

**T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

BSM 498 BİTİRME ÇALIŞMASI

TEST OTOMASYON PROJESİ

G171210113 – Dilek Ayşe GÜN

G171210017 – Tuğba ULUSOY

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ
Tez Danışmanı : Doç. Dr. Ünal ÇAVUŞOĞLU**

2020-2021 Bahar Dönemi

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

TEST OTOMASYON PROJESİ

BSM 498 - BİTİRME ÇALIŞMASI

Dilek Ayşe GÜN
Tuğba ULUSOY

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

.....
Jüri Başkanı

.....
Üye

.....
Üye

ÖNSÖZ

Yazılım testi ile yazılım geliştirme sürecinde herhangi bir hata varsa, erken tespit edilebilir ve yazılım ürünü müşteriye teslim edilmeden önce hatalar çözülebilir. Düzgün bir şekilde test edilen yazılım ürünü güvenilirlik, güvenlik ve yüksek performans sağlar. Bu da zaman tasarrufu, maliyet kazancı ve müşteri memnuniyeti ile sonuçlanır. Yazılım testinin en savunmasız ve hassas yararı güvenlidir. İnsanlar güvenilir ürünler arıyor. Yazılım testi, riskleri ve sorunları daha erken gidermeye yardımcı oluyor. Herhangi bir yazılım ürününün olmazsa olmazı ürün kalitesidir. Test, müşterilere kaliteli bir ürün teslim edilmesini sağlar. Herhangi bir ürünün temel amacı müşterilerine memnuniyet vermektir. UI / UX Testi, en iyi kullanıcı deneyimini sağlar. Yazılım testinin en önemli avantajlarından biri de maliyet tasarrufudur. Herhangi bir BT projesini zamanında test etmek, uzun vadede maddi açıdan da tasarruf etmeye yardımcı olur. Yazılım testinin önceki aşamasında hataların yakalanması durumunda anında düzeltilmesi daha az maliyetlidir. Aynı zamanda yazılım hataları pahalı ve hatta tehlikeli olabileceğinden projeleri test etmek önemlidir. Yazılım hataları potansiyel olarak maddi ve de insani kayba neden olabilir. Tarihte bu türde birçok örneğe ulaşmak mümkündür.

İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER LİSTESİ.....	viii
TABLolar LİSTESİ.....	xi
ÖZET.....	xii
BÖLÜM 1.	
GİRİŞ.....	1
1.1. Selenium Nedir? Niçin Kullanılır?.....	1
1.2. Cucumber.....	2
BÖLÜM 2.	
TEST OTOMASYON PROJESİ İŞLEYİŞİ.....	4
2.1. Ögeyi Denetle(Inspect) İle Yol Belirteçleri.....	7
2.2. Branch İşlemi.....	8
2.3. Projede Kullanılan Class ve Dosyalar.....	9
2.4. Commit İşlemi.....	12
BÖLÜM 3.	
JUNIT ASSERTIONS.....	14
3.1. JUnit Assertions Tipleri.....	14
BÖLÜM 4.	
FIND() METHODU.....	16

BÖLÜM 5.	
XPATH HELPER CHROME.....	17
5.1. Pop-up XPath'ini Bulma Adımları.....	18
BÖLÜM 6.	
ALERT POP-UP ONAYLAMA.....	20
BÖLÜM 7.	
PROJE SENARYOLARINI DÜZENLEME.....	21
7.1. Tek Feature Dosyasında Birden Fazla Senaryo Yazma.....	21
7.2. Feature Dosyasında Background Kullanmanın Faydası.....	22
BÖLÜM 8.	
PROJEDE KULLANILAN BAZI FONKSİYONLAR.....	23
8.1. ToLowerCase() Metodu.....	23
8.2. SelectByIndex() Fonksiyonu.....	23
8.3. ScrollToElement() Fonksiyonu.....	24
BÖLÜM 9.	
CLICK VS CLICKJS.....	25
BÖLÜM 10.	
PROJEYE DOSYA EKLEME.....	26
10.1. File.Separator.....	27
10.2. GetAbsolute() Metodu.....	27
BÖLÜM 11.	
CLASS VE TEXT İLE SEÇİCİ BULMA.....	28
11.1. Contains İle Class Bulma.....	28
11.2. Text İle Seçici XPath'ini Bulma.....	28
BÖLÜM 12.	
SELENIUM'DA İFRAME KULLANIMI.....	29

BÖLÜM 13.	
HATA OKUMA.....	31
 BÖLÜM 14.	
GİTHUB'DA PROJEYİ MERGE EDEBİLMEK İÇİN GEREKLİ ALT ADIMLAR.....	32
14.1. Commit And Push.....	32
14.2. Pull.....	33
 BÖLÜM 15.	
LİSTE KONTROL İŞLEMİ.....	34
 BÖLÜM 16.	
AÇILAN PENCEREYİ KAPATMA (MOVEBYOFFSET).....	36
16.1. Örnek.....	36
16.2. MoveByOffset Methodunun İncelenmesi.....	37
 BÖLÜM 17.	
RANDOM METHODU.....	38
 BÖLÜM 18.	
ATTRIBUTE.....	39
 BÖLÜM 19.	
RUNNER.....	40
19.1. Birden Çok Test Senaryosunu Çalıştırma.....	40
19.2. TestNG.....	42
 BÖLÜM 20.	
BUGFIX.....	43

BÖLÜM 21.	44
SONUÇLAR VE ÖNERİLER.....	45
KAYNAKLAR.....	46
ÖZGEÇMİŞ.....	47
BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	

ŞEKİLLER LİSTESİ

Şekil 1.1.	Test case' ler	1
Şekil 1.2.	Cucumber tool'u yüklenmesi.....	3
Şekil 2.1.	Kayıt işlemleri menüsünün test otomasyonu gerçekleştirildi.....	5
Şekil 2.2.	iKampüs web sitesi için yazılan test senaryoları.....	5
Şekil 2.3.	iKampüs test senaryolarının kısaltma kodları.....	7
Şekil 2.4.	Web sitesinde ögeyi denetle yol belirteci oluşturma.....	8
Şekil 2.5.	Branch sembolü.....	8
Şekil 2.6.	Yeni branch oluşturma.....	9
Şekil 2.7.	Constants class' ında değişkene Xpath'ler tanımlandı.....	10
Şekil 2.8.	Java kodları Page class' ında yazıldı.....	10
Şekil 2.9.	StepDefinations class' ı ve features dosyası oluşturuldu.....	11
Şekil 2.10.	Steps'te selenium methodları oluşturuldu.....	11
Şekil 2.11.	Feature dosyasında test senaryoları yazıldı.....	12
Şekil 2.12.	Commit işlemi.....	13
Şekil 3.1.	Assert kullanarak gelen mesajın kontrolü yapıldı.....	15
Şekil 4.1.	Dosya seçme işlemi.....	16
Şekil 4.2.	Constants' ta buton ve dosyanın yolunu belirtildi.....	16
Şekil 4.3.	Find() ile dosya seçme işlemi yapıldı.....	16
Şekil 5.1.	XPath Helper Chrome' u tarayıcıya ekleme işlemi.....	17

Şekil 5.2.	XPath Helper Chrome ile “SMS Gönder” butonunun Xpath’ inin doğruluğunu test etme işlemi.....	18
Şekil 5.3.	Pop-up XPath’ ini bulma.....	19
Şekil 5.4.	XPath Helper’ da XPath’in yazıldığı yere tıklayınca “Uyarı” kelimesinin bulunduğu textbox sarı renge dönüyor.....	19
Şekil 6.1.	Alert Pop-up.....	20
Şekil 6.2.	Alert Pop-up onaylama kodu.....	20
Şekil 7.1.	Yeni kayıt görüşmesi senaryosu tek fonksiyonda farklı olayların gerçekleşmesi.....	21
Şekil 7.2.	Tek feature dosyasında birden fazla senaryo yazma.....	21
Şekil 7.3.	Tek feature dosyasında birden fazla senaryo yazma.....	22
Şekil 8.1.	toLowerCase() methodunun kullanımı gösterilmektedir.....	23
Şekil 8.2.	selectByIndex() fonksiyonunun kullanımı gösterilmektedir...	23
Şekil 8.3.	selectByIndex() fonksiyonu ile index seçimi.....	23
Şekil 8.4.	scrollToElement() fonksiyonunun kullanımı gösterilmektedir.	24
Şekil 10.1.	Test-file klasöründe eklenen dosyalar görünmektedir.....	26
Şekil 10.2.	Eklenen dosyada sağ tık ile Absolute Path seçeneği seçilmesi.	26
Şekil 10.3.	File.separator methodunun kullanımı gösterilmektedir.....	27
Şekil 11.1.	“Eğitim Kurumu Ekle” butonunun seçicisi text() ile bulundu..	28
Şekil 12.1.	Selenium’ da iframe kullanımı.....	29
Şekil 12.2.	İframe XPath kodu.....	29
Şekil 12.3.	İframe Page class kodu.....	30
Şekil 12.4.	İframe PageSteps class kodu.....	30
Şekil 12.5.	İframe feature dosyası kodu.....	30
Şekil 13.1.	Mavi altı çizili class ismine basıldığında hatanın olduğu yere hızlı şekilde gidilebilir.....	31

Şekil 14.1.	Commit ve Push işlemi.....	32
Şekil 14.2.	Pull işlemi.....	33
Şekil 15.1.	“Add Admin School” senaryosunda eklenen kontrol cümleleri gösterilmektedir.....	34
Şekil 15.2.	Son kayıt numarasının alındığı ve Integer veri tipine dönüştürüldüğü fonksiyon gösterilmektedir.....	34
Şekil 15.3.	Eski kayıt numarasının alındığı fonksiyon gösterilmektedir....	35
Şekil 15.4.	Assertion ile, oldNumber ve newNumber sayıları arasındaki farkı gösterilen kontrol fonksiyonu gösterilmektedir.....	35
Şekil 16.1.	Doğum tarihi alanına tıklandığında açılan pencere gösterilmektedir.....	36
Şekil 16.2.	moveByOffset methodunun kullanımı gösterilmektedir.....	37
Şekil 17.1.	Page class’ ında “Random” tanımlaması.....	38
Şekil 17.2.	Random methodunun fonksiyondaki işleyişi.....	38
Şekil 18.1.	XPath’ lerde Attribute’ lar.....	39
Şekil 18.2.	Attribute’ lerin senaryoda kodlaması yapılırken getAttribute fonksiyonu ile kullanıldı.....	39
Şekil 19.1.	İstemediğimiz feature’ ların çalışmasını önleme.....	40
Şekil 19.2.	Suites klasöründe yeni oluşturulan TugbaDilekRunner.java tanımlandı.....	41
Şekil 19.3.	Runner klasörü altındaki TugbaDilekRunner.java çalıştırılması	41
Şekil 19.4.	Run işleminden sonra testte çıkan hatalar görüntüleniyor.....	42
Şekil 19.5.	Run işleminden sonra testte çıkan hatalar görüntüleniyor.....	42
Şekil 19.6	AllSuite.xml içerisine TestNG eklenmesi.....	42
Şekil 20.1.	Bugfix commit edildi.....	43

TABLÖLAR LİSTESİ

Tablo 1.1.	İş ilerleme tablosu.....	6
------------	--------------------------	---

ÖZET

Anahtar kelimeler: Yazılım Testi, Test Otomasyonu, Java, Selenium, Cucumber, Maven, TestNG

CAZ Bilişim Hizmetleri A.Ş. ürünlerinin testleri için Selenium üzerinde testler yazarak iKampüs web sitesi test edildi. Proje adı “Test Otomasyonu Projesi” dir. Proje Java tabanlı IntelliJ IDEA CE platformunda kodlanmıştır. Ürünlerin testleri için Selenium framework’ ü kullanıldı. Selenium, bir test komut dosyası dilini (Selenium IDE) öğrenmeye gerek kalmadan işlevsel testleri yazmak için bir oynatma aracı sağlar. Selenium için “Web tarayıcıları üzerinde test otomasyonu yazmamızı sağlayan araç” tanımını kullanabiliriz. “Test Otomasyonu” terimini basit şekilde tanımlayacak olursak, belirli test senaryolarının sürekli olarak programlar aracılığı ile koşulması diyebiliriz. Projede Selenium framework’ ü ile birlikte Cucumber tool u kullanıldı. Cucumber, davranış odaklı geliştirmeyi destekleyen bir tür open source tool dur. Projede ChromeDriver teknolojisinden faydalanıldı.Çalışma prensibi olarak Agile methodology, Sprintler ve günlük toplantılar (daily) eşliğinde düzenli bir çalışma ortamı deneyimi elde edildi.

