) kullanarak fikirlerin hızlıca uygulamaya geçirilmesini sağlamaktadır [14]. Telefon, tablet ve masaüstü bilgisayarlar için farklı ve cihazın ekran boyutuyla orantılı şekilde içeriğin görüntülenmesini sağlayan temalar kolaylıkla hazırlanabilir. Gerekli olan tüm bileşenler içerisinde bulunduğu için uyumluluk konusunda ekstra kodlama yapılmasına ihtiyaç duyulmamaktadır.

2.4.3. JQuery Kütüphanesi

JQuery, hızlı, küçük boyutlu, zengin özellikli JavaScript kütüphanesidir. Çok sayıda tarayıcıda sorunsuz çalışsan, kullanımı kolay API (Uygulama Programlama Arayüzü (Application Programming Interface))'si ile HTML belgesinde gezinme ve işleme, animasyon ile gösterme ve Ajax ile içerik güncelleme gibi birçok şeyi basitleştirir [15].

2.4.4. MySQL Veri Tabanı

MySQL, en popüler açık kaynak kodlu Oracle tarafından geliştirilen, dağıtılan ve desteklenen SQL veri tabanı yönetim sistemidir. Facebook, Google, Adobe gibi dünyanın en büyük ve en hızlı büyüyen organizasyonlarının çoğu yüksek hacimli web sitelerine ve paket yazılımlarına güç kazanmak ve geliştirme sürecinde veri tabanı alanında zaman kazanmak için MySQL kullanmaktadır [16].

2.4.5. LDAP Servisi

LDAP, Basit Dizin Erişim Protokolü (Lightweight Directory Access Protocol), dizin sunucuları ile etkileşimde bulunmak için olgun, esnek ve iyi desteklenen standartlara dayalı bir mekanizmadır. Genellikle kullanıcılar, gruplar ve uygulamalar hakkında kimlik doğrulama amacıyla kullanılır. Şifreler ve diğer hesap bilgileri gibi hassas bilgileri

depolanmasına rağmen LDAP dizin sunucusu oldukça genel amaçlı bir veri deposudur ve çok çeşitli uygulamalarda kullanılabilir [17].

2.4.6. FullCalendar.io Kütüphanesi

FullCalendar; güçlü ve hafif, geliştirici dostu açık kaynak kodlu popüler bir JavaScript takvimdir. Takvimde olayların gösterilmesi ve planlaması işlemlerini var olan yetenekleri ve ek paketler aracılığıyla oldukça kolaylaştırmaktadır. Ön yüzde farklı tema(bootstrap, temel tema, vb.) imkânlarını geliştiricilere sunmaktadır [18].

2.4.7. DataTables.net Kütüphanesi

DataTables, açık kaynak kodlu geliştirilen jQuery Javascript Kütüphanesi için bir eklentidir. Herhangi bir HTML tablosuna sayfalama, hızlı arama, çoklu kolon sıralaması, farklı kaynaklardan veri alma, kolayca tema değişikliği yapabilme, mobil dostu özelliklerini ekleyen, aşamalı geliştirimin temelleri üzerine inşa edilmiş, oldukça esnek bir araçtır [19].

2.4.8. Highcharts Kütüphanesi

Highcharts, açık kaynak kodlu SVG (Ölçeklendirilebilir Vektör Grafikler (Scalable Vector Graphics)) tabanlı, çok platformlu grafik kütüphanesidir. Kullanımı kolay, kapsamlı özelleştirme seçenekleri ile geliştiricilerin birçok taleplerini karşılamaktadır. Birçok cihaza adapte olup grafiklerin kullanılan cihazdan bağımsız olarak düzgün görüntülenmesini sağlamaktadır. Barındırdığı birçok yetenek ile web uygulaması geliştirme süresini oldukça hızlandırmaktadır [20].

3. YAZILIM TEST VE RAPORLAMA ARACI YETENEKLERİ

Yazılım Test Planlama ve Raporlama Aracı, kullanıcı dostu arayüzü tasarım temelleri ilke edinilerek tasarlanmıştır. Görsel bileşenler, görev tanımı ve yapılacak işleme göre gruplandırılarak verilmiştir. Araca kullanıcı girişi ile giriş yapılmakta ve kullanıcıların yetkilendirmesinde LDAP (bkz. Bölüm 2.4.5) kullanıcıları ve kullanıcı grupları kullanılmaktadır. Kullanıcı araca geçerli kullanıcı adı ve şifresi ile giriş yaptığında, LDAP sunucusundan kullanıcının fotoğrafı, dâhil olduğu kullanıcı grupları, adı, soyadı, görev aldığı projeler, görev adlığı projelerdeki rolleri gibi bilgiler sisteme aktarılmaktadır. Giriş yapan kullanıcının rolüne (yazılım test mühendisi, proje yöneticisi, geliştirme ekip sorumlusu, vb.) göre menü elemanları ve işlevler kısıtlanmaktadır. Örneğin proje yöneticisi sadece koşturulan testlerin raporlarını görüntüleyebilirken yazılım test sorumlusu testleri koşturabilir ve rapor oluşturabilir.

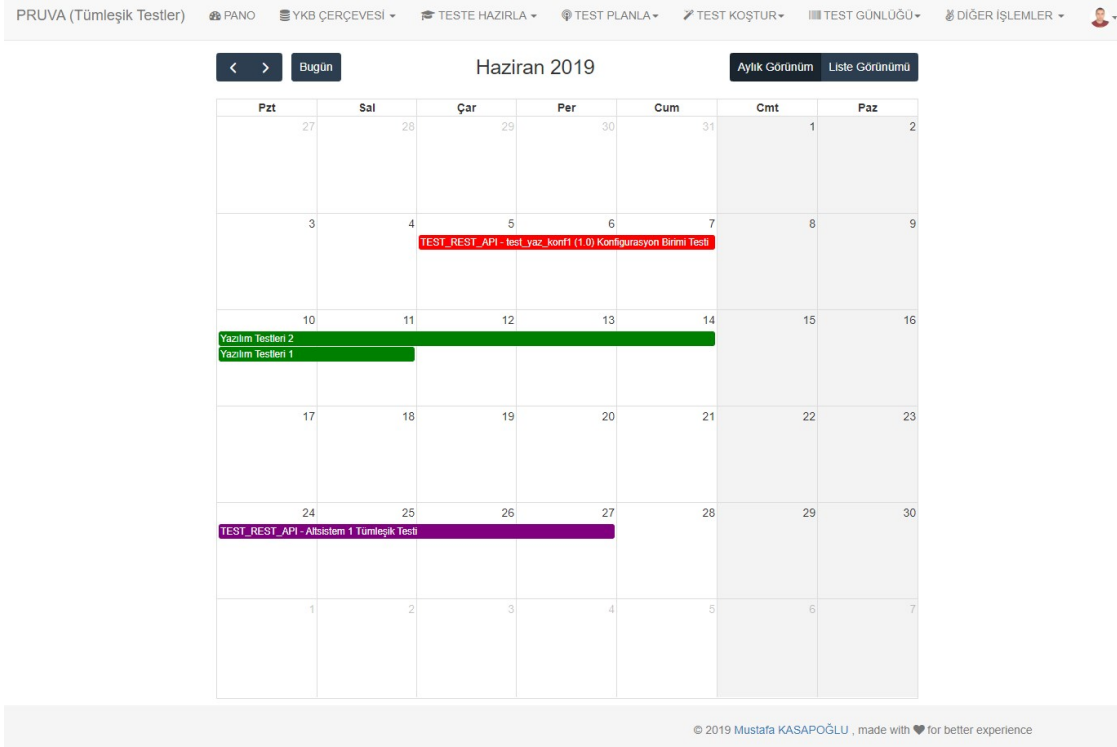
Geliştirilen araçta tüm ekranlarda tüm çözünürlüklerde içeriğin düzgün görüntülenmesini sağlayacak yapılar (bkz. Bölüm 2.4.2) kullanılmıştır.

Aracın tablo gösterimlerinde filtreleme, sunulan kolonlara göre alfabetik sıralama, sunulan kolonların gösteriminin açılıp kapatılması, çıktı alınması, işletim sistemi gececi hafızasına kopyalanması, .pdf, .csv ve .xlsx formatlarında verinin dışa aktarımı yapılabilmektedir. Ayrıca tablo gösterimlerinde yeni eleman tanımlama, mevcut elemanların düzenlenmesi ve silinmesi işlemleri için yetkilendirme gerekmektedir. Böylece yetkisiz kullanıcıların eleman ekleme, düzenleme ve silme işlemlerini gerçekleştirmesi engellenmiştir.

Yazılım Test Planlama ve Raporlama Aracı'nda eleman ekleme ve düzenleme formlarında alanların uygunlukları ve zorunlu alanların boş olmaması kontrol edilir, eğer uygun olmayan bir ya da birden çok alan varsa uyarı verilerek kullanıcının ilgili alan(lar)ı düzeltilmesi gerektiği (hataya neden olan sebepler ile birlikte) kullanıcıya bildirilir.

3.1. Ana Sayfa

Kullanıcı giriş yaptıktan sonra planlanan ve yapılan işlemlerin görüntülediği ana sayfaya erişir. (Bkz. Şekil 3.1)



Şekil 3.1. Ana Sayfa Genel Görünüm.

Sonraki bölümlerde detaylarından bahsedilecek olan ana menüde yer alan işlevler:

1. YKB (Yazılım Konfigurasyon Birimi) Çerçevesi
2. Teste Hazırla
3. Test Planla
4. Test Koştur
5. Test Günlüğü
6. Diğer İşlemler
7. Kullanıcı Profili

3.1.1. YKB Çerçevesi

Bu menü altından Konfigurasyon Birimi ile ilgili olan proje, altsistem ve sürüm bilgileri veri girişleri yapılmaktadır. Veri tabanında kayıtlı olan proje, alt sistem, konfigurasyon birimleri ve sürümleri de bu menüler altından tablo olarak sunulmaktadır.

3.1.1.1. Proje

Proje menüsü açıldığında veri tabanında kayıtlı projeler listelenmektedir. (Bkz. Şekil 3.2)

Kod	Ad	Dosya Yolu	İşlemler
1	TEST_REST_API		Güncelle Sil
1234	Yazılım Test Planlama ve Raporlama Aracı		Güncelle Sil

İlk Önceki 1 Sonraki Son

Şekil 3.2. Proje Genel Görünüm.

Ekle: Yeni proje ekleme ekranını açar. (Bkz. Şekil 3.3) Proje adı, kodu ve dosya yolu bilgileri girildikten sonra ekleme yapılır.

Kayıt Ekle
×

Ad:

En fazla 200 karakter

Kod:

Dosya Yolu:

En fazla 200 karakter

İptal
Kaydet

Şekil 3.3. Proje Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. Güncelleme ekranında projenin altında tanımlı olan alt sistemlerin listesi de verilir. (Bkz. Şekil 3.4)

Kayıt Güncelle
×

Ad:

En fazla 200 karakter

Kod:

Dosya Yolu:

En fazla 200 karakter

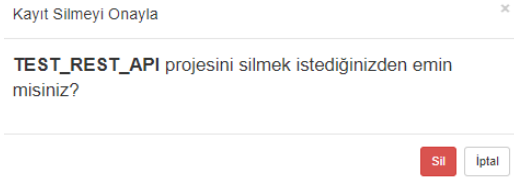
Altsistemler

- 1 TEST_REST_API - test_sistemaltsistem1
- 1 TEST_REST_API - test_sistemaltsistem2

İptal
Kaydet

Şekil 3.4. Proje Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.5.) Onay verildikten sonra silme işlemi gerçekleşir.



Kayıt Silmeyi Onayla

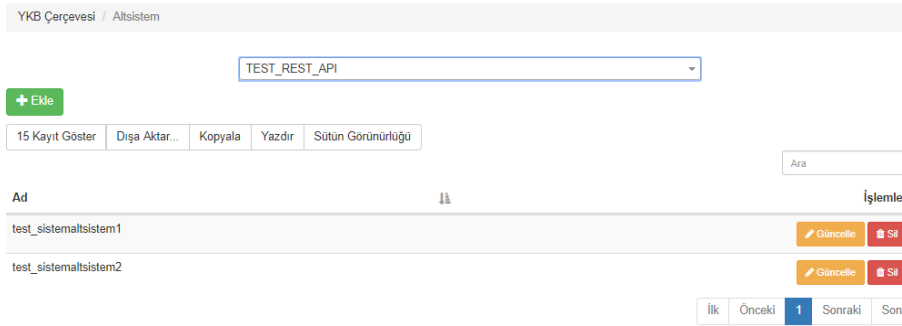
TEST_REST_API projesini silmek istediğinizden emin misiniz?

Sil İptal

Şekil 3.5. Proje Silme Ekranı.

3.1.1.2. Altsistem

Altsistem menüsü açıldığında proje seçimi yapıldıktan sonra veri tabanında kayıtlı altsistemler listelenmektedir. (Bkz. Şekil 3.6)



YKB Çerçevesi / Altsistem

TEST_REST_API

+ Ekle

15 Kayıt Göster Dişta Aktar... Kopyala Yazdır Sütun Görünürlüğü

Ara

Ad	İşlemler
test_sistemaltsistem1	Güncelle Sil
test_sistemaltsistem2	Güncelle Sil

İlk Önceki 1 Sonraki Son

Şekil 3.6. Altsistem Genel Görünüm.

Ekle: Yeni altsistem ekleme ekranını açar. (Bkz. Şekil 3.7) Altsistem adı girildikten sonra başta seçilen proje için ekleme yapılır.



Kayıt Ekle

Ad:

En fazla 200 karakter

İptal Kaydet

Şekil 3.7. Alt Sistem Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. Güncelleme ekranında altsistemin altında tanımlı olan konfigürasyon birimlerinin listesi de verilir. (Bkz. Şekil 3.8)

Kayıt Güncelle ×

Ad:

En fazla 200 karakter

Konfigürasyon Birimleri

- test_yaz_konf1 / test_yaz_konf1

Şekil 3.8. Altsistem Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.9.) Onay verildikten sonra silme işlemi gerçekleşir.

Kayıt Silmeyi Onayla ×

test_sistemaltsistem1 altsistemini silmek istediğinizden emin misiniz?

Şekil 3.9. Altsistem Silme Ekranı.

3.1.1.3. Konfigürasyon Birimi

Konfigürasyon birimi menüsü açıldığında proje ve altsistem seçimi yapıldıktan sonra veri tabanında kayıtlı konfigürasyon birimleri listelenmektedir. (Bkz. Şekil 3.10)

YKB Çerçevesi / Konfigürasyon Birimi

TEST_REST_API test_sistemaltsistem1

15 Kayıt Göster Dışa Aktar... Kopyala Yazdır Sütun Görünürlüğü

Ara

Stok Num.	Kısa Ad	Ad	İşlemler
test_yaz_konf1	test_yaz_konf1	test_yaz_konf1	<input type="button" value="Güncelle"/> <input type="button" value="Sil"/>

İlk Önceki 1 Sonraki Son

Şekil 3.10. Konfigürasyon Birimi Genel Görünüm.

Ekle: Yeni konfigürasyon birimi ekleme ekranını açar. (Bkz. Şekil 3.11) Konfigürasyon birimi adı, kısa adı, stok numarası ve açıklama bilgileri girildikten sonra başta seçilen proje altındaki altsistem için ekleme yapılır.

Kayıt Ekle ×

Ad:

En fazla 200 karakter

Kısa Ad:

En fazla 200 karakter

Stok Numarası:

En fazla 200 karakter

Açıklama:

En fazla 200 karakter

Şekil 3.11. Konfigürasyon Birimi Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. Güncelleme ekranında konfigürasyon birimi altında tanımlı olan sürümlerin listesi de verilir. (Bkz. Şekil 3.12)

Kayıt Güncelle ×

Ad:

En fazla 200 karakter

Kısa Ad:

En fazla 200 karakter

Stok Numarası:

En fazla 200 karakter

Açıklama:

En fazla 200 karakter

Sürümler

- v1.1 - test_yaz_konf1 / test_yaz_konf1
- v1.2 - test_yaz_konf1 / test_yaz_konf1

Şekil 3.12. Konfigürasyon Birimi Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.13.) Onay verildikten sonra silme işlemi gerçekleşir.

Kayıt Silmeyi Onayla ×

test_yaz_konf1 konfigürasyon birimini silmek istediğinizden emin misiniz?

Şekil 3.13. Konfigürasyon Birimi Silme Ekranı.

3.1.1.4. Sürüm

Sürüm menüsü açıldığında proje, altsistem ve konfigürasyon birimi seçimi yapıldıktan sonra veri tabanında kayıtlı sürümler listelenmektedir. (Bkz. Şekil 3.14)

YKB Çerçevesi / Sürüm

TEST_REST_API test_sistemalsistem1 test_yaz_konf1

+ Ekle

15 Kayıt Göster Dışa Aktar... Kopyala Yazdır Sütun Görünürlüğü

Ara

Num.	Revisyon	Duyurulma Tarihi	İşlemler
v1.1		2019-05-13	Güncelle Sil
v1.2		2019-05-16	Güncelle Sil

İlk Önceki 1 Sonraki Son

Şekil 3.14. Sürüm Genel Görünüm.

Ekle: Yeni sürüm ekleme ekranını açar. (Bkz. Şekil 3.15) Sürüm numarası, Açıklama, Revizyon ve Duyuru Tarihi bilgileri girildikten sonra başta seçilen konfigürasyon birimi için ekleme yapılır.

Kayıt Ekle

Sürüm Numarası:

En fazla 200 karakter

Açıklama:

En fazla 200 karakter

Revizyon:

En fazla 200 karakter

Duyurulma Tarihi:

İptal Kaydet

Şekil 3.15. Sürüm Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. Güncelleme ekranında ilgili yazılım sürümünün dâhil olduğu test döngüleri listesi de verilir. (Bkz. Şekil 3.16)

Kayıt Güncelle ×

Sürüm Numarası:

En fazla 200 karakter

Açıklama:

En fazla 200 karakter

Revizyon:

En fazla 200 karakter

Duyurulma Tarihi:

Test Döngüleri

- Yazılım Testleri 1

Şekil 3.16. Sürüm Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.17.) Onay verildikten sonra silme işlemi gerçekleşir.

Kayıt Silmeyi Onayla ×

v1.1 sürümünü silmek istediğinizden emin misiniz?

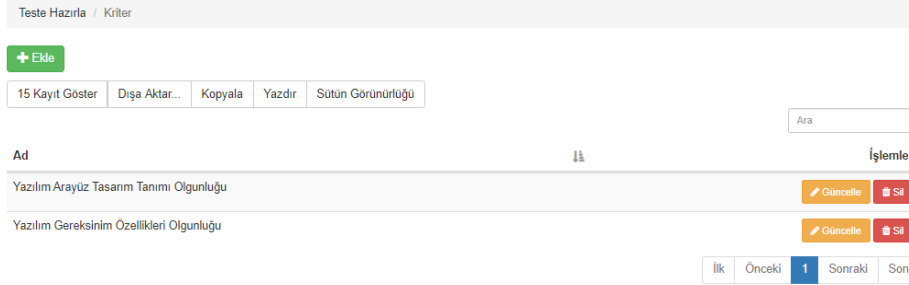
Şekil 3.17. Versiyon Silme Ekranı.