BÖLÜM 1. GİRİŞ

Bir uygulamanın kullanıcılara ulaşabilmesi ve kullanılmaya başlanması için önce bu uygulamanın test ekibi tarafından incelenmesi gerekir. Uygulama kullanışlı mı? Sayfada absürt bir şey var mı? Butonlar doğru çalışıyor mu ve yapması gereken işlevi gerçekleştiriyor mu? Sayfa nasıl daha estetik ve kullanışlı hale getirilebilir? Kullanıcı bakar bakmaz şu buton şu işi yapıyor diyebiliyor mu yoksa yapılmak istenen iş için çok mu incelemek gerekiyor? Dikkat ve detaycı bir bakış açısıyla bakmayı gerektiren bir alandır. Test ekibi yazılım ekibiyle iletişim halinde çalışır. Yazılım tarafında yapılan değişiklikler test ekibine haber verilir böylece test edilmiş kısımlarda yapılan düzeltmelere göre yeniden test yapmak daha kolay olmaktadır.

1.1 Selenium Nedir? Niçin Kullanılır ?

Selenium için “Web tarayıcıları üzerinde test otomasyonu yazmamızı sağlayan araç” tanımını kullanabiliriz. “Test Otomasyonu” terimini basit şekilde tanımlayacak olursak ise, belirli test senaryolarının sürekli olarak programlar aracılığı ile koşulması diyebiliriz.[6]

Örneğin, programımızda bir Kullanıcı Adı, Şifre alanları ile Giriş butonunun bulunduğu bir Login ekranı ve bu Login ekranı ile ilgili en basit şekilde aşağıdaki test caseleri mevcut olsun;

Case No	Test Adımı	Beklenen Sonuç	Durumu
1	Kullanıcı Adı ve Şifre boş iken siteye giriş yapılabilir mi?	Kullanıcı Adı ve Şifre boş iken sisteme giriş yapılamaz.	Başarılı
2	Kullanıcı Adı dolu ve hatalı, Şifre boş iken siteye giriş yapılabilir mi?	Kullanıcı Adı dolu ve hatalı, Şifre boş iken siteye giriş yapılamaz.	Başarılı
3	Kullanıcı Adı dolu ve hatalı, Şifre doğru iken siteye giriş yapılabilir mi?	Kullanıcı Adı dolu ve hatalı, Şifre doğru iken siteye giriş yapılamaz.	Başarılı
4	Kullanıcı Adı boş, Şifre dolu ve hatalı iken siteye giriş yapılabilir mi?	Kullanıcı Adı boş, Şifre dolu ve hatalı iken siteye giriş yapılamaz.	Başarılı
5	Kullanıcı Adı doğru, Şifre boş iken sisteme giriş yapılabilir mi?	Kullanıcı Adı doğru, Şifre boş iken sisteme giriş yapılamaz.	Başarılı
6	Kullanıcı Adı doğru, Şifre dolu ve hatalı iken sisteme giriş yapılabilir mi?	Kullanıcı Adı doğru, Şifre dolu ve hatalı iken sisteme giriş yapılamaz.	Başarılı
7	Kullanıcı Adı dolu ve hatalı, Şifre dolu ve hatalı iken siteye giriş yapılabilir mi?	Kullanıcı Adı dolu ve hatalı, Şifre dolu ve hatalı iken siteye giriş yapılamaz.	Başarılı
8	Kullanıcı Adı doğru ve Şifre doğru iken sisteme giriş yapılabilir mi?	Kullanıcı Adı doğru ve Şifre doğru iken sisteme giriş yapılabilir.	Başarılı

Şekil 1.1. Test case'ler

Yukarıda Login ekranı ile ilgili en basit şekilde yazmamıza rağmen sekiz adet case ortaya çıktı. Daha karmaşık ekranlar için yüzlerce hatta binlerce sayıda caseler ortaya çıkabiliyor.

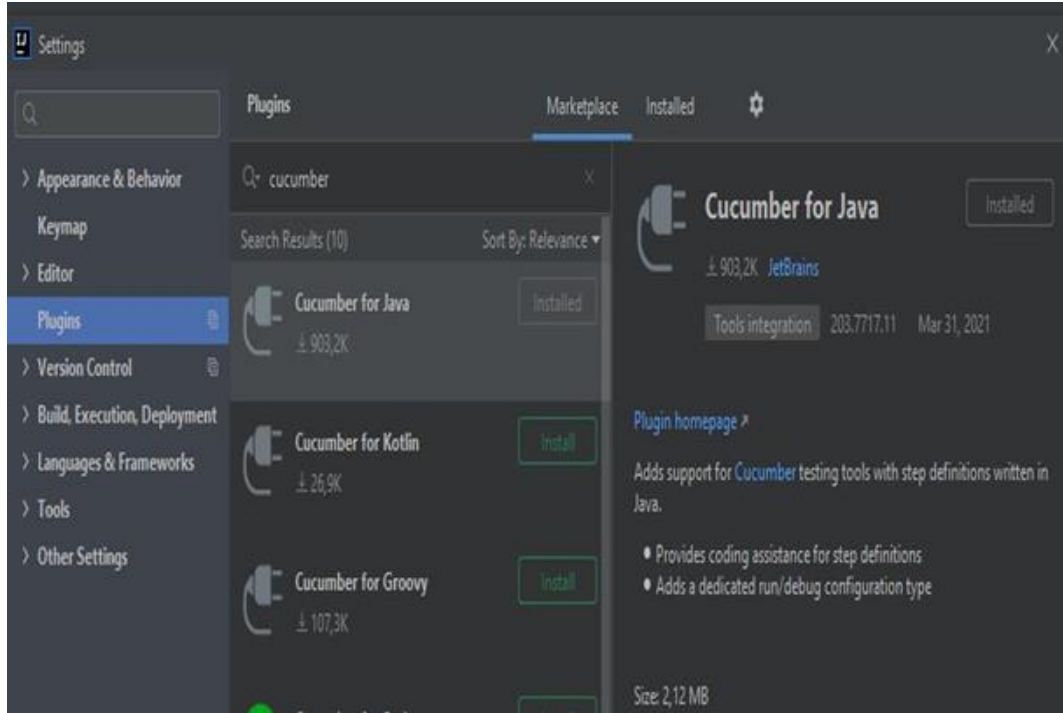
Login ekranımızda bir değişiklik yapıyoruz, yukarıdaki koşulların değişmemesi gerek ve bunların kontrolünü sağlamak zorundayız. Aynı test caselerinin sürekli manuel olarak koşulması gereksiz enerji, kaynak kullanımına sebep olabiliyor. Ancak Selenium ile testler otomatize edilebiliyor. Selenium’da test caselerinin girdilerini ve olması gereken çıktıları belirledikten sonra aynı caselerin sürekli olarak aynı şekilde program tarafından koşulmasını isteyebilir ve çıktılarımızı alabiliriz.

Bu şekilde test caseleri arasında yaptığımız çalışmadan sonra değişiklik olup olmadığını görebiliriz.

Yazılım dünyasından basit bir benzetme yapacak olursak ise, farklı sayfalarda aynı işi yapan kodların tekrar tekrar yazılmaması için bir class oluşturularak gerekli sayfalarda classımızın çağrılması da diyebiliriz.

1.2. Cucumber

Cucumber; Behavior Driven Development (BDD) destekleyen bir API dir. Selenium/Appium da oluşturduğunuz method larınızı cümlelere çevirerek hem test scriptlerinizi dokümante edebilir hem de kod bilgisi olmayan kişiler ile birlikte test otomasyon case leri yazma imkanı sağlıyor. Cucumber ile test case ler Gherkin dili ile yazılıyor. Steps klasöründe selenium methodları oluşturuldu, Feature klasöründe de test senaryoları yazıldı. [1]



Şekil 1.2. Cucumber tool'u yüklenmesi

File -> Settings -> Plugins -> Cucumber for Java seçilerek plugin indirildi.

Constants klasöründe web sitedeki buton, text alanı vb. araçların yolları bir değişkene atılıp, burada tanımlanıyor.

BÖLÜM 2. TEST OTOMASYON PROJESİ İŞLEYİŞİ

iKampüs web sitesinin (<https://demo.ikampus.com/>) test otomasyonunu gerçekleştirildi. İntellij IDEA CE programında Java diliyle senaryoları kodlandı.

Test Senaryoları bizimle paylaşılan excel dosyasında oluşturuldu. Test Senaryosu yazarken bu başlıklar olmalı: Test Senaryo No, Test Senaryo Adı, Test Adım No, Test Adımları, Beklenen Sonuç, Test Verisi, Ön Koşullar, Açıklama, Sonuç.

Test Senaryo No: Senaryonun baş harflerinin kısaltılması.

Test Senaryo Adı: Senaryonun adı.

Test Adım No: Senaryodaki adımların numaralandırılması.

Test Adımları: Senaryoların cümleler halinde adım adım, açıkça belirtilmesi.

Beklenen Sonuç: Test sonucunda beklenen sonuç.