3.1.2. Teste Hazırla

Teste Hazır Olma Durumu değerlendirme menüsü daha önce tanımlanmış yazılım sürümlerinin teste hazır olma durumlarının değerlendirilmesi için gerekli ekranlardan oluşur. Teste hazır olma değerlendirilmesi yapılabilmesi için öncelikle Kriter menüsü altında kriterler ve bu kriterlere ilişkin detayların tanımlanmış olması gerekir. (Bkz. Bölüm 3.1.2.1 ve 3.1.2.2) Kriter detayları da tanımlandıktan sonra değerlendirme yapılacak olan yazılım sürümü seçilerek değerlendirme işlemi tamamlanır. (Bkz. Bölüm 3.1.2.3)

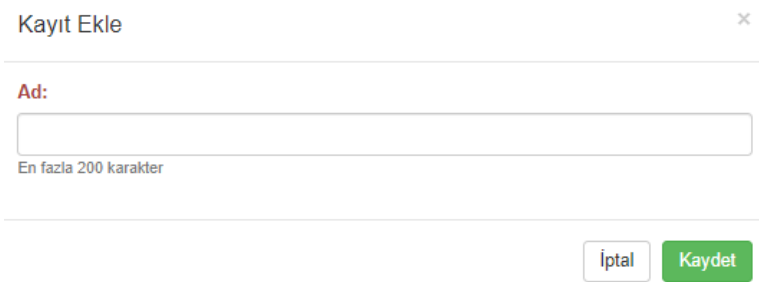
3.1.2.1. Kriter

Kriter menüsü açıldığında veri tabanında kayıtlı teste hazır olma kriterleri listelenmektedir. (Bkz. Şekil 3.18)



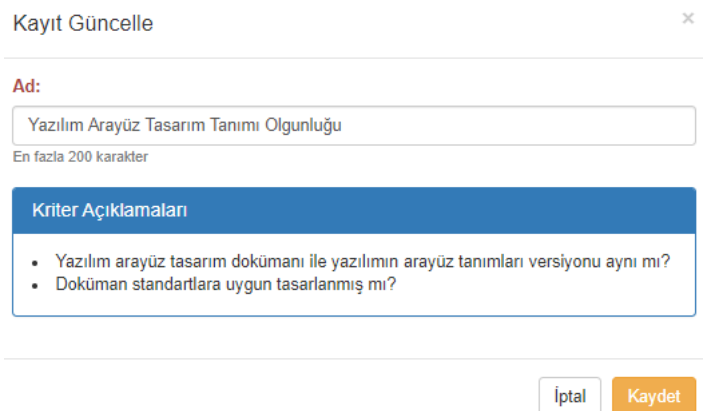
Şekil 3.18. Teste Hazır Olma Kriterleri Genel Görünüm.

Ekle: Yeni değerlendirme kriteri ekleme ekranını açar. (Bkz. Şekil 3.19) Kriter adı girilerek ekleme yapılır.



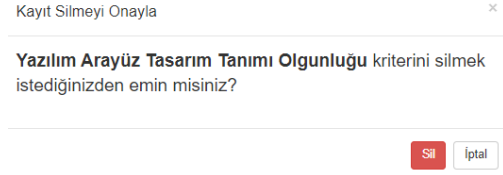
Şekil 3.19. Değerlendirme Kriteri Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. Güncelleme ekranında ilgili değerlendirme kriteri altında tanımlanmış kriter detaylarının listesi de verilir. (Bkz. Şekil 3.20)



Şekil 3.20. Değerlendirme Kriteri Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.21.) Onay verildikten sonra silme işlemi gerçekleşir.



Şekil 3.21. Değerlendirme Kriteri Silme Ekranı.

3.1.2.2. Kriter Açıklaması

Kriter Detayı menüsü açıldığında veri tabanında kayıtlı teste hazır olma kriter detayları bağlı oldukları kriterlere göre gruplu şekilde listelenmektedir. (Bkz. Şekil 3.22)

Teste Hazır / Kriter Açıklaması

+ Ekle

15 Kayıt Göster Dışa Aktar... Kopyala Yazdır Sütun Görünürlüğü

Ara

Kriter	Açıklama	Not	En Yüksek Puan	Actions
Yazılım Arayüz Tasarım Tanımı Olgunluğu (2)				
Yazılım Arayüz Tasarım Tanımı Olgunluğu	Yazılım arayüz tasarım dokümanı ile yazılımın arayüz tanımları versiyonu aynı mı?		10	Güncelle Sil
Yazılım Arayüz Tasarım Tanımı Olgunluğu	Doküman standartlara uygun tasarlanmış mı?		10	Güncelle Sil
Yazılım Gereksinim Özellikleri Olgunluğu (1)				
Yazılım Gereksinim Özellikleri Olgunluğu	Yazılım gereksinim özellikleri yeterince detaylı mı?		10	Güncelle Sil

İlk Önceki 1 Sonraki Son

Şekil 3.22. Teste Hazır Olma Kriter Detayları Genel Görünüm.

Ekle: Yeni değerlendirme kriter detayı ekleme ekranını açar. (Bkz. Şekil 3.23) Kriter, detay, açıklama ve maksimum puan bilgileri girilerek ekleme yapılır.

Kayıt Ekle ×

Kriter:

----- ▾

Açıklama:

En fazla 200 karakter

Not:

En fazla 500 karakter

En Yüksek Puan:

Kriter açıklamasının alabileceği en yüksek puan.

Şekil 3.23. Değerlendirme Kriter Detayı Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. (Bkz. Şekil 3.24)

Kayıt Güncelle ×

Kriter:

Yazılım Arayüz Tasarım Tanımı Olgunluğu ▾

Açıklama:

Yazılım arayüz tasarım dokümanı ile yazılımın arayüz tanımları versiyonu aynı mı?

En fazla 200 karakter

Not:

En fazla 500 karakter

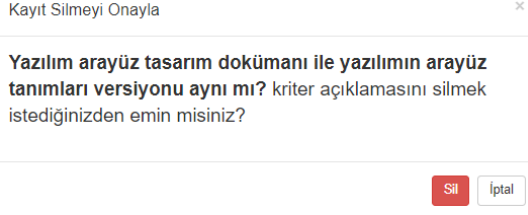
En Yüksek Puan:

10

Kriter açıklamasının alabileceği en yüksek puan.

Şekil 3.24. Değerlendirme Kriter Detayı Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.25.) Onay verildikten sonra silme işlemi gerçekleşir.



Şekil 3.25. Değerlendirme Kriter Detayı Silme Ekranı.

3.1.2.3. Teste Hazır Olma Gözden Geçirme

Teste Hazır Olma Gözden Geçirme menüsü açıldığında yazılım testlerine başlamadan önce test edilecek yazılım sürümlerinin, testlere ne kadar hazır olduğunu anlamak için daha önce belirlenen kriter (bkz. Bölüm 3.1.2.1) ve kriter detaylarına (bkz. 3.1.2.2.) göre değerlendirilme ekranı açılır. (Bkz. Şekil 3.26.) Bu ekran üzerinden ilgili kriter için alabileceği maksimum puan üzerinden değerlendirme yapılır ve kaydedilir.

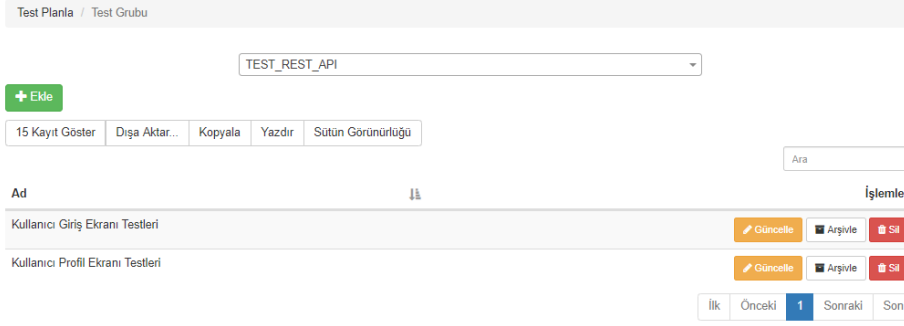
Şekil 3.26. Yazılım Sürümü Teste Hazır Olma Gözden Geçirme Ekranı.

3.1.3. Test Planla

Yazılım Testleri planlama menüsü daha önce tanımlanmış yazılım sürümlerini test koşusuna dâhil edip testlerin planlanması için gerekli ekranlardan oluşur. Test koşusunun planlanabilmesi için öncelikle Test Grubu menüsü altında test grupları ve bu gruplara ilişkin test durumlarının tanımlanmış olması gerekir. (Bkz. Bölüm 3.1.3.1 ve 3.1.3.2) Test Grupları ve Durumları tanımlandıktan sonra yazılım sürümlerinin dahil edilebildiği test döngüleri planlanabilir. (Bkz. Bölüm 3.1.2.3)

3.1.3.1. Test Grubu

Test Grubu menüsü açıldığında proje seçimi yapıldıktan sonra veri tabanında kayıtlı test grupları listelenmektedir. (Bkz. Şekil 3.27)



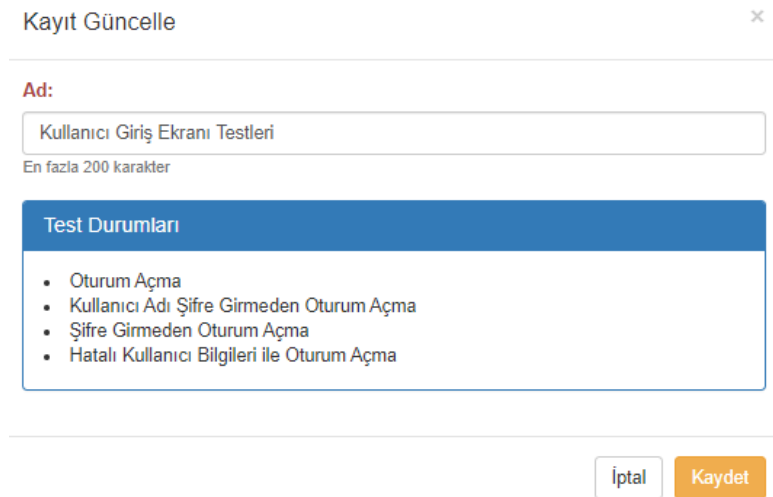
Şekil 3.27. Test Grupları Genel Görünüm.

Ekle: Yeni test grubu ekleme ekranını açar. (Bkz. Şekil 3.28) Test Grubu adı girildikten sonra başta seçilen proje için ekleme yapılır.



Şekil 3.28. Test Grubu Ekleme Ekranı.

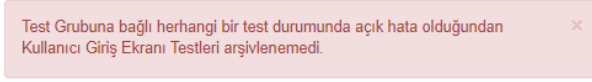
Güncelle: Seçili elemanı güncelleme ekranını açar. Güncelleme ekranında test grubu altında tanımlı olan test durumlarının listesi de verilir. (Bkz. Şekil 3.29)



Şekil 3.29. Test Grubu Güncelleme Ekranı.

Arşivle: Seçili test grubunu ve test grubunun altındaki tüm test durumlarını arşivler. Arşivlenen test grupları ve durumları test koşusu planlama ekranlarında

görüntülenmemektedir. Arşivlenmek istenen test grubuna bağlı test durumlarının herhangi birinde açık hata varsa o test grubu arşivlenmez ve kullanıcı bu konuda bilgi mesajı ile uyarılır. (Bkz. Şekil 3.30)



Şekil 3.30. Test Grubu Arşivleme Uyarı Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.31.) Onay verildikten sonra silme işlemi gerçekleşir.



Şekil 3.31. Test Grubu Silme Ekranı.

3.1.3.2. Test Durumu

Test Durumu menüsü açıldığında proje seçimi yapıldıktan sonra veri tabanında kayıtlı test durumları test gruplarına göre gruplanarak listelenmektedir. (Bkz. Şekil 3.32)

Grup	Ad	İşlemler
Kullanıcı Giriş Ekranı Testleri (4)		
Kullanıcı Giriş Ekranı Testleri	Oturum Açma	Güncelle Arşivle Sil
Kullanıcı Giriş Ekranı Testleri	Kullanıcı Adı Şifre Girmeden Oturum Açma	Güncelle Arşivle Sil
Kullanıcı Giriş Ekranı Testleri	Şifre Girmeden Oturum Açma	Güncelle Arşivle Sil
Kullanıcı Giriş Ekranı Testleri	Hatalı Kullanıcı Bilgileri ile Oturum Açma	Güncelle Arşivle Sil
Kullanıcı Profil Ekranı Testleri (2)		
Kullanıcı Profil Ekranı Testleri	Profil Ekranının Açılması	Güncelle Arşivle Sil
Kullanıcı Profil Ekranı Testleri	Kullanıcı Profil Bilgileri	Güncelle Arşivle Sil

Şekil 3.32. Test Durumları Genel Görünüm.

Ekle: Yeni test durumu ekleme ekranını açar. (Bkz. Şekil 3.33) Test Grubu seçilir ve adı, açıklama ve not bilgileri girildikten sonra başta seçilen proje için ekleme yapılır.

Kayıt Ekle ×

Grup:

----- ▾

Ad:

Maximum 200 character

Açıklama:

Not:

Şekil 3.33. Test Durumu Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. (Bkz. Şekil 3.34)

Kayıt Güncelle ×

Grup:

Kullanıcı Giriş Ekranı Testleri ▾

Ad:

Oturum Açma

Maximum 200 character

Açıklama:

Kullanıcı giriş ekranından kullanıcı adı ve şifre girilerek oturum açılır ve oturumun açıldığı görülür.

Not:

Kullanıcı adı: mkasapoglu

Şekil 3.34. Test Durumu Güncelleme Ekranı.

Arşivle: Seçili test durumunu arşivler. Arşivlenen durumları test döngüsü planlama ekranlarında görüntülenmemektedir. Arşivlenmek istenen test durumuna bağlı açık hata varsa o test durumu arşivlenmez ve kullanıcı bu konuda bilgi mesajı ile uyarılır. (Bkz. Şekil 3.35)

Test durumuna bağlı açık hata olduğundan Kullanıcı Profil Bilgileri silinemedi.

Şekil 3.35. Test Durumu Arşivleme Uyarı Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.36.) Onay verildikten sonra silme işlemi gerçekleşir.

Kayıt Silmeyi Onayla

Oturum Açma test durumunu silmek istediğinizden emin misiniz?

Şekil 3.36. Test Durumu Silme Ekranı.

3.1.3.3. Test Döngüsü

Test Döngüsü menüsü açıldığında proje seçimi yapıldıktan sonra o proje için veri tabanında kayıtlı test döngüleri listelenmektedir. (Bkz. Şekil 3.37)

Test Planla / Test Döngüsü

TEST_REST_API

15 Kayıt Göster

Ara

Ad	Başlangıç Tarihi	Bitiş Tarihi	İşlemler
Yazılım Testleri 1	2019-06-10	2019-06-11	<input type="button" value="Sürümler"/> <input type="button" value="Sonuçlar"/> <input type="button" value="İşlemler"/>
Yazılım Testleri 2	2019-06-10	2019-06-14	<input type="button" value="Sürümler"/> <input type="button" value="Sonuçlar"/> <input type="button" value="İşlemler"/>

Şekil 3.37. Test Döngüsü Genel Görünüm.

Ekle: Yeni test döngüsü ekleme ekranını açar. (Bkz. Şekil 3.38) Test Döngüsü adı, başlangıç ve bitiş tarihi, açıklama bilgileri girilir ve daha önce seçili projede yer alan bir ya da birden fazla konfigürasyon birimi (eğer konfigürasyon birimi testi planlanacaksa bir tane konfigürasyon birimi, entegrasyon testi planlanacaksa test düzeneğinde yer alan tüm konfigürasyon birimleri) için sürüm seçilerek başta seçilen proje için ekleme yapılır. Eklenen tüm test döngüleri ana sayfadaki takvim üzerinde başlangıç ve bitiş tarihine göre gösterilir. (Bkz. Şekil 3.1.)

Kayıt Ekle



Ad:

En fazla 200 karakter.

Başlangıç Tarihi:

Bitiş Tarihi:

Sürümler:

```
test_yaz_konf1 / test_yaz_konf1
v1.1 - test_yaz_konf1 / test_yaz_konf1
v1.2 - test_yaz_konf1 / test_yaz_konf1
test_yaz_konf2 / test_yaz_konf2
v1.0 - test_yaz_konf2 / test_yaz_konf2
```

Birden fazla yazılım sürümü seçebilirsiniz.

Açıklama:

En fazla 200 karakter.

İptal

Kaydet

Şekil 3.38. Test Koşusu Ekleme Ekranı.

Sürümler: Seçili eleman için seçili tüm konfigürasyon birimlerinin sürümlerini gösteren ekranı açar. (Bkz. Şekil 3.39)

Ad	Başlangıç Tarihi	Bitiş Tarihi	Sürümler	İşlemler
Yazılım Testleri 1	2019-06-10	2019-06-11	<ul style="list-style-type: none">v1.2 - test_yaz_konf1 / test_yaz_konf1v1.0 - test_yaz_konf2 / test_yaz_konf2	<input type="button" value="Sürümler"/> <input type="button" value="Sonuçlar"/> <input type="button" value="İşlemler"/>
Yazılım Testleri 2	2019-06-10	2019-06-14		<input type="button" value="Sürümler"/> <input type="button" value="Sonuçlar"/> <input type="button" value="İşlemler"/>

İlk Önceki 1 Sonraki Son

Şekil 3.39. Test Döngüsü Sürümler Ekranı.

Sonuçlar: Seçili test döngüsüne dahil edilen test durumlarının sonuçlarına hızlıca bakış için test sonuçları ekranını açar. (Bkz. Şekil 3.40) Test durumları sonuç ekranında; test döngüsüne dâhil edilen test durumları test gruplarına göre gruplanmış olarak gösterilir ve test durumlarının açıklama ve not bilgilerine test durumu üzerine tıklanarak erişilebilir.

#	Durum	Sonuç
Kullanıcı Giriş Ekranı Testleri		
1	Oturum Açma	Geçti
Açıklama: Kullanıcı giriş ekranından kullanıcı adı ve şifre girilerek oturum açılır ve oturumun açıldığı görülür. Not: Kullanıcı adı: mkasapoglu		
2	Kullanıcı Adı Şifre Girmeden Oturum Açma	Kaldı
3	Şifre Girmeden Oturum Açma	Geçti
4	Hatalı Kullanıcı Bilgileri ile Oturum Açma	Geçti
Kullanıcı Profil Ekranı Testleri		
1	Profil Ekranının Açılması	Geçti
2	Kullanıcı Profil Bilgileri	Kaldı

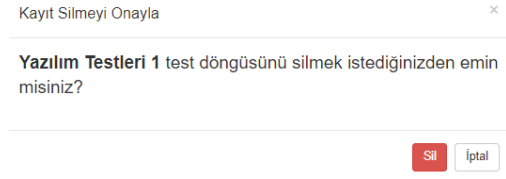
Şekil 3.40. Test Durumu Sonuçları Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. (Bkz. Şekil 3.41)

Kayıt Güncelle	X
Ad:	<input type="text" value="Yazılım Testleri 1"/>
En fazla 200 karakter.	
Başlangıç Tarihi:	<input type="text" value="06/10/2019"/>
Bitiş Tarihi:	<input type="text" value="06/11/2019"/>
Sürümler:	<div style="border: 1px solid #ccc; padding: 5px;"><p>test_yaz_konf1 / test_yaz_konf1</p><p>v1.1 - test_yaz_konf1 / test_yaz_konf1</p><p>v1.2 - test_yaz_konf1 / test_yaz_konf1</p><p>test_yaz_konf2 / test_yaz_konf2</p><p>v1.0 - test_yaz_konf2 / test_yaz_konf2</p></div>
Birden fazla yazılım sürümü seçebilirsiniz.	
Açıklama:	<div style="border: 1px solid #ccc; height: 40px;"></div>
En fazla 200 karakter.	
<input type="button" value="İptal"/> <input type="button" value="Kaydet"/>	

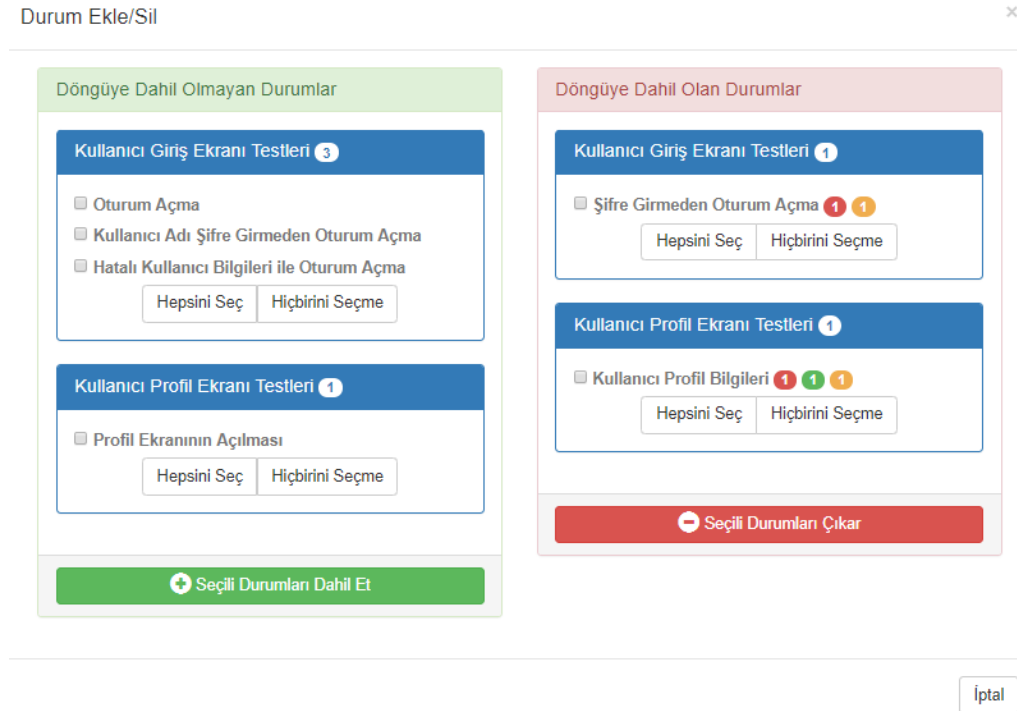
Şekil 3.41. Test Döngüsü Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.42.) Onay verildikten sonra silme işlemi gerçekleşir.