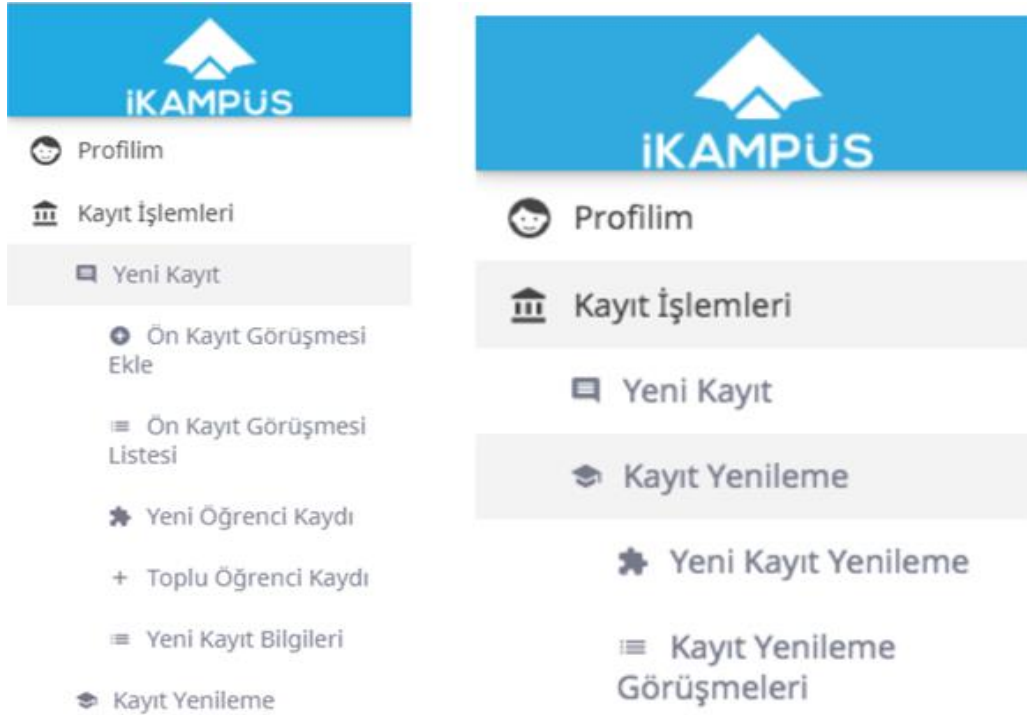
Test Verisi: Test edilirken hangi verileri yazacağımızı belirtiyoruz.

Ön Koşullar: Senaryo öncesi yapılması gereken adım varsa yazılır.

Açıklama: Senaryoya ait detaylı açıklama varsa yazılır.

Sonuç: Senaryonun test edilmesi sonucu ne olacaksa belirtiyoruz.

Hafta içi her gün düzenli olarak meets toplantıları yapıldı. Neler yapıldı, neler yapılacak, sıradaki görevler neler vs. konuşuldu. Projedeki yapılanlar ekran paylaşımı yapılarak gösterildi. Departman müdürü tarafından İş İlerleme Durumu tablosu da takip edildi.



Şekil 2.1. Kayıt İşlemleri Menüsünün test otomasyonu gerçekleştirildi.

İKAMPÜS TEST SENARYOLARI									
Dosya Düzenle Görünüm Ekle Bıçım Veri Araçlar Eklenitler Yardım Son düzenleme birkaç saniye önce yapıldı									
Test Senaryo Adı									
Test Senaryo No	Test Senaryo Adı	Test Adım No	Test Adımları	Beklenen Sonuç	Test Verisi	Ön Koşullar	Açıklama	Sonuç	
TÖK-1	TOPLU ÖĞRENCİ KAYDI	1	Sol menüde bulunan "Kayıt İşlemleri" butonuna t						
		2	"Yeni Kayıt" butonuna tıklanmalı.	Kullanıcı "Kayıt İş					
		3	"Toplu Öğrenci Kaydı" butonuna tıklanmalı.	leştikten sonra "Ye	XLS dosyası seçilme	Kullanıcının ka	Toplu kayıt işlemi		
		4	Dosya seç butonuna tıklanmalı.	başarılı" mesajını		hesabı olmalı	başarılı olmasına		
		5	Verileri incele butonuna tıklanmalı.	Öğrenci Kaydı Lis	eklediği kaydı gön		başarılı olmasına		
		6	"Kaydet" butonuna tıklanmalı.				başarılı olmasına		
							başarılı olmasına		
YKB	YENİ KAYIT BİLGİLERİ	1	Sol menüde bulunan "Kayıt İşlemleri" butonuna t						
8 den sonra		2	"Yeni Kayıt" butonuna tıklanmalı.	"Aktif Yıl Yeni Kay					
		3	"Yeni Kayıt Bilgileri" butonuna tıklanmalı.	İstatistikleri" kayıt	İndirmek istenen dosya	Kullanıcının ka	Dosyalar		
		4	"PDF" butonuna tıklanmalı.	İşlemlerinden soni	türü seçilmeli,	hesabı olmalı	gösteriliyor,		
				haliye görüntülen					
YKY	YENİ KAYIT YENİLEME	1	Sol menüde bulunan "Kayıt İşlemleri" butonuna t						
		2	"Kayıt Yenileme" butonuna tıklanmalı.						
		3	"Yeni Kayıt Yenileme" butonuna tıklanmalı.						
		4	"VELİYE ULAŞ" alanları doldurulmalı.						
		5	"SMS Gönder" butonuna tıklanmalı.	SMS başarıyla gö					
		6	"GENEL BİLGİLER" alanı doldurulmalı.	Kullanıcı "Kayıt İş		Kullanıcının ka			
		7	Aranan öğrencinin geldiği teyit edilmeli.	başarılı" mesajını		hesabı olmalı			
		8	"Okul Numarası" alanı doldurulmalı.	İleştikten sonra yer					
				kaydının yenilendi					
				görebilmeli.					

Şekil 2.2. iKampüs web sitesi için yazılan test senaryoları

Tablo 2.1. İş İlerleme Durumu Tablosu

	İş Kodu	İş Adı	Yapan Kişi	Üzerinde Çalışılıyor	Merge	Tamamlandı
1	ÖKGE-1	ÖN KAYIT GÖRÜŞMESİ EKLE	Dilek Ayşe GÜN			X
2	ÖKGE-2	ÖN KAYIT GÖRÜŞMESİ EKLE	Dilek Ayşe GÜN			X
3	ÖKGL-1	ÖN KAYIT GÖRÜŞMESİ LİSTESİ	Tuğba ULUSOY			X
4	ÖKGD-1	ÖN KAYIT GÖRÜŞMESİ DÜZENLE	Tuğba ULUSOY			X
5	AÖKGD-1	ÖN KAYIT GÖRÜŞMESİ SONUCLANDIRMA	Tuğba ULUSOY			X
6	YÖK-1	YENİ ÖĞRENCİ KAYDI	Dilek Ayşe GÜN			X
7	TÖK-1	TOPLU ÖĞRENCİ KAYDI	Dilek Ayşe GÜN			X
8	YKY	YENİ KAYIT YENİLEME	Tuğba ULUSOY			X
9	YKB	YENİ KAYIT BİLGİLERİ	Tuğba ULUSOY			X
10	KYG	KAYIT YENİLEME GÖRÜŞMELERİ	Dilek Ayşe GÜN			X
11	AÖEK-1	DOSYA EKLEMELİ ÖDEV EKLEME	Dilek Ayşe GÜN			X
12	AÖEK-2	METİN ŞEKLİNDE ÖDEV EKLEME	Tuğba ULUSOY			X
13	AVÖ-1	ADMİN VERDİĞİM ÖDEVLER	Tuğba ULUSOY			X
14	AVÖ-2	ADMİN VERDİĞİM ÖDEVLER	Dilek Ayşe GÜN			X
15	AVÖ-3	ADMİN VERDİĞİM ÖDEVLER	Dilek, Tuğba			X
16	AÖ-1	ADMİN ÖDEVLERİM	Tuğba ULUSOY			X
17	AGÖ-1	ADMİN GEÇMİŞ ÖDEVLER	Dilek Ayşe GÜN			X
18	AEKE-1	ADMİN EĞİTİM KURUMLARI EKLE	Tuğba ULUSOY			X
19	AEKL-1	ADMİN EĞİTİM KURUMLARI LİSTESİ	Tuğba ULUSOY			X
20	AEKD-1	ADMİN EĞİTİM KURUMU DÜZENLE	Tuğba ULUSOY			X
21	AEKS-1	ADMİN EĞİTİM KURUMU SİL	Tuğba ULUSOY			X
22	AKE-1	ADMİN KAMPÜS EKLE	Dilek Ayşe GÜN			X
23	AKE-2	ADMİN TOPLU KAMPÜS EKLE	Dilek Ayşe GÜN			X
24	AKL-1	ADMİN KAMPÜS LİSTESİ	Dilek Ayşe GÜN			X
25	AKD-1	ADMİN KAMPÜS DÜZENLE	Dilek Ayşe GÜN			X
26	AKS-1	ADMİN KAMPÜS SİL	Dilek Ayşe GÜN			X
27	AGS-1	ADMİN GİRİŞ SAYFASI	Tuğba ULUSOY			X

28	AGS-2	ADMİN GİRİŞ SAYFASI RESİM SİL	Tuğba ULUSOY			X
29	AOE-1	ADMİN OKUL EKLE	Dilek Ayşe GÜN			X
30	AOE-2	ADMİN TOPLU OKUL EKLE	Dilek Ayşe GÜN			X
31	AOL-1	ADMİN OKUL LİSTESİ	Dilek Ayşe GÜN			X
32	AOL-2	ADMİN OKUL DÜZENLE	Tuğba ULUSOY			X
33	AOS-1	ADMİN OKUL SİL	Tuğba ULUSOY			X
34	YTE-1	YER TANIMI EKLE	Tuğba ULUSOY			X

	A	B	C
51	SBD-1	SÖZLEŞME BEDELİ DÜZENLE	
52	SBS-1	SÖZLEŞME BEDELİ SİL	
53	İBE-1	İNDİRİM BURS EKLE	
54	İBL-1	İNDİRİM BURS LİSTESİ	
55	İBS-1	İNDİRİM BURS SİL	
56	İBD-1	İNDİRİM BURS DÜZENLE	
57	SE-1	SINIF EKLE	
58	TSE-1	TOPLU SINIF EKLE	
59	SL-1	SINIF LİSTESİ	
60	SD-1	SINIF DÜZENLE	
61	SS-1	SINIF SİL	
62	ŞE-1	ŞUBE EKLE	
63	ŞL-1	ŞUBE LİSTESİ	
64	ŞD-1	ŞUBE DÜZENLE	
65	ŞS-1	ŞUBE SİL	
66	TT-1	TEŞVİK TANIMLAMALARI	
67	TTL-1	TEŞVİK TANIMLAMALARI LİSTESİ	
68	YLG-1	LİSTE YÜKLE	
69	YLG-2	YEMEK LİSTESİ	
70	NÖT-1	NÖBETÇİ ÖĞRETMEN TANIMLA	
71	NÖL-1	NÖBETÇİ ÖĞRETMEN LİSTESİ	
72	MGE-1	MESLEK GRUBU EKLE	
73	MGL-1	MESLEK GRUBU LİSTESİ	
74	ME-1	MESLEK EKLE	
75	MI-1	MESLEK LİSTESİ	

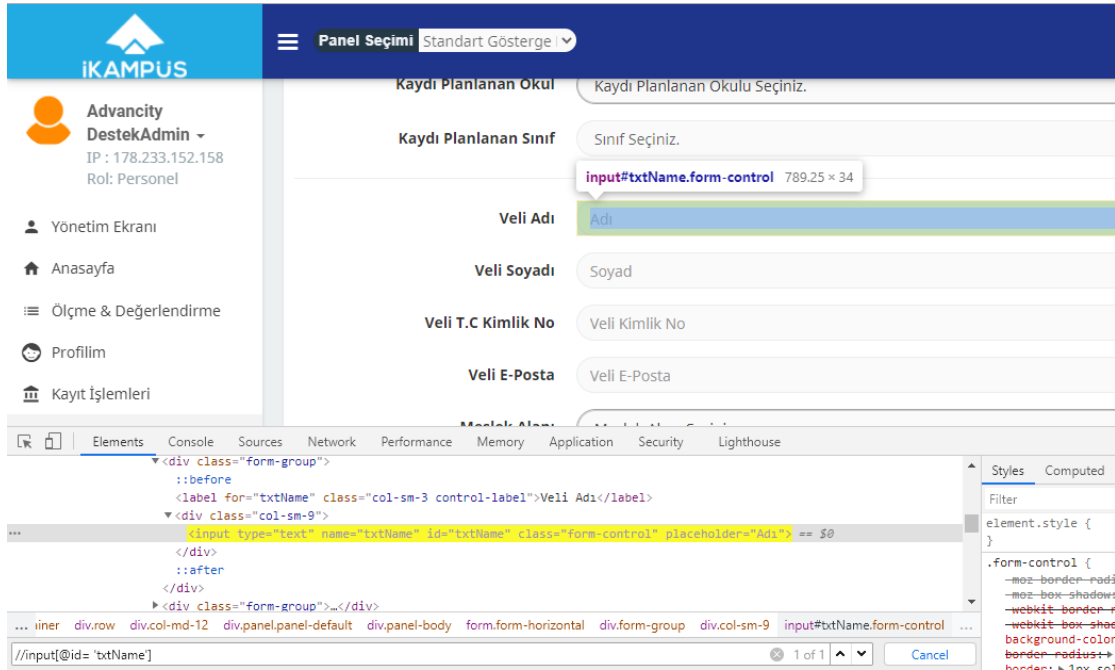
Şekil 2.3. iKampüs Test Senaryolarının kısaltma kodları

2.1. Öğeyi Denetle(Inspect) İle Yol Belirteçleri

Sağ tık öğeyi denetle (inspect) yi seçerek işlemi yapacağımız alanın yolu bulunmalı.

Örneğin: //input[@id= 'txtName']

Yolu yazarken id almaya dikkat etmeliyiz. Id olmadığı zamanlar class, name gibi tanımlamaları da kullanabiliriz.



Şekil 2.4. Web sitesinde öğeyi denetle yol belirteci oluşturma.

2.2 Branch İşlemi

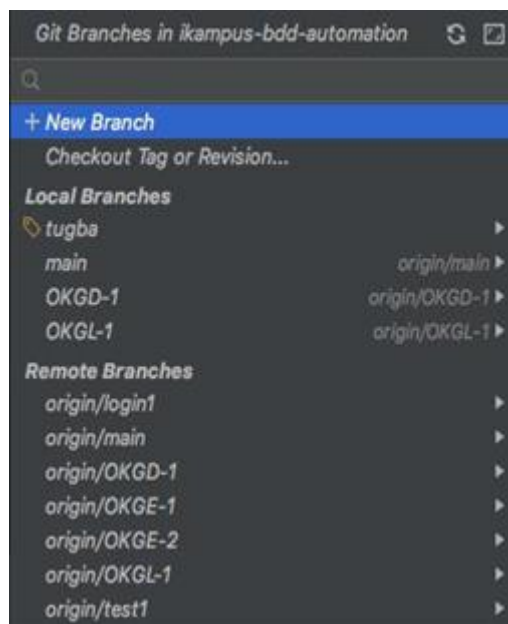
Projede çalışmaya başlarken ilk adım branch oluşturmaktır. Projeyi açınca en alt sağ köşedeki branch sembolüne tıkladık.



Şekil 2.5. Branch sembolü

New Branch denilerek yeni branch oluşturuldu. Biz başlangıçta senaryo isimlerimize göre her seferinde bir branch oluşturuyorduk. Ama daha sonra kolaylık olması için isimlerimiz ile birer branch oluşturduk. Senaryo isimlerimiz ile commit yapıyoruz böylece her seferinde branch işlemi gerçekleştirmemiş olduk.

Senaryolarımızın “Class ve Dosyaları” nı bu branch’ in içinde oluşturup kodluyoruz.



Şekil 2.6. Yeni branch oluşturma

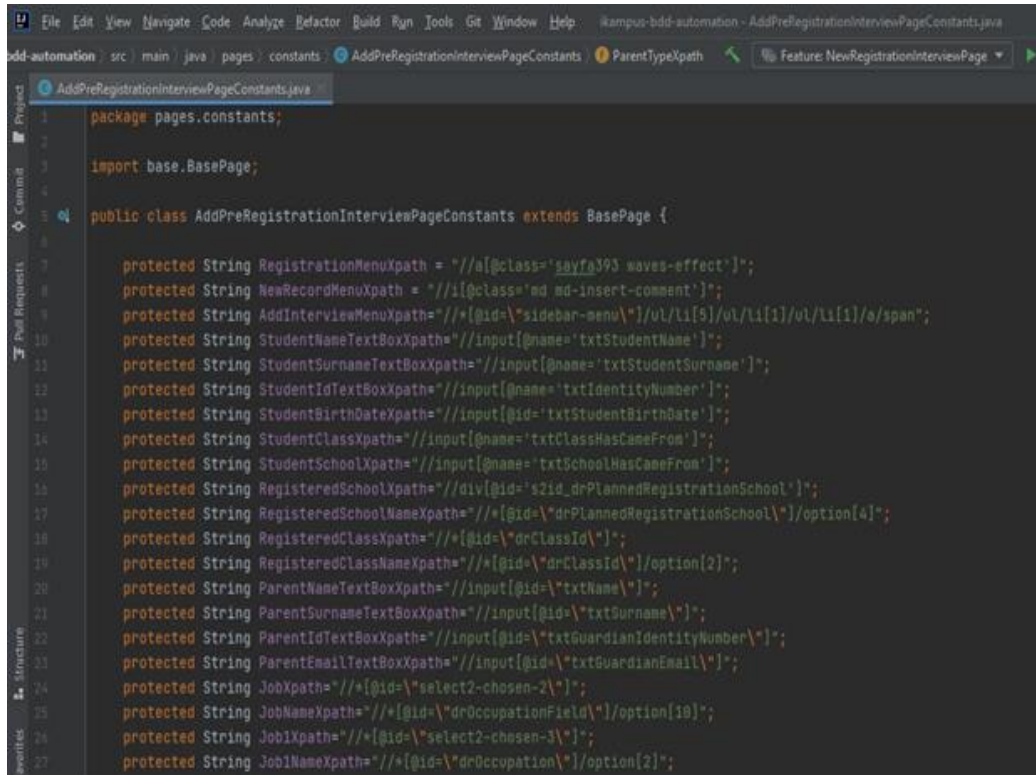
2.3 Projede Kullanılan Class Ve Dosyalar

Constants : Xpath lerin yazıldığı class.

Page : Senaryolara ait Java kodlarının bulunduğu class.

PageSteps : Senaryoların kaç adımdan oluştuğunu gösteren class.

.feature : Senaryoları yazdığımız ve çalıştırdığımız dosya [2].



```

package pages.constants;

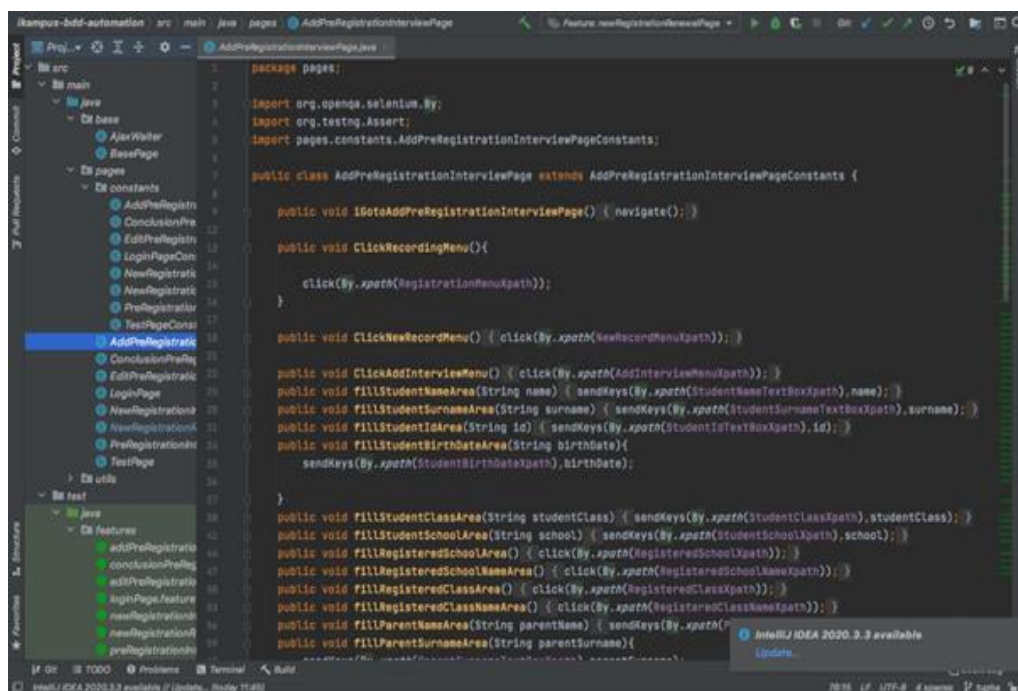
import base.BasePage;

public class AddPreRegistrationInterviewPageConstants extends BasePage {

    protected String RegistrationMenuXPath = "//a[@class='sayfa393 waves-effect']";
    protected String NewRecordMenuXPath = "//i[@class='md md-insert-comment']";
    protected String AddInterviewMenuXPath = "/*[@id='\"sidebar-menu\"']/ul/li[5]/ul/li[1]/ul/li[1]/a/span";
    protected String StudentNameTextBoxXPath = "//input[@name='txtStudentName']";
    protected String StudentSurnameTextBoxXPath = "//input[@name='txtStudentSurname']";
    protected String StudentIdTextBoxXPath = "//input[@name='txtIdentityNumber']";
    protected String StudentBirthDateXPath = "//input[@id='txtStudentBirthDate']";
    protected String StudentClassXPath = "//input[@name='txtClassHasCameFrom']";
    protected String StudentSchoolXPath = "//input[@name='txtSchoolHasCameFrom']";
    protected String RegisteredSchoolXPath = "//div[@id='s2id_drPlannedRegistrationSchool']";
    protected String RegisteredSchoolNameXPath = "/*[@id='\"drPlannedRegistrationSchool\"']/option[4]";
    protected String RegisteredClassXPath = "/*[@id='\"drClassId\"']";
    protected String RegisteredClassNameXPath = "/*[@id='\"drClassId\"']/option[2]";
    protected String ParentNameTextBoxXPath = "//input[@id='\"txtName\"']";
    protected String ParentSurnameTextBoxXPath = "//input[@id='\"txtSurname\"']";
    protected String ParentIdTextBoxXPath = "//input[@id='\"txtGuardianIdentityNumber\"']";
    protected String ParentEmailTextBoxXPath = "//input[@id='\"txtGuardianEmail\"']";
    protected String JobXPath = "/*[@id='\"select2-chosen-2\"']";
    protected String JobNameXPath = "/*[@id='\"drOccupationField\"']/option[10]";
    protected String Job1XPath = "/*[@id='\"select2-chosen-3\"']";
    protected String Job1NameXPath = "/*[@id='\"drOccupation\"']/option[2]";
}