Şekil 3.42. Test Döngüsü Silme Ekranı.

Durum Ekle/Sil: Seçili test döngüsüne test durumları ekleme çıkarma ekranını açar. (Bkz. Şekil 3.43.) Test döngüsüne test durumu ekleme ekranında o test döngüsüne dâhil olan test durumları ve dâhil olmayan test durumları ayırt edici şekilde gösterilir. Ayrıca test durumlarında açık olan, çözülmüş olan ve yeni olan hata sayıları da ayrıca verilmektedir.



Şekil 3.43. Test Döngüsüne Test Durumu Ekle/Sil Ekranı.

Kopyala: Seçili test döngüsünün tüm yazılım sürümleri ve dâhil edilen test durumları ile birlikte yeni bir kopyasını oluşturur.

Rapor Oluştur: Seçili test döngüsünün sonuçlarını duyurmak için rapor oluşturma ekranını açar. (Bkz. Şekil 3.44.) İlgili test döngüsü için rapor adı girilerek; o anki test sonuçlarına göre test döngüsü raporu oluşturulur.

Kayıt Ekle ×

Ad:

En fazla 200 karakter

Şekil 3.44. Test Raporu Oluşturma Ekranı.

Oluşturulmuş Raporlar: Seçili test döngüsü için daha önce oluşturulmuş test raporlarının listesini verir. (Bkz. Şekil 3.45.) Test raporu;

- Genel Görünüm: Testlerin başlangıç bitiş zamanı, konfigürasyon birim(ler)inin sürüm, test sonuçları tablosu (Geçen, kalan, henüz tamamlanmış, koşturulmamış, test döngüsüne dahil edilmemiş test durumları), doğrulanan, doğrulanamayan ve yeni bulunan değişiklik önerileri bilgilerini içerir. (Bkz. Şekil 3.46.)
- Test Sonuçları: Test döngüsüne dâhil edilen testlerin test sonuçları (bkz. Şekil 3.47.) ve dâhil edilmeyen test durumu bilgilerinin (bkz. Şekil 3.48.) verildiği bölümdür. Bu ekranda test durumları ile ilgili bulguların detay bilgilerine de erişilebilir.
- Yeni Bulunan Bulgular: İlgili test döngüsünde tespit edilen bulguların listesini içerir. (Bkz. Şekil 3.49.) Bulgular konfigürasyon birimlerine göre gruplanarak sunulur.
- Devam Eden Bulgular: Daha önceki test döngülerinde tespit edilmiş ve hala çözülememiş bulguların listesinin içerir. (Bkz. Şekil 3.50.) Bulgular konfigürasyon birimlerine göre gruplanarak sunulur.
- Doğrulanın Bulgular: İlgili test döngüsünde doğrulanın bulguların listesini içerir. (Bkz. Şekil 3.51.) Bulgular konfigürasyon birimlerine göre gruplanarak sunulur.
- Grafikler: Daha önceki test döngülerinde kalan, geçen, tamamlanmamış, koşturulmamış ve dâhil edilmemiş test durumları, tespit edilen hata sayıları ve iyileştirme önerileri sayıları grafik olarak sunulur. (Bkz. Şekil 3.52.)
- Daha Önce Oluşturulan Raporlar: İlgili proje için daha önce oluşturulan test raporları listesi sunulur. (Bkz. Şekil 3.53.)

bilgileri içerir.

Test Planla / Test Döngüsü

TEST_REST_API

+ Ekle

15 Kayıt Göster | Dışa Aktar... | Kopyala | Yazdır | Sütün Görünürlüğü

Ara

Ad	Başlangıç Tarihi	Bitiş Tarihi	İşlemler
Yazılım Testleri 1	2019-06-10	2019-06-11	Sürümler Sonuçlar İşlemler
Yazılım Testleri 2	2019-06-10	2019-06-14	Sürümler İlk Önce

Cycle Actions

- Güncelle
- Sil
- Durum Ekle/Sil
- Kopyala

Raporlar

- Rapor Oluştur
- Oluşturulmuş Raporlar 1
- Yazılım Testleri 1

Şekil 3.45. Test Raporları Listesi.

Genel Bilgiler

Başlangıç Tarihi	Bitiş Tarihi				
13 Mayıs 2019 Pazartesi	14 Mayıs 2019 Salı				
Konfigürasyon Birimleri Sürümü					
#	Konfigürasyon Birimi	Sürüm			
1	test_yaz_konf1 / test_yaz_konf1	v1.2			
2	test_yaz_konf2 / test_yaz_konf2	v1.0			
Test Sonuçları					
Geçti	Kaldı	Devam Ediyor	Koşuturulmadı	Döngüye Dahil Olmayan	Toplam
0	1	1	0	4	6
Doğrulananan Bulgu Sayısı	Mevcut Bulgular	Yeni Bulgular			
0 Hata 0 İyileştirme Önerisi	2 Hata 0 İyileştirme Önerisi	0 Hata 1 İyileştirme Önerisi			

Şekil 3.46. Test Raporu Genel Görünüm.

Test Sonuçları

#	Konu	Açıklama	Not	Bağlı Bulgular	Sonuç						
1	Şifre Girmeden Oturum Açma	Giriş ekranında sadece kullanıcı adı girilerek giriş yapma düğmesine basılır ve giriş yapılamadığı görülür.	Kullanıcı Adı: mkasapoglu	<p>1 mevcut bulgu</p> <table border="1"> <thead> <tr> <th>SİVT</th> <th>Konu</th> <th>Durum</th> </tr> </thead> <tbody> <tr> <td>00056129</td> <td>Oturum Açma Uyarısı</td> <td>Çözüldü</td> </tr> </tbody> </table>	SİVT	Konu	Durum	00056129	Oturum Açma Uyarısı	Çözüldü	Devam Ediyor
SİVT	Konu	Durum									
00056129	Oturum Açma Uyarısı	Çözüldü									
1	Kullanıcı Profili Bilgileri	Kullanıcı profili ekranı açılır ve kullanıcının; <ul style="list-style-type: none"> * Fotoğraf * Kullanıcı adı * Ünvanı * Bağlı olduğu müdürlüğü * Telefon numarası * E-Posta adresi bilgilerinin gösterildiği görülür.	-	2 mevcut bulgu	Katıl						

Şekil 3.47. Test Sonuçları Genel Görünüm.

Döngüye Dahil Olmayan Test Durumları

#	Konu	Açıklama	Not
1	Oturum Açma	Kullanıcı giriş ekranından kullanıcı adı ve şifre girilerek oturum açılır ve oturumun açıldığı görülür.	Kullanıcı adı: mkasapoglu
2	Kullanıcı Adı Şifre Girmeden Oturum Açma	Kullanıcı giriş ekranında kullanıcı adı ve şifre girilmeden oturum açılması denir ve "Kullanıcı adını ve şifreyi giriniz" uyarısı çıktığı görülür.	-
3	Hatalı Kullanıcı Bilgileri ile Oturum Açma	Kullanıcı giriş ekranında kullanıcı adı ve şifre bilgileri sistemde tanımlı olmayacak bir kullanıcı ile oturum açma denir ve kullanıcı adı ve şifre eşleşmediği için oturum açılmadı uyarısının çıktığı görülür.	-

#	Konu	Açıklama	Not
1	Profil Ekranının Açılması	Kullanıcı Profili ekranı menüden açılır ve oturum açan kullanıcı bilgilerinin görüntülediği görülür.	-

Şekil 3.48. Test Döngüsüne Dâhil Edilmeyen Test Durumları Görünümü.

Yeni Bulgular

ClearQuest ID	Konu	Açıklama	Tespit Edilen Test Durumu	Öncelik	Tip
	Ekran Yerleşimi	Okunabilirlik açısından ekranda bilgilerin alt alta değil ilgili olmalarına göre yanyana verilmelidir. Örneğin; Fotoğraf solda daha büyük diğer bilgiler sağ tarafta.	Yazılım Testleri 2 Kullanıcı Profili Ekranı Testleri - Kullanıcı Profili Bilgileri	Düşük	İyileştirme Önerisi

Şekil 3.49. Yeni Bulunan Bulguların Listesi.

Mevcut Bulgular

ClearQuest ID	Konu	Açıklama	Tespit Edilen Test Durumu	Öncelik	Tip
00056129	Oturum Açma Uyarısı	Kullanıcı girişi ekranında kullanıcı adı ve şifre girilmediğinde bu bilgilerin girilmesi gerektiği uyarısı çıkmamaktadır.	Yazılım Testleri 1 Kullanıcı Giriş Ekranı Testleri - Şifre Girmeden Oturum Açma	Düşük	Hata
00056130	Kullanıcı Profili Ekranında E-posta Bilgisinin Görüntülenmemesi	Kullanıcı profil ekranında e-posta adresinin görüntülenmediği görüldü.	Yazılım Testleri 1 Kullanıcı Profili Ekranı Testleri - Kullanıcı Profili Bilgileri	Yüksek	Hata

Şekil 3.50. Devam Eden Bulguların Listesi.

Doğrulanan Bulgular

ClearQuest ID	Konu	Açıklama	Tespit Edilen Test Durumu	Öncelik	Tip
00056130	Kullanıcı Profili Ekranında E-posta Bilgisinin Görüntülenmemesi Doğrulandı	Kullanıcı profili ekranında e-posta adresinin görüntülenmediği görüldü.	Yazılım Testleri 1 Kullanıcı Profil Ekranı Testleri - Kullanıcı Profil Bilgileri	Yüksek	Hata

Şekil 3.51. Doğrulanan Bulguların Listesi.



Şekil 3.52. Grafikler Görünümü.

Daha Önce Oluşturulan Raporlar

#	Ad	Döngü	Oluşturulma Tarihi
1	Yazılım Testleri 1	Yazılım Testleri 1	13 Mayıs 2019 Pazartesi
2	Henüz rapor oluşturulmadı.	Yazılım Testleri 2	-

Şekil 3.53. Daha Önce Oluşturulan Raporların Listesi.

3.1.4. Test Koştur

Yazılım testleri koşturma menüsü altında daha önce tanımlanan (bkz. Bölüm 3.1.3.3.) test döngülerinin koşturulması ve bulguların incelenmesi ekranlarından oluşur. Yazılım Test Planlama ve Raporlama aracının hata takip aracı olan IBM ClearQuest yazılımı ile tanımlanmış arayüzü aracılığıyla bulguların hem içeri hem de dışarı aktarılması sağlanmaktadır. Böylece bulguların senkronizasyonu sağlanmış olup raporlama için dış ortamda bulunan veri de içeri aktarılmış olmaktadır.

3.1.4.1. Test Koştur

Test koşturma menüsü daha önce tanımlanmış test dönülerinin (bkz. Bölüm 3.1.3.3.) koşturulması için gerekli ekranlardan oluşur. Testlerin koşturulacağı proje ve test

döngüsü seçilerek testleri koşturma ekranı açılır. (Bkz. Şekil 3.1.54.) Açılan ekranda test durumu, tanımı, ilişkili açık bulgular, durum bilgileri sunulduğu gibi diğer taraftan test durumu değiştirilebilir (bkz. Şekil 3.1.55.), yeni bulgular da eklenebilir. (Bkz. Şekil 3.1.56.) Eklenen bulgu test aracı veri tabanına kaydetmenin yanı sıra IBM ClearQuest aracı veri tabanına da aynı anda gönderilip kayıt işlemi yapılabilmektedir.

Test Durumu	Açıklama	Bağlı Bulgular	Sonuç	İşlemler
Sfite Girmeden Oturum Açma	Giriş ekranında sadece kullanıcı adı girilerek giriş yapma düğmesine basılır ve giriş yapılmadığı görülür.	<ul style="list-style-type: none"> mevcut bulgu SIVT: 00056129 Konu: Oturum Açma Uyarısı Durum: Çözüldü 	Devam Ediyor	+ Yeni Bulgu
Kullanıcı Profili Bilgileri	Kullanıcı profili ekranı açılır ve kullanıcıların; <ul style="list-style-type: none"> * Fotoğrafı * Kullanıcı adı * Unvanı * Bağlı olduğu mudurluğu * Telefon numarası * E-Posta adresi bilgilerinin gösterildiği görülür.	<ul style="list-style-type: none"> mevcut bulgu SIVT Konu: Ekran Yerleşimi Durum: Yeni 	Kaldı	+ Yeni Bulgu

Şekil 3.54. Test Koşturma Ekranı.

Test Durumu	Açıklama	Bağlı Bulgular	Sonuç	İşlemler
Sfite Girmeden Oturum Açma	Giriş ekranında sadece kullanıcı adı girilerek giriş yapma düğmesine basılır ve giriş yapılmadığı görülür.	<ul style="list-style-type: none"> mevcut bulgu SIVT: 00056129 Konu: Oturum Açma Uyarısı Durum: Çözüldü 	Devam Ediyor	+ Yeni Bulgu

Şekil 3.55. Test Durumu Durum Güncelleme Ekranı.

Kayıt Ekle

Tespit Edilen Tarih: 09/23/2019

Tip: Hata

Öncelik: Orta

Konfigürasyon Birimi: -----

Konu:

En fazla 100 karakter

Açıklama:

En fazla 500 karakter

+ Oluştur + Oluştur & CQ'ye Gönder İptal

Şekil 3.56. Değişiklik Kaydı Ekleme Ekranı.

Daha önce IBM ClearQuest veri tabanına girilmiş ve Yazılım Test Planlama ve Raporlama aracında yer alan herhangi bir bulgu test koşusu esnasında “Çözüldü” durumunda ise bu değişiklik kaydı için test koşturma ekranında değişiklik kaydının IBM ClearQuest aracından içeriği ve durum bilgisi alınarak güncellenmesi, “Atandı” durumuna çekilmesi, “Atandı” durumuna çekip IBM ClearQuest aracına gönderilmesi, “Doğrulandı” durumuna çekilmesi ve “Doğrulandı” durumuna çekip IBM Clearquest aracına gönderilmesi işlemleri yapılabilmektedir. (Bkz. Şekil 3.57.)

Bulgu Detayı
×

Durum	Tip	Öncelik	Tespit Edilme Tarihi
Çözüldü	Hata	Düşük	13 Mayıs 2019

Konfigürasyon Birimi	Clearquest ID
test_yaz_konf1 / test_yaz_konf1	00056129

Tespit Edilen Test Durumu	Doğrulan Test Durumu
Yazılım Testleri 1 - Şifre Girmeden Oturum Açma	-

Detaylar

Konu: Oturum Açma Uyarısı

Açıklama: Kullanıcı girişi ekranında kullanıcı adı ve şifre girilmediğinde bu bilgilerin girilmesi gerektiği uyarısı çıkmamaktadır.

Not: -

↓ CQ'den Al
✖ Ata
✖ Ata & CQ'ye Gönder
✓ Doğrula
✓ Doğrula & CQ'ye Gönder

Kapat

Şekil 3.57. Bulgu Atama, Doğrulama Ekranı.

3.1.4.2. Bulgu

Bulgu menüsü daha önce koşturulmuş test döngülerinde (bkz. Bölüm 3.1.4.1.) tespit edilen bulguların sunulduğu ekranlardan oluşur. Bulgular proje, konfigürasyon birimi, tip, öncelik, ClearQuest ID'si, konu ve durum alanlarına göre filtrelenerek gösterilir. (Bkz. Şekil 3.1.58)

#	CQ ID	Durum	Öncelik	Konu	Açıklama	İşlemler
1	00056129	Çözüldü	↑ Düşük	Oturum Açma Uyarısı	Kullanıcı girişi ekranında kullanıcı adı ve şifre girilmediğinde bu bilgilerin girilmesi gerektiği uyarısı çıkmamaktadır.	İşlemler
2	00056130	Doğrulandı	↑ Yüksek	Kullanıcı Profili Ekranında E-posta Bilgisinin Görüntülenmemesi	Kullanıcı profili ekranında e-posta adresinin görüntülenmediği görüldü.	İşlemler
3		Yeni	↑ Düşük	Ekran Yerleşimi	Okunabilirlik açısından ekranda bilgilerin alt alta değil olmalarına göre yarıyana verilmelidir. Örneğin, Fotoğraf solda daha büyük diğer bilgiler sağ tarafta.	İşlemler

Şekil 3.58. Değişiklik Kayıtları Listesi.

Güncelle: Seçili elemanı güncelleme ekranını açar. (Bkz. Şekil 3.59)

Kayıt Güncelle
×

Tespit Edilen Test Durumu:

Yazılım Testleri 1 - Şifre Girmeden Oturum Açma

Konfigürasyon Birimi:

test_yaz_konf1 / test_yaz_konf1

Tespit Edilen Tarih:

05/13/2019

Tip:

- İyileştirme Önerisi
- Hata

Öncelik:

- Düşük
- Orta
- Yüksek

CQ ID:

00056129

Maximum 200 character

Konu:

Oturum Açma Uyarısı

En fazla 100 karakter

Açıklama:

Kullanıcı girişi ekranında kullanıcı adı ve şifre girilmediğinde bu bilgilerin girilmesi gerektiği uyarısı çıkmamaktadır.

En fazla 500 karakter

Durum:

- Yeni
- Reddedildi
- Çözüldü
- Doğrulandı
- Diğer

Çözüm Sürümü:

Doğrulan Test Durumu:

Not:

En fazla 200 karakter

İptal
Kaydet

Şekil 3.59. Bulgu Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.60.) Onay verildikten sonra silme işlemi gerçekleşir.



Şekil 3.60. Bulgu Silme Ekranı.

ClearQuest Kaydı Oluştur: Seçili elemanın IBM ClearQuest hata takip aracında kaydı açılmamış ise; IBM ClearQuest'te yeni kayıt oluşturma işlemi gerçekleşir. Kayıt oluşturulduktan sonra Yazılım Test Planlama ve Raporlama aracındaki kayıta IBM ClearQuest aracının verdiği ID bilgisi de otomatik güncellenerek kaydedilir.

Clearquest'ten Al: Seçili elemanın IBM ClearQuest hata takip aracında kaydı açılmış ise; IBM ClearQuest'te kayda ilişkin yapılan değişiklikleri içeri aktarma işlemi gerçekleşir.

ClearQuest'e Gönder: Seçili elemanda yapılan değişikliklerin IBM ClearQuest hata takip aracına aktarılması işlemini gerçekleştirir.

3.1.5. Test Günlüğü

Test döngüsü planlama menüsü konfigürasyon birimi ve entegrasyon testlerinin planlanması için gerekli ekranlarından oluşur. Planlanması yapılan testlerin gösterimi ana sayfadaki takvim üzerinde plan olduğu fark edilecek biçimde gösterilir. Böylece gerçekleşen test koşullarından ayrılır ve kaynak planlamasının etkin bir şekilde yapılması sağlanır. (Bkz. Şekil 3.1)

3.1.5.1. Konfigürasyon Birimi Test Kaydı

Konfigürasyon birimi test kaydı menüsü açıldığında proje seçimi yapıldıktan sonra veri tabanında kayıtlı test kayıtları listelenmektedir. (Bkz. Şekil 3.61)

Konf. Birim Adı	Sürüm	Başlangıç Tarihi	Bitiş Tarihi	Sorumlu	İşlemler
test_yaz_konf1	1.0	2019-06-05	2019-06-07	mikasapoglu	Güncelle Sil

Şekil 3.61. Konfigürasyon Birimi Test Kayıtları Genel Görünüm.