```

Şekil 2.7. Constants class' ında değişkene XPath' leri tanımlandı.



```

package pages;

import org.openqa.selenium.By;
import org.testng.Assert;
import pages.constants.AddPreRegistrationInterviewPageConstants;

public class AddPreRegistrationInterviewPage extends AddPreRegistrationInterviewPageConstants {

    public void gotoAddPreRegistrationInterviewPage() { navigate(); }

    public void ClickRecordingMenu() {
        click(By.xpath(RegistrationMenuXPath));
    }

    public void ClickNewRecordMenu() { click(By.xpath(NewRecordMenuXPath)); }

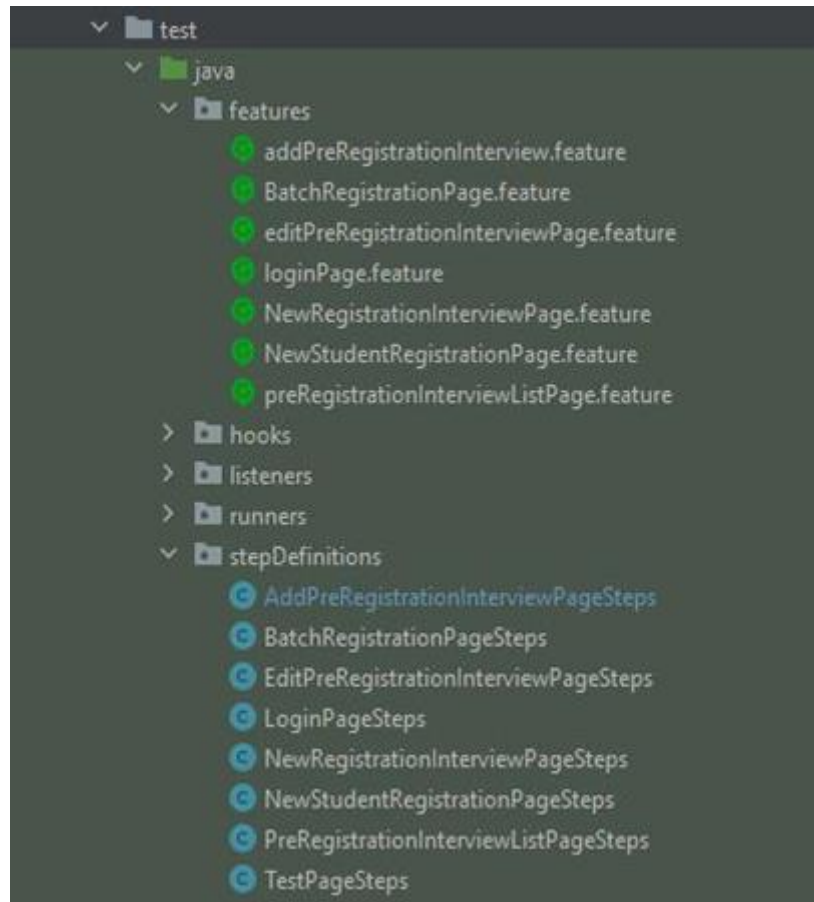
    public void ClickAddInterviewMenu() { click(By.xpath(AddInterviewMenuXPath)); }

    public void fillStudentNameArea(String name) { sendKeys(By.xpath(StudentNameTextBoxXPath), name); }
    public void fillStudentSurnameArea(String surname) { sendKeys(By.xpath(StudentSurnameTextBoxXPath), surname); }
    public void fillStudentIdArea(String id) { sendKeys(By.xpath(StudentIdTextBoxXPath), id); }
    public void fillStudentBirthDateArea(String birthDate) {
        sendKeys(By.xpath(StudentBirthDateXPath), birthDate);
    }

    public void fillStudentClassArea(String studentClass) { sendKeys(By.xpath(StudentClassXPath), studentClass); }
    public void fillStudentSchoolArea(String school) { sendKeys(By.xpath(StudentSchoolXPath), school); }
    public void fillRegisteredSchoolArea() { click(By.xpath(RegisteredSchoolXPath)); }
    public void fillRegisteredSchoolNameArea() { click(By.xpath(RegisteredSchoolNameXPath)); }
    public void fillRegisteredClassArea() { click(By.xpath(RegisteredClassXPath)); }
    public void fillRegisteredClassNameArea() { click(By.xpath(RegisteredClassNameXPath)); }
    public void fillParentNameArea(String parentName) { sendKeys(By.xpath(ParentNameTextBoxXPath), parentName); }
    public void fillParentSurnameArea(String parentSurname) {
        sendKeys(By.xpath(ParentSurnameTextBoxXPath), parentSurname);
    }
}

```

Şekil 2.8. Java kodları Page class' ı içinde yazıldı.



Şekil 2.9. StepDefinitions class' ı ve features dosyası oluşturuldu.

```

11 public class AddPreRegistrationInterviewPageSteps {
12
13     AddPreRegistrationInterviewPage addPreRegistrationInterviewPage = new AddPreRegistrationInterviewPage()
14
15     @And("^I go to add pre-registration interview page$")
16     public void iGotoAddPreRegistrationInterviewPage(){ addPreRegistrationInterviewPage.iGotoAddPreRegistrationInterviewPage()
17     }
18
19     @And("^add pre-registration I click Recording Menu$")
20     public void ClickRecordingMenu(){ addPreRegistrationInterviewPage.ClickRecordingMenu()
21     }
22
23     @And("^add pre-registration I click new record menu$")
24     public void ClickNewRecordMenu(){ addPreRegistrationInterviewPage.ClickNewRecordMenu()
25     }
26
27     @And("^add pre-registration I click add interview menu$")
28     public void ClickAddInterviewMenu(){ addPreRegistrationInterviewPage.ClickAddInterviewMenu()
29     }
30
31     @And("^add pre-registration Fill name with (.*)$")
32     public void fillStudentNameArea(String name){ addPreRegistrationInterviewPage.fillStudentNameArea(name)
33     }
34
35     @And("^add pre-registration Fill phone number with (.*)$")
36     public void fillPhoneNumberArea(String phone){ addPreRegistrationInterviewPage.fillPhoneNumberArea(phone)
37     }
38 }

```


Şekil 2.10. Steps' te selenium methodları oluşturuldu.

```

PreRegistrationInterview.feature
1 Feature: pre-registration interview feature
2 Background:
3   Given I go to login page
4   When Fill user email with [redacted]
5   Then Fill password with [redacted]
6   And I click login button
7   And I click Kayıt İşlemleri button sidebar menu
8   And I click Yeni Kayıt button first sub menu
9
10 Scenario: Pre-registration interview list must be displayed with successfully
11   And I click Ön Kayıt Görüşmesi Ekle button second sub menu
12   And Fill name with Melisa in add pre registration page
13   Then Fill surname with Yildiz in add pre registration page
14   And Fill parent name with Sinem in add pre registration page
15   Then Fill parent surname with Yildiz in add pre registration page
16   And Fill parent email with sinemyildiz@hotmail.com in add pre registration page
17   Then Fill parent cell phone with 5355555554 in add pre registration page
18   And I click save button in add pre registration page
19   And I click Ön Kayıt Görüşmesi Listesi button second sub menu
20   And I click excel button
21   And I click csv button
22   And I click pdf button
23

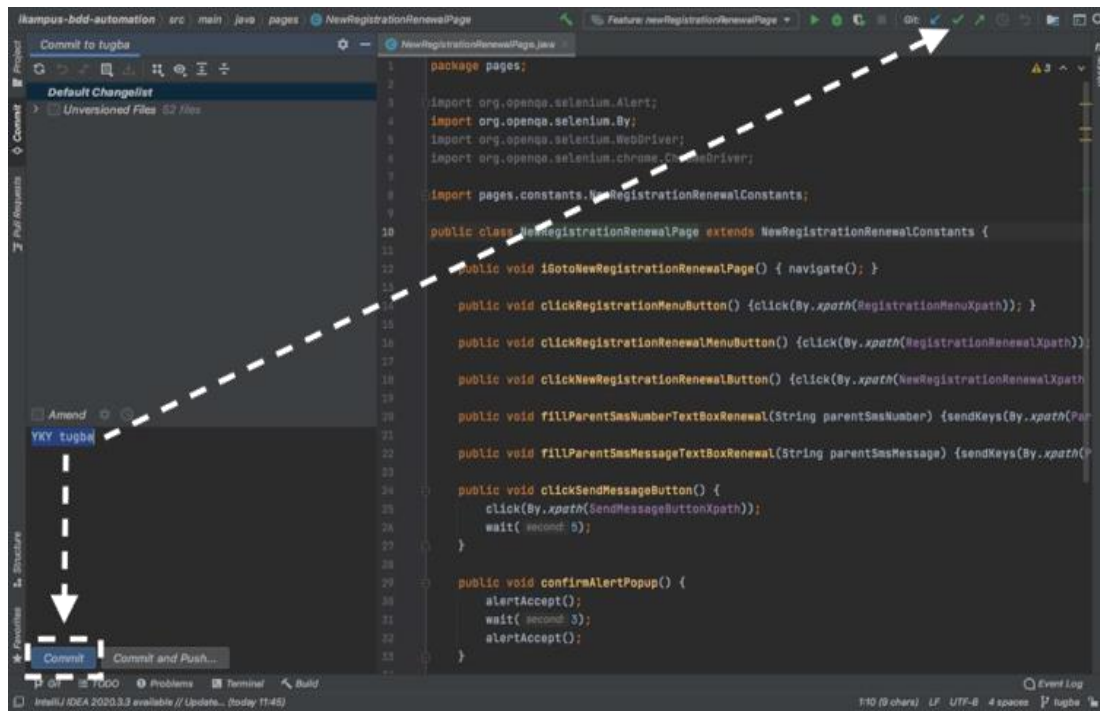
```

Şekil 2.11. Feature dosyasında test senaryoları yazıldı.

2.4. Commit İşlemi

Yapılan değişiklikleri projede kaydetmek için commit işlemi yapıldı. Senaryoyu commit yaptıktan sonra senaryoya ait class ve feature dosyaları adlarının rengi beyaza dönüyor. Eğer henüz commit yapılmamış bir senaryo ise class ve feature dosyalarının adlarının rengi yeşil görünüyor.

Commit yaparken önce yeşil tike tıklandı. Sonra senaryo adı ve branch adı yazıldı. Son olarak Commit butonuna tıklandı.



Şekil 2.12. Commit işlemi

BÖLÜM 3. JUNIT ASSERTIONS

JUnit Assertions Kullanımı yazılan test senaryosunun sonunda elde edilen sonuç ile beklenen sonuçların karşılaştırılması yapılması için kullanılır. Bir test senaryosundan beklenen sonuç ile elde edilmiş olan sonuç aynı ise test başarılı olarak nitelendirilir. Assertions kullanımında birebir aynı sonuçların gelmesini beklememiz gerekmektedir [3].

3.1. JUnit Assertions Tipleri

Sekiz adet assertions tipi vardır. Bunlar;

1. assertEquals
2. assertTrue
3. assertFalse
4. assertNull
5. assertNotNull
6. assertSame
7. assertNotSame
8. assertArrayEquals

Projede assertEquals kullanıldı;

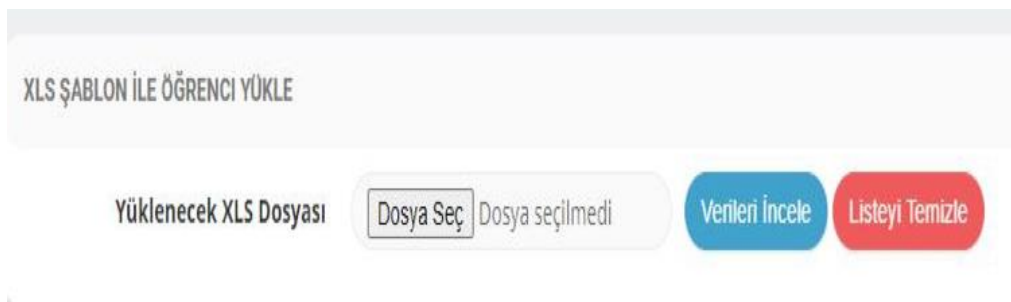
AssertEquals, beklenen sonucu gerçek sonuç ile karşılaştırmak için kullanılır. Beklenen sonuç ile gerçek sonuç eşit değil ise gerçekleştirilen test senaryosu sonucunda assertionError hatası fırlatır. [4]

```
159 public void AddPreRegistrationInterviewPageControl(){
160     System.out.println(getElementText(By.xpath(popoverMessageTitleXPath)));
161     System.out.println(getElementText(By.xpath(popoverMessageContentXPath)));
162     Assert.assertEquals(getElementText(By.xpath(popoverMessageTitleXPath)),
163         s1: "Bilgi", s2: "Ön kayıt formu eklenemedi!");
164     Assert.assertEquals(getElementText(By.xpath(popoverMessageContentXPath)),
165         s1: "İşlem başarılı!", s2: "Ön kayıt formu eklenemedi!");
166 }
167
```

Şekil 3.1. Assert kullanarak gelen mesajın kontrolü yapıldı.

BÖLÜM 4. FIND() METHODU

Dosya seçimi için find methodu kullanıldı. Dosya seç butonunun yolu ve yüklenecek dosyanın bulunduğu yerin adresi string değişkene atandı. SendKeys ile verilen dosyanın yolu belirtildi ve find ile bulduruldu [5].



Şekil 4.1. Dosya seçme işlemi.

```
protected String RegistrationMenuXPath = "//a[@class='sayfa393 waves-effect']";
protected String NewRecordMenuXPath = "//i[@class='md md-insert-comment']";
public String BatchRegistrationMenuXPath="//*[@id='sidebar-menu']/ul/li[5]/ul/li[1]/ul/li[4]/a/i";
public String filePath="C:\\Users\\Dilek\\ogrenciYukle.xls";
public String UploadFileButtonId="txtFile1";
public String dataReviewButtonXPath="//button[@id='btnSubmit']";
```

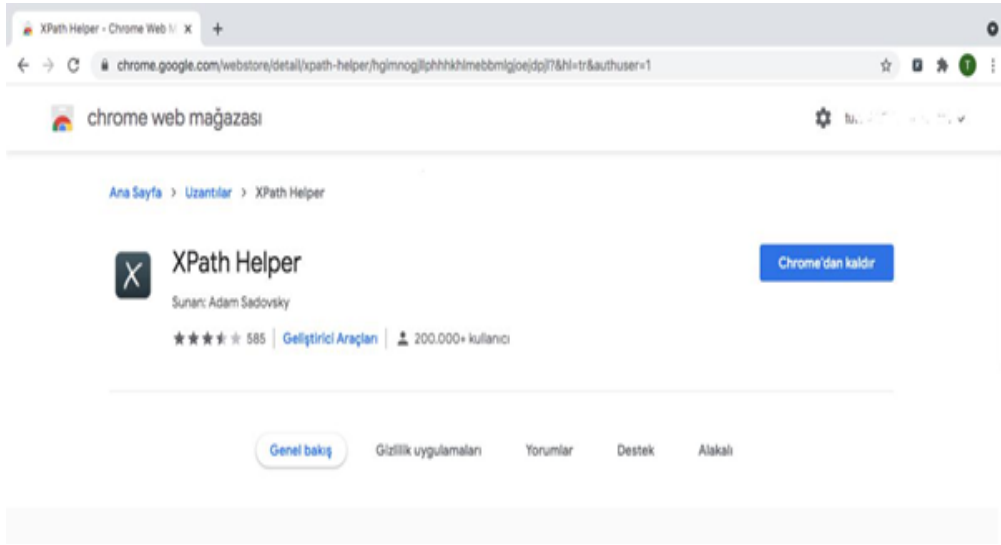
Şekil 4.2. Constants'ta buton ve dosyanın yolunu belirtildi.

```
public void Findfile(){
    find(By.id(UploadFileButtonId)).sendKeys(filePath);
}
```

Şekil 4.3. Find() ile dosya seçme işlemi yapıldı.

BÖLÜM 5. XPATH HELPER CHROME

XPath Helper Chrome ‘u tarayıcıya eklendi. Bu bizim bulduğumuz XPath’ lerin (işlemi yapacağımız alanın yolunu) doğruluğundan emin olmada yardımcı oluyor [6].

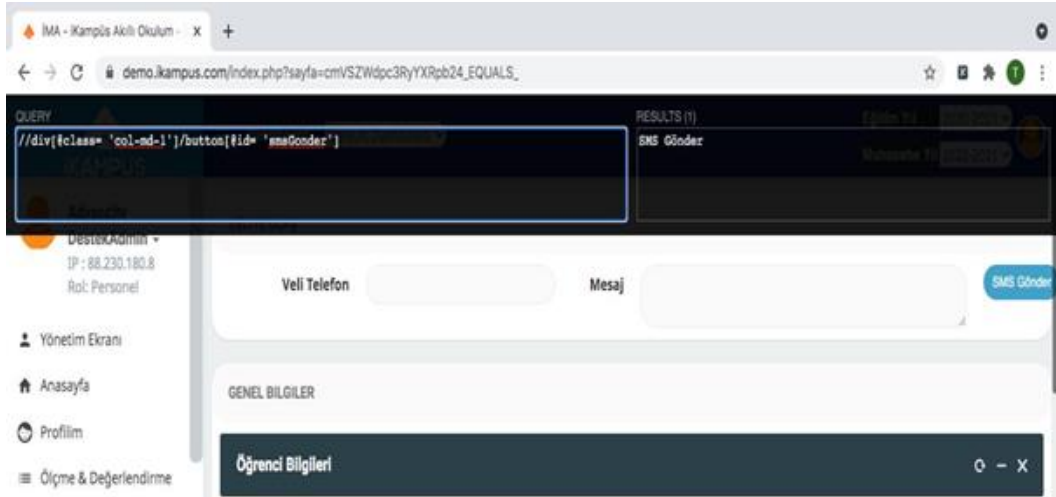


Şekil 5.1. XPath Helper Chrome’ u tarayıcıya ekleme işlemi.

Örneğin: `//div[@class= 'col-md-1']/button[@id= 'smsGonder']`

“SMS Gönder” butonunun yolunu yani XPath’ini bulmak için daha önce bahsedildiği gibi butonda sağ tık -> inspect yaparak html kodlarına bakarak yolu bulundu. Bu yolun doğruluğu XPath Helper yardımıyla test edildi.

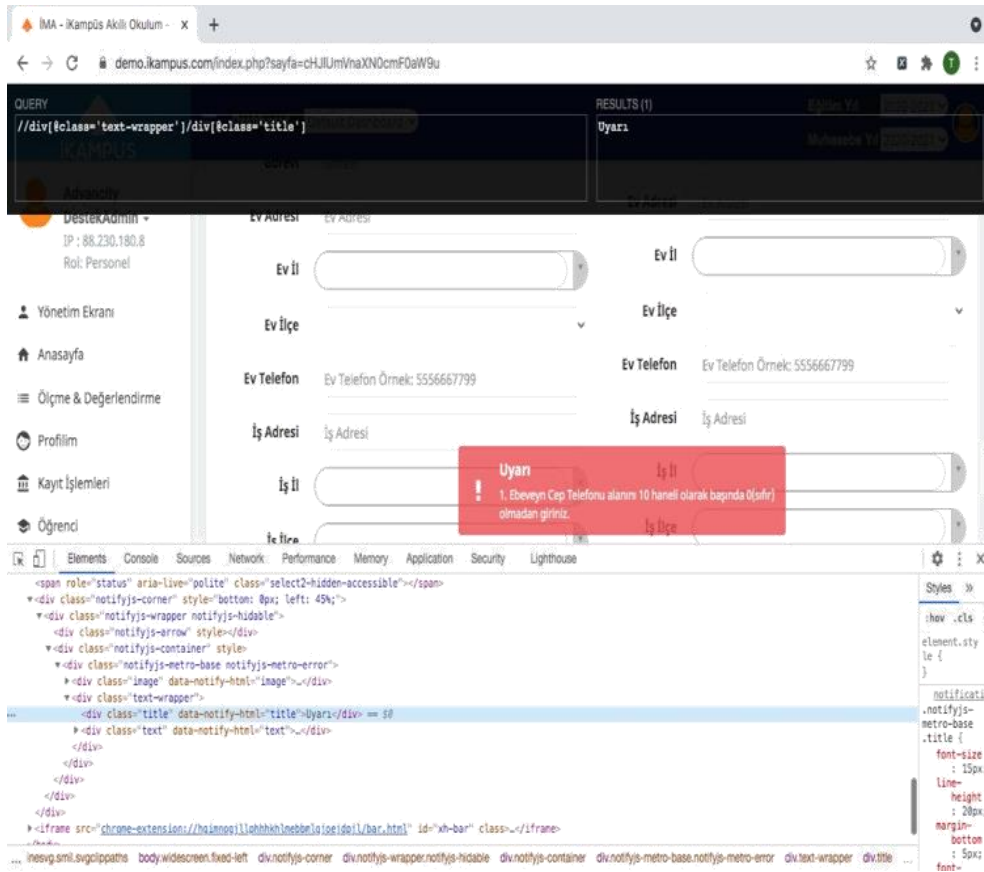
XPath Helper, Pop-up larda xpath bulurken de oldukça yardımcı olan bir eklenti. Özellikle bu eklentiye tarayıcıya eklenme sebebi; pop-up’ ların XPath’ini bulabilmek.