Ekle: Yeni konfigürasyon birimi test kaydı ekleme ekranını açar. (Bkz. Şekil 3.62) Konfigürasyon birimi, sürüm, başlangıç ve bitiş zamanı, testlerden sorumlu kişi, test tipi, altyüklenici kullanım durumu, test tanımlarının güncellenme durumu, test takip dokümanı oluşturulma durumu, test raporu oluşturma durumu, test raporu dizini, test raporunun duyurulma durumu bilgileri girildikten sonra başta seçilen proje için ekleme yapılır. Eklenen tüm konfigürasyon birimi test kayıtları ana sayfadaki takvim üzerinde başlangıç ve bitiş tarihine göre gösterilir. (Bkz. Şekil 3.1)

Kayıt Ekle ×

Konfigürasyon Birimi:

----- ▼

Sürüm:

En fazla 100 karakter

Başlangıç Tarihi:

Sorumlu:

En fazla 200 karakter

Test Stratejisi:

Yineleme ▼

Altyüklenici Kullanım Durumu:

Test ▼

Yazılım Test Tanımı Güncellendi mi?

Bilinmiyor ▼

Test Durum Dokümanı Oluşturuldu mu?

Bilinmiyor ▼

Test Durum Dokümanı Dizin Yolu:

En fazla 200 karakter

Yazılım Test Raporu Oluşturuldu mu?

Bilinmiyor ▼

Yazılım Test Raporu Dizin Yolu:

En fazla 200 karakter

Yazılım Test Raporu Duyuruldu mu?

Bilinmiyor ▼

Tamamlanma Tarihi:

Şekil 3.62. Konfigürasyon Birimi Test Kaydı Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. (Bkz. Şekil 3.63)

Kayıt Güncelle

Konfigürasyon Birimi:
test_yaz_konf1 / test_yaz_konf1

Sürüm:
1.0
En fazla 100 karakter

Başlangıç Tarihi:
06/05/2019

Sorumlu:
mkasapoglu
En fazla 200 karakter

Test Stratejisi:
Tüm Test

Altyüklenici Kullanım Durumu:
Test

Yazılım Test Tanımı Güncellendi mi?
Hayır

Test Durum Dokümanı Oluşturuldu mu?
Hayır

Test Durum Dokümanı Dizin Yolu:
En fazla 200 karakter

Yazılım Test Raporu Oluşturuldu mu?
Hayır

Yazılım Test Raporu Dizin Yolu:
En fazla 200 karakter

Yazılım Test Raporu Duyuruldu mu?
Hayır

Tamamlanma Tarihi:
06/07/2019

Şekil 3.63. Konfigürasyon Birimi Test Kaydı Güncelleme Ekranı.

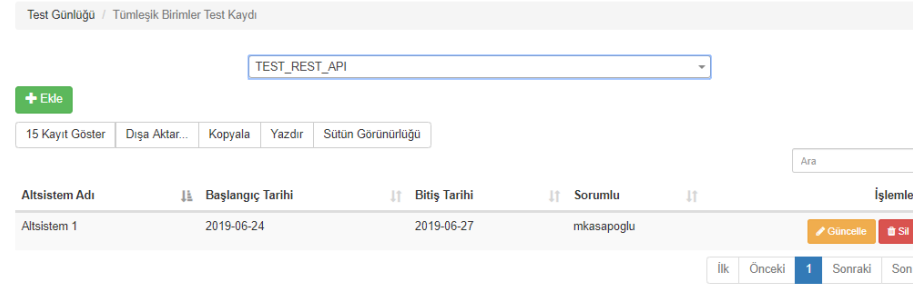
Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.64.) Onay verildikten sonra silme işlemi gerçekleşir. Silme işlemi tamamlandıktan sonra ilgili konfigürasyon birimi test kaydı ana sayfadaki takvim üzerinden de kaldırılır.



Şekil 3.64. Konfigürasyon Birimi Test Kaydı Silme Ekranı.

3.1.5.2. Tümlleşik Birimler Test Kaydı

Tümlleşik birimler test kaydı menüsü açıldığında proje seçimi yapıldıktan sonra veri tabanında kayıtlı test kayıtları listelenmektedir. (Bkz. Şekil 3.65)



Şekil 3.65. Tümlleşik Birimler Test Kayıtları Genel Görünüm.

Ekle: Yeni tümlleşik birimler test kaydı ekleme ekranını açar. (Bkz. Şekil 3.66) Altsistem, başlangıç ve bitiş zamanı, testlerden sorumlu kişi, test tipi, test tanımlarının güncellenme durumu, test takip dokümanı oluşturulma durumu, test raporu oluşturma durumu, test raporu dizini, testler sonrasında toplantı yapılma durumu, ilgili yazılım sürümlerinin sistem test düzeneğine kurulma durumu bilgileri girildikten sonra başta seçilen proje için ekleme yapılır. Eklenen tüm tümlleşik birimler test kayıtları ana sayfadaki takvim üzerinde başlangıç ve bitiş tarihine göre gösterilir. (Bkz. Şekil 3.1)

Kayıt Ekle



Altsistem:

Maximum 100 character

Başlangıç Tarihi:

Sorumlu:

En fazla 200 karakter

Test Stratejisi:

Yazılım Test Tanımı Güncellendi mi?

Yazılım Test Durumu Dokümanı Oluşturuldu mu?

Toplatı Yapıldı mı?

Yazılım Test Raporu Dizin Yolu:

En fazla 200 karakter

Yazılımlar Sistem Test Düzenine Kuruldu mu?

Tamamlanma Tarihi:

İptal

Kaydet

Şekil 3.66. Tümeşik Birimler Test Kaydı Ekleme Ekranı.

Güncelle: Seçili elemanı güncelleme ekranını açar. (Bkz. Şekil 3.67)

Kayıt Güncelle ×

Altsistem:

Maximum 100 character

Başlangıç Tarihi:

Sorumlu:

En fazla 200 karakter

Test Stratejisi:

Yazılım Test Tanımı Güncellendi mi?

Yazılım Test Durumu Dokümanı Oluşturuldu mu?

Toplanı Yapıldı mı?

Yazılım Test Raporu Dizin Yolu:

En fazla 200 karakter

Yazılımlar Sistem Test Düzenine Kuruldu mu?

Tamamlanma Tarihi:

Şekil 3.67. Tümleşik Birimler Test Kaydı Güncelleme Ekranı.

Sil: Seçili elemanın silinmesi için onay ekranını açar. (Bkz. Şekil 3.68.) Onay verildikten sonra silme işlemi gerçekleşir. Silme işlemi tamamlandıktan sonra ilgili tümleşik birimler test kaydı ana sayfadaki takvim üzerinden de kaldırılır.



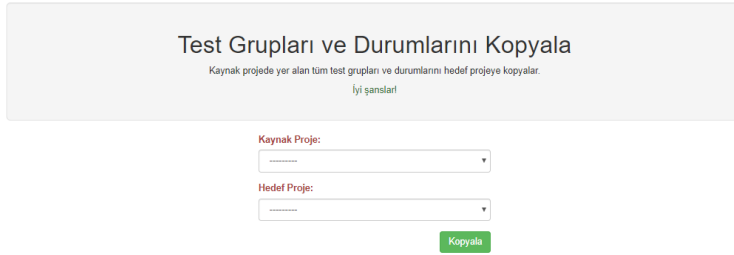
Şekil 3.68. Tümüleşik Birimler Test Kaydı Silme Ekranı.

3.1.6. Diğer İşlemler

Diđer işlemler menüsü daha çok sistem yöneticileri, ekip sorumluları için tanımlı işlemleri kolaylaştırmak amacıyla tasarlanan ekranlardan oluşur. Diđer ekranlarda tek tek yapılan işlemler topluca yapılarak işlemlerin hızlı bir şekilde tamamlanması sağlanır.

3.1.6.1. Test Grupları ve Durumlarını Kopyala

Test grupları ve durumlarını kopyalama menüsü açıldığında kaynak ve hedef proje seçim ekranı açılır. (Bkz. Şekil 3.69) Kopyalama işlemi tamamlandığında kaynak projedeki tüm test grupları ve bu test gruplarına bađlı test durumlarının tamamı hedef projeye kopyalanır.



Şekil 3.69. Test Grubu ve Durumları Kopyalama Ekranı.

3.1.6.2. Altsistem ve Konfigürasyon Birimlerini İçeri Al

Altsistemler ve konfigürasyon birimlerini içeri aktarma menüsü açıldığında IBM ClearQuest veri tabanında bulunan projeler alınarak proje seçim ekranı açılır. (Bkz. Şekil 3.70)

Diğer İşlemler / Altsistem ve Konfigürasyon Birimlerini İçeri Al

Altsistem ve Konfigürasyon Birimlerini İçeri Al

Seçili proje için ClearQuestte tanımlı tüm altsistem ve konfigürasyon birimlerini içeri alır.
İyi Şanslar!

Proje seçiniz...

Ad:

En fazla 200 karakter

Kod:

Dosya Yolu:

En fazla 200 karakter

Şekil 3.70. Altsistem ve Konfigürasyon Birimi İçeri Aktarma Ekranı.

IBM ClearQuest hata takip aracında bulunan ve Yazılım Test Planlama ve Raporlama aracına aktarılmak istenen proje seçilerek içeri aktarma işlemi tamamlandığında o proje altında yer alan tüm alt sistemler ve bağlı tüm konfigürasyon birimleri aktarılarak kullanıcı bilgilendirilir. (Bkz. Şekil 3.71)

Diğer İşlemler / Altsistem ve Konfigürasyon Birimlerini İçeri Al

Başarıyla tamamlandı!

2 altsistem ve 5 konfigürasyon birimi içeri alındı.

Şekil 3.71. Altsistem ve Konfigürasyon Birimi İçeri Aktarma Bilgilendirme Ekranı.

3.1.6.3. Bulguları Güncelle

Bulguları güncelle menüsü açıldığında Yazılım Test Planlama ve Raporlama aracında bulunan tüm projelerin listesini içeren proje seçim ekranı açılır. (Bkz. Şekil 3.72)

Diğer İşlemler / Bulguları Güncelle

Bulguları Güncelle

Seçili proje için girilmiş tüm bulguların Clearquestteki son halini içeri alır.
İyi şanslar!