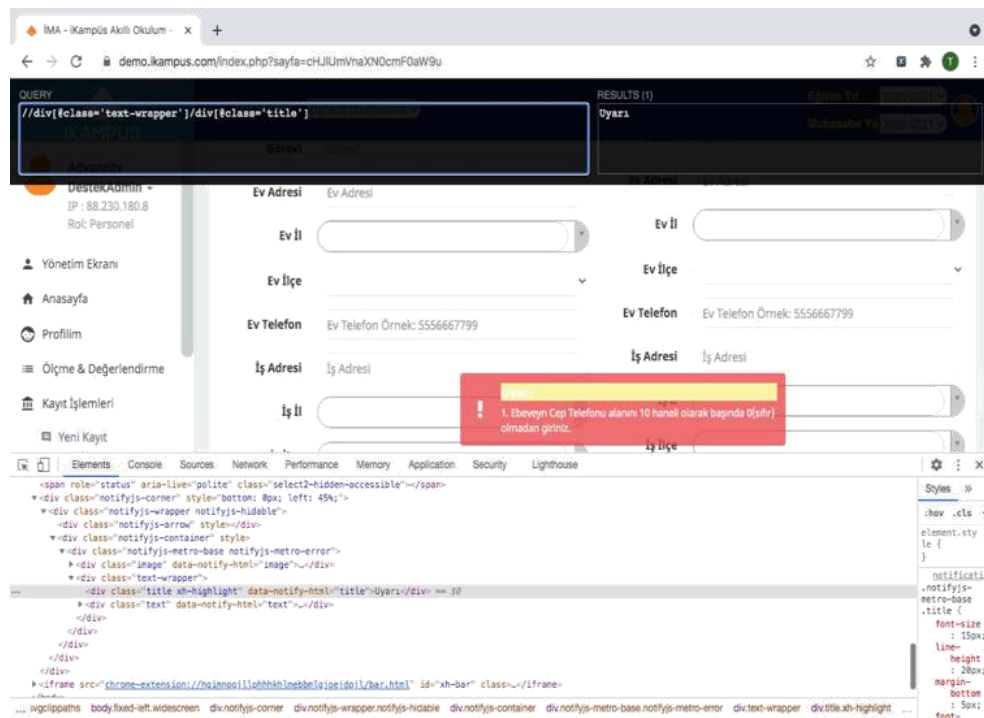
Şekil 5.2. XPath Helper Chrome ile “SMS Gönder” butonunun XPath’ inin doğruluğunu test etme işlemi.

5.1. Pop-up XPath’ini Bulma Adımları

- 1) Pop-up üzerinde sağ tık incele.
- 2) XPath Helper eklentisini aç.
- 3) Pop-up taki “Uyarı” yazısının XPath’ ini html kodlarına bakarak yaz.
- 4) Tekrar pop-up’ ta sağ tık incele de.
- 5) XPath Helper’ da XPath’ i yazdığın yere tıkla.
- 6) Doğru XPath yazıldıysa sarı renkte görünmekte.



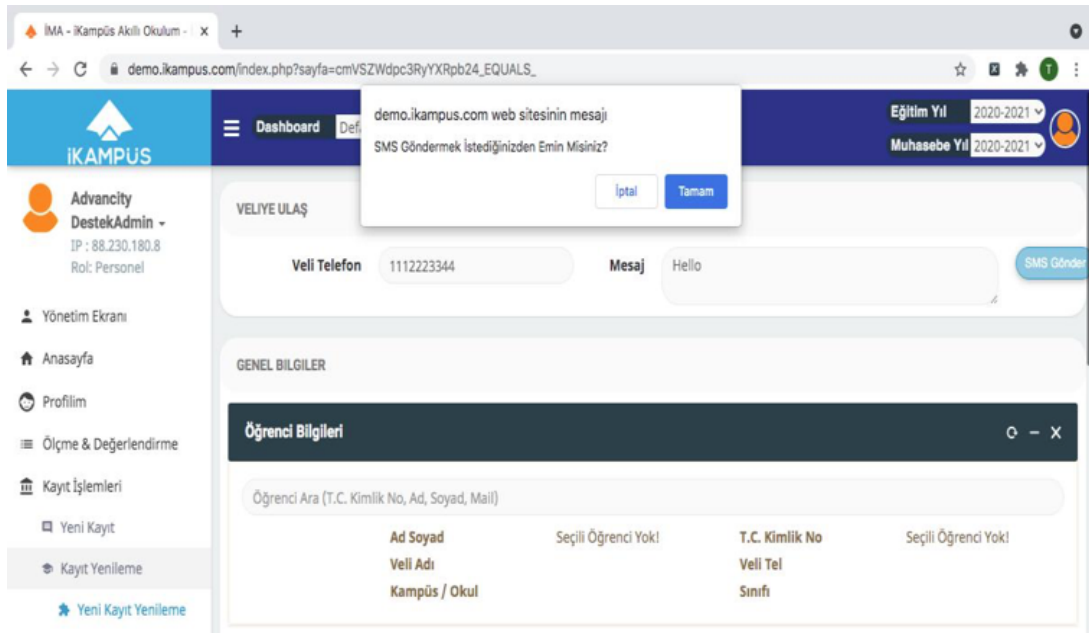
Şekil 5.3. Pop-up XPath' ini bulma.



Şekil 5.4. XPath Helper'da XPath'in yazıldığı yere tıklayınca "Uyarı" kelimesinin bulunduğu textbox sarı renge döndüyor.

BÖLÜM 6. ALERT POP-UP ONAYLAMA

AlertAccept() fonksiyonu Şekil 6.1' de görüntülenen uyarı iletişim kutusunu kabul eder. Genellikle bu, iletişim kutusundaki 'Tamam' düğmesine tıklamaya eşdeğerdir.



Şekil 6.1. Alert Pop-up

```
public void fillParentSmsNumberTextBoxRenewal(String parentSmsNumber) {sendKeys(By.xpath(ParentSmsNumberTextBoxRenewalXpath));}

public void fillParentSmsMessageTextBoxRenewal(String parentSmsMessage) {sendKeys(By.xpath(ParentSmsMessageTextBoxRenewalXpath));}

public void clickSendMessageButton() {
    click(By.xpath(SendMessageButtonXpath));
    wait(second: 5);
}

public void confirmAlertPopup() {
    alertAccept();
    wait(second: 3);
    alertAccept();
}
```

Şekil 6.2. Alert Pop-up onaylama kodu

BÖLÜM 7. PROJE SENARYOLARINI DÜZENLEME

Değişkenlerin yollarında ortak noktalar olabiliyor. Bu durumlarda fonksiyon sayısını azaltıp daha düzenli hale getirebiliyoruz. Aşağıdaki örnekte sadece butonun ismi değişiyordu. Görseldeki gibi tek bir fonksiyonda dört farklı olayı gerçekleştirdik. Senaryoda Copy, Excel, Csv, Pdf butonlarına basılması için, aşağıdaki gibi dört satır yazdık.

```

22 >> Scenario: New registration interviews
23     And I click Kayıt Yenileme Görüşmeleri button second sub menu
24     And I click accounting year
25     And I click accounting year area
26     And I click copy button in the new registration interview page
27     And I click excel button in the new registration interview page
28     And I click csv button in the new registration interview page
29     And I click pdf button in the new registration interview page

```

Şekil 7.1. Yeni kayıt görüşmeleri senaryosu, tek fonksiyonda farklı olayların gerçekleşmesi

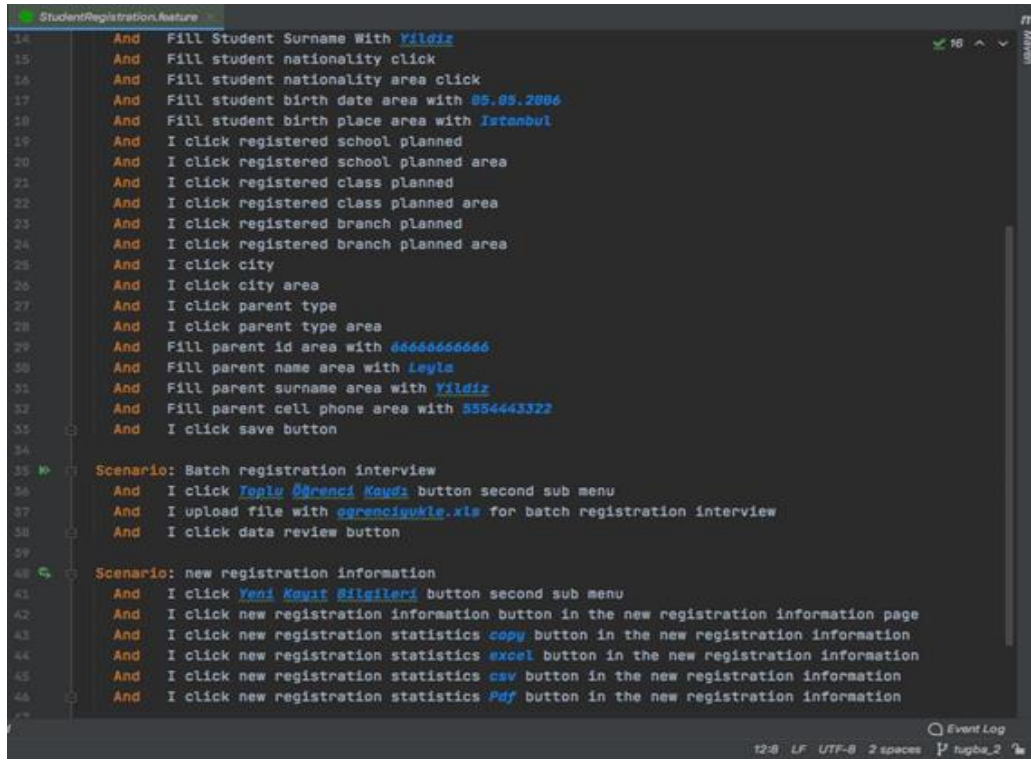
7.1. Tek Feature Dosyasında Birden Fazla Senaryo Yazma

```

StudentRegistration.feature
1 Feature: Student Registration
2   Background:
3     Given I go to login page
4     When Fill user email with 
5     Then Fill password with 
6     And I click login button
7     And I click Kayıt İşlemleri button sidebar menu
8     And I click Yeni Kayıt button first sub menu
9
10  Scenario: New student registration
11    And I click Yeni Öğrenci Kaydı button second sub menu
12    And Fill student id with 7777777777
13    And Fill Student Name With Yusuf
14    And Fill Student Surname With Yıldız
15    And Fill student nationality click
16    And Fill student nationality area click
17    And Fill student birth date area with 09.05.2006
18    And Fill student birth place area with İstanbul
19    And I click registered school planned
20    And I click registered school planned area
21    And I click registered class planned
22    And I click registered class planned area
23    And I click registered branch planned
24    And I click registered branch planned area
25    And I click city
26    And I click city area
27    And I click parent type
28    And I click parent type area
29    And Fill parent id area with 666666666666
30    And Fill parent name area with Leylek
31    And Fill parent surname area with Yıldız
32    And Fill parent cell phone area with 9994443322
33    And I click save button

```

Şekil 7.2. Tek feature dosyasında birden fazla senaryo yazma



Şekil 7.3. Tek feature dosyasında birden fazla senaryo yazma

Ekran görüntülerinde görüldüğü gibi “ Background: ” altında feature dosyası içindeki yazılacak senaryolarda, ortak olarak gerçekleştirilen adımlar yazılır. Sonrasında senaryolarda, background altındaki adımlardan sonra devam edecekleri adımlardan başlayarak kalan adımlar sırayla yazılır [4].