Yazılım Test Planlama ve Raporlama Aracı ▼ Tüm Bulguları Güncelle

Şekil 3.72. Bulguları Güncelleme Ekranı.

Bu ekrandan proje seçimi yapıldıktan sonra ilgili projede daha önce açılmış değişiklik kayıtlarının içeriği IBM ClearQuest hata takip aracından alınarak Yazılım Test Planlama ve Raporlama aracı veri tabanındaki bilgiler güncellenir. Yapılan işlemlerin sonuçları ile ilgili kullanıcı bilgi ekranı gösterilir. (Bkz. Şekil 3.73)

Bulguları Güncelle

Seçili proje için girilmiş tüm bulguların Clearquest'teki son halini içeri alır.
İyi şanslar!

Clearquest ID	Konu	Sonuç	Sonuç Açıklaması
00056129	Oturum Açma Uyarısı	✓	Başarıyla güncellendi.
00056130	Kullanıcı Profili Ekranında E-posta Bilgisinin Görüntülenmemesi	✓	Başarıyla güncellendi.
00056131	Profil Fotoğrafının Çözünürlüğü	✓	Başarıyla güncellendi.
	Ekran Yerleşimi	✗	CRRCM9996E: Kaynak bulunamadı.

Şekil 3.73. Bulguları Güncelleme Bilgilendirme Ekranı.

4. SONUÇLAR VE TARTIŞMA

Yazılım ürününü test etmek iyi planlanması ve belirlenen kriterlere göre standartlaştırılması gereken önemli ve ciddi bir süreçtir. Bu amaçla sunulan tez çalışmasında geliştirilen Yazılım Test Planlama ve Raporlama Aracı, yazılım geliştirme sürecinde yer alan tüm paydaşlar için standartlar ve yönetim kolaylığı sağlayan özel amaçlı yazılım test yönetim ve raporlama aracıdır.

Yazılım Test Planlama ve Raporlama Aracı aşağıdaki yeteneklere sahiptir:

- Kullanıcı dostu grafik arayüz tasarımı
- LDAP sunucusu aracılığıyla kullanıcı yetkilendirme
- Araç içeriğine yetki kontrolü ile erişim
- Tüm tablo gösterimlerinde filtreleme, seçili kolona göre çıktı alma, .pdf, .csv, ve .xlsx dosyası olarak çıktı alma
- Tüm tablo gösterimlerinde yazıcı dostu çıktı alma
- LDAP sunusundan kullanıcı detay bilgilerinin alınması ve bu bilgilerin yazılım test duyuruları gibi işlemlerin yapılmasında kullanılması
- Proje özelinde yetki kontrolü (sadece yetkilendirilmiş kullanıcıların o proje ile ilgili içeriğe erişebilmesi)
- Ana sayfada takvim üzerinde planlanan ve gerçekleşen test koşullarının aylık olarak gösterilmesi
- Proje, Altsistem, Konfigürasyon Birimi bilgilerinin girilmesi, düzenlenmesi
- Proje, Altsistem, Konfigürasyon Birimi bilgilerinin IBM ClearQuest yazılımından içeri aktarılması
- Sürüm bilgilerinin girilmesi ve düzenlenmesi
- Teste hazır olma kriter başlıklarının ve kriterlerinin girilmesi, düzenlenmesi
- Konfigürasyon Birimi sürümlerinin teste hazır olma durumlarının değerlendirilmesi
- Proje özelinde test gruplarının girilmesi, düzenlenmesi
- Proje özelinde test gruplarına bağlı test durumlarının girilmesi, düzenlenmesi

- Proje için artık kullanılmayan test gruplarının ve durumlarının arşivlenmesi
- Test grupları ve durumlarının kaynak projeden hedef projeye kopyalanması
- Test döngülerinin başlangıç ve bitiş tarihi ve konfigürasyon birimi seçilerek oluşturulması
- Test döngüsüne projede tanımlı test durumlarının eklenilmesi ve çıkarılması
- Test döngüsünün kopyasının oluşturulması
- Test döngüsü için test raporu oluşturulması
- Test döngülerinin koşturulması
- Test döngüsünde tespit edilen değişiklik kayıtlarının aracın veri tabanına kaydı ile birlikte IBM ClearQuest veri tabanına da aktarılması
- Test döngüsü sırasında IBM ClearQuest'de bulunan değişiklik kaydının içeri aktarılması, güncellenmesi
- Test döngüsünde test durumları ile ilgili daha önce girilmiş o anda yeni ya da çözülmüş durumda olan bulguların listesinin verilmesi
- Test koştururken tek tıkla hata doğrulama
- Test raporlarının tek tıkla oluşturulması
- Test raporunda genel durum, test sonuçları, kayıt altına alınan bulgular, doğrulanmamış bulgular, doğrulanan bulgular, test durumlarının önceki test sonuçlarıyla birlikte grafiksel gösterimi, tespit edilen bulgu sayısının grafiksel gösterimi ve o proje için daha önce oluşturulan test raporlarının listesi bilgilerinin sunulması

Diğer taraftan Yazılım Test Planlama ve Raporlama Aracı'nın daha etkin kullanılması için geliştirilmesi aşağıdaki alanlarda yetenekleri geliştirilmelidir:

- Test tanımlarını oluştururken yazılım gereksinim özelliklerine izlenebilirliklerin kurulması
- Yazılım test süreçlerinde yaygın olarak kullanılan diğer hata takip araçları (Jira, Redmine, GitLab, vb.) ile entegre çalışması
- Yazılım test raporlarında önceki test sonuçları bilgilerinin de verilmesi

- Yazılım test raporunda daha gelişmiş grafiklerin (bulgu – test durumları ısı haritası, sıklıkla hata tespit edilen test durumlarının gösterimi, vb.) bulunması

5. KAYNAKLAR

- [1] O. Legunsen, F. Hariri, A. Shi, Y. Lu, L. Zhang, and D. Marinov, “An extensive study of static regression test selection in modern software evolution,” *Proc. 2016 24th ACM SIGSOFT Int. Symp. Found. Softw. Eng. - FSE 2016*, pp. 583–594, 2016.
- [2] R. Potvin and J. Levenberg, “Why Google stores billions of lines of code in a single repository,” *Commun. ACM*, vol. 59, no. 7, pp. 78–87, Jun. 2016.
- [3] S. Biswas, R. Mall, M. Satpathy, and S. Sukumaran, “Regression test selection techniques: A survey,” *Inform.*, vol. 35, no. 3, pp. 289–321, 2011.
- [4] ISTQB, Certified Tester Foundation Level Syllabus. p. 13, 2018.
- [5] Pentti Pohjolainen, Software Testing Tools. University of Kuopio, 2002.
- [6] G. Rothermel and M. J. Harrold, “A framework for evaluating regression test selection techniques,” *Proc. 16th Int. Conf. Softw. Eng.*, pp. 201–210, 1994.
- [7] R. Kazmi, D. N. A. Jawawi, R. Mohamad, and I. Ghani, “Effective Regression Test Case Selection: A Systematic Literature Review” *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–32, 2017.
- [8] ISTQB, Certified Tester Foundation Level Syllabus. p. 15, 2018.
- [9] Lee Copeland, A Practitioner's Guide to Software Test Design, Artech House, 2004
- [10] Pentti Pohjolainen, “Software Testing Tools” University of Kuopio, 2002.
- [11] ISTQB, Certified Tester Foundation Level Syllabus. p. 71, 2018.
- [12] Atsushi Yamada, ISO/IEC 9126-2 TR: Software Engineering– Product quality –Part 2 : External Metrics, 2000.
- [13] Anonim, Why Django, <https://www.djangoproject.com/start/overview/> (Erişim tarihi: **1 Ağustos 2019**).
- [14] Anonim, Bootstrap, <https://getbootstrap.com/> (Erişim tarihi: **1 Ağustos 2019**).
- [15] Anonim, What is jQuery?, <https://jquery.com/> (Erişim tarihi: **1 Ağustos 2019**).
- [16] Anonim, What is MySQL?, <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html> (Erişim tarihi: **1 Ağustos 2019**).
- [17] Anonim, Why Choose LDAP?, <https://ldap.com/why-choose-ldap/> (Erişim tarihi: **1 Ağustos 2019**).

1 Ağustos 2019).

[18] Anonim, FullCalendar, <https://fullcalendar.io/> (Erişim tarihi: **1 Ağustos 2019**).

[19] Anonim, Using DataTables, <https://datatables.net/> (Erişim tarihi: **1 Ağustos 2019**).

[20] Anonim, Highcharts, <https://www.highcharts.com/blog/products/highcharts/> (Erişim tarihi: **1 Ağustos 2019**).



HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/~~DOKTORA~~ TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 25/09/2019

Tez Başlığı / Konusu: YAZILIM TEST PLANLAMA VE RAPORLAMA ARACI'NIN TASARLANMASI VE KODLANMASI

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam 53 sayfalık kısmına ilişkin, 25/09/2019 tarihinde ~~çalışmam~~/tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 2'dir.

Uygulanan filtrelemeler:

- 1- Kaynakça hariç
- 2- Alıntılar dâhil

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygularıyla arz ederim.

25/09/2019

Adı Soyadı: Mustafa Kasapoğlu
Öğrenci No: N16120618
Anabilim Dalı: İstatistik
Programı: İstatistik
Statüsü: Y.Lisans Doktora Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

İbrahim Zor

Dr. Öğretim Üyesi İbrahim Zor

ÖZGEÇMİŞ

Adı Soyadı : Mustafa KASAPOĞLU
Doğum yeri : Epernay/FRANSA
Doğum tarihi : 14.10.1985
Medeni hali : Evli
Yazışma adresi : Aselsan Gölbaşı Yerleşkesi Konya Yolu 8.km Oğulbey Mahallesi 3051. Sokak No:3 06830 Gölbaşı/Ankara Türkiye
Telefon : +90 312 592 60 00
Elektronik posta adresi : mkasapoglu@hacettepe.edu.tr
Yabancı dili : İngilizce

EĞİTİM DURUMU

Lisans : Hacettepe İstatistik

İş Tecrübesi

2015 – ... ASELSAN A.Ş. Ankara, Türkiye

Yazılım Test Kıdemli Uzmanı

2010 – 2015 MİKES A.Ş. Ankara, Türkiye

Yazılım Test Uzmanı