7.2. Feature Dosyasında Background Kullanmanın Faydası

Her senaryo için ayrı bir feature dosyası oluşturmak yerine tek feature dosyasında birden fazla senaryo çalıştırmayı sağlıyor. Böylelikle ortak kullanılan sidebar menü ve first sub menüsü bulunan senaryolar için fazladan feature dosyası oluşturmaya gerek kalmıyor. Bu da projedeki kafa karışıklıklarını önlemekte ve daha düzenli bir hâlde olmasını sağlamakta.

BÖLÜM 8. PROJEDE KULLANILAN BAZI FONKSİYONLAR

8.1. toLowerCase() Methodu

Büyük küçük harf duyarlılığı için kullanıldı [7].

```
public void iClickGivenParameterButtonInTheNewRegistrationInterviewPage(String parameter){
    click(By.xpath("//div[@class='table-responsive']/div/div/a[contains" +
        "(@class,'buttons-'+parameter.toLowerCase()+')']/span"));
}
```

Şekil 8.1. toLowerCase() methodunun kullanımı gösterilmektedir.

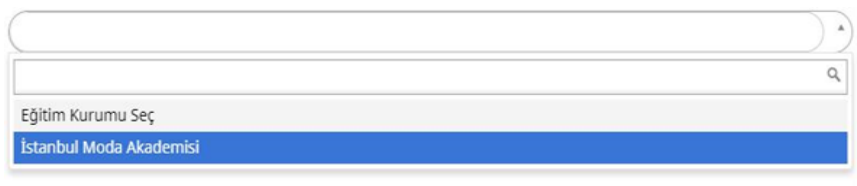
8.2. SelectByIndex() Fonksiyonu

İndeks belirterek seçim yapmak istenildiği durumlarda bu fonksiyonu kullanılır. Örnekte de görüldüğü üzere ilk değeri seçme işlemi yapıldı [8].

```
22 public void EducationalInstitutionArea(){
23     clickJS(By.xpath(EducationalInstitutionXPath));
24     selectByIndex(By.xpath(EducationalInstitutionAreaXPath), index: 1);
25 }
```

Şekil 8.2. selectByIndex() fonksiyonunun kullanımı gösterilmektedir.

Bu fonksiyon ile, ilk indeks olan “İstanbul Moda Akademisi” seçimi yapılacaktır.



Şekil 8.3. selectByIndex() fonksiyonu ile index seçimi

8.3. scrollToElement() Fonksiyonu

Selenium sayfanın altlarında kalan menülere, butonlara vs. ulaşamayabiliyor. Böyle durumlarda scrollToElement() fonksiyonu kullanılır. Bu fonksiyon ile sayfada kaydırma yaparak altta kalan araçlar seçilebilir [9].

```
11  
12 public void GeneralDefinitionsMenu(){  
13     scrollToElement(By.xpath(GeneralDefinitionsMenuXPath));  
14     clickJS(By.xpath(GeneralDefinitionsMenuXPath));  
15 }
```

Şekil 8.4. scrollToElement() fonksiyonunun kullanımı gösterilmektedir.

BÖLÜM 9. CLICK VS CLICKJS

Click tıklaması ve ClickJS tıklaması sürücü tarafından yürütülen tıklama, JavaScript HTMLElement sırasında gerçek bir kullanıcının davranışını mümkün olduğunca yakın bir şekilde simüle etmeye çalışır. click (), öge etkileşimli olmasa bile click olayı için varsayılan eylemi gerçekleştirir.

İki farklı şekilde click (butona tıklama işlemi) yapılabilir. Bazı fonksiyonlarda Click kullanıldığı zaman, tıklama işlemi tarayıcıda gerçekleşmeyebiliyor. Böyle durumlarda JavaScript ile tıklama yapabiliriz. Kullanımı “clickJS” dir.

Javascript kullanarak tıklama işlemi gerçekleştirmek için iki yol vardır:

1. FindElement(Javascript) + Click (Javascript)
2. FindElement(WebDriver) + Click (Javascript)

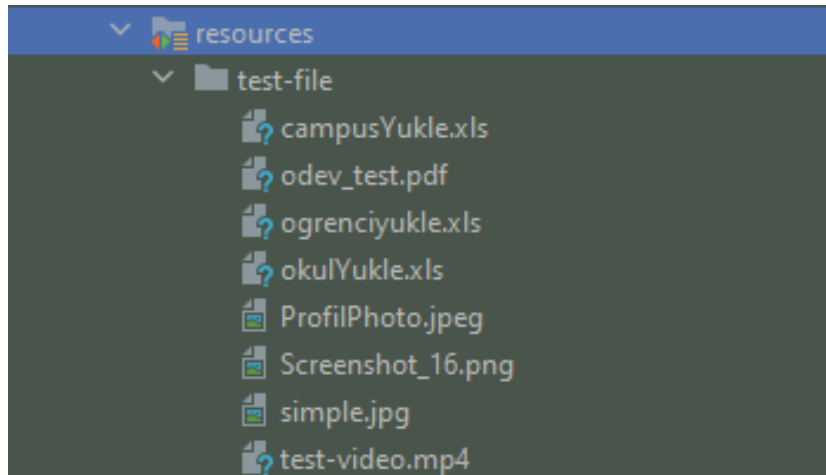
Her iki şekilde de javascript kullanarak tıklama işlemi yapmak yaygındır ancak eleman bulmak farklıdır.

İlk durumda, hem bir öge bulmak hem de tıklama işlemi gerçekleştirmek yalnızca Javascript kullanır.

İkinci durumda, elemanın bulunması web sürücüsünde yapılacak ve tıklama işlemi Javascript'te gerçekleştirilecektir.

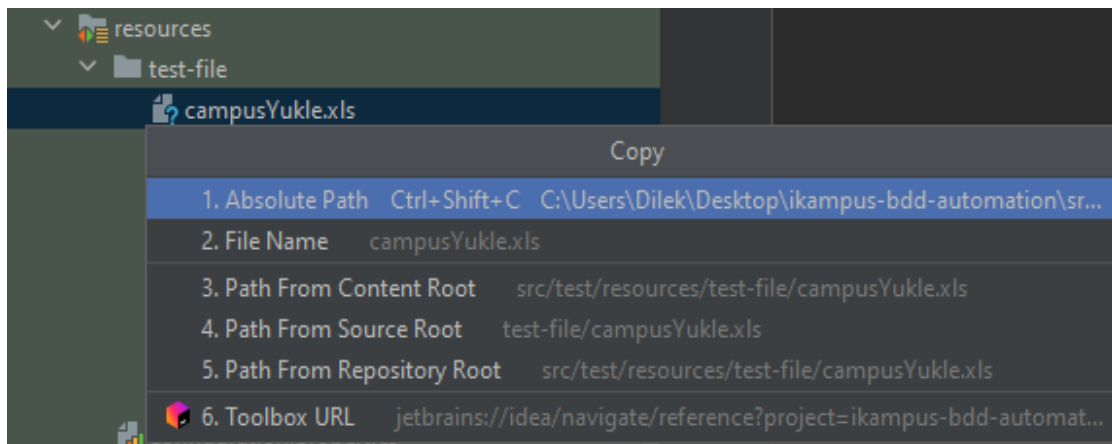
BÖLÜM 10. PROJEYE DOSYA EKLEME

Senaryoda dosya seçimi yapılacak alan olduğu zaman, test-file' a dosyalar eklenmektedir. Böylelikle projeyi başka bilgisayardan kullananlar da aynı dosyaya erişim sağlayabilir.



Şekil 10.1. Test-file klasöründe eklenen dosyalar görünmektedir

Yüklenen dosyaya sağ tık -> Copy Path dediğimizde, dosyanın projedeki yolunu alarak kodda kullanabiliriz.



Şekil 10.2. Eklenen dosyada sağ tık ile Absolute Path seçeneği seçilmesi

10.1. File.seperator

AddNewHomeworkWithFile fonksiyonunda dosya seçme işlemini yaptık.

Buradaki File.separator' un görevi, yolun sonundaki slash'ı eklemek. Tüm işletim sistemlerinde çalışmasını sağlıyor. Dosya yolunu bulmada hata olmasını önüyor. İşletim sistemine göre slash' ın görünümünü otomatik yapıyor. Böylelikle farklı bilgisayarlardan kullanımda aldığımız yol sorun çıkmıyor.

Örnek:

Mac OS işletim sisteminde -> /Users/tugba/Desktop/tugba.pdf

Windows işletim sisteminde -> C: \\Dilek\\Desktop\\dilek.pdf

File.separator “/” ın yönünü işletim sistemine göre belirliyor.

```
public void AddNewHomeworkWithFile(String file){  
    String filePath = new File( pathname: "src/test/resources/test-file").getAbsolutePath()  
        + File.separator + file.toLowerCase();  
    find(By.id(UploadFileButtonId)).sendKeys(filePath);  
    wait( second: 20);  
}
```

Şekil 10.3. File.separator methodunun kullanımı gösterilmektedir.

10.2. getAbsolutePath() Methodu

Verdiğimiz dosya yolunun kesinlikle bu olduğunu belirtiyoruz. Böylelikle dosyayı bulurken karıştırmıyor.

BÖLÜM 11. CLASS VE TEXT İLE SEÇİCİ BULMA

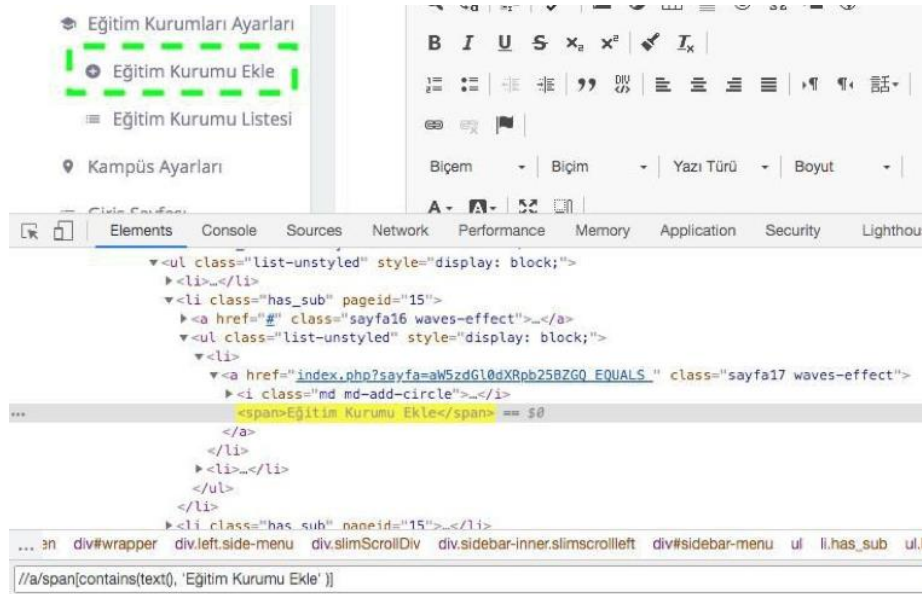
11.1. Contains İle Class Bulma

İd ya da class ile seçici yolunu bulamadığımızda “Contains” kullanarak seçicinin Xpath’ ini bulabiliriz.

Örnek: `//a[contains (@class, ‘waves-effect’)]`

11.2. Text İle Seçici Xpath’ ini Bulma

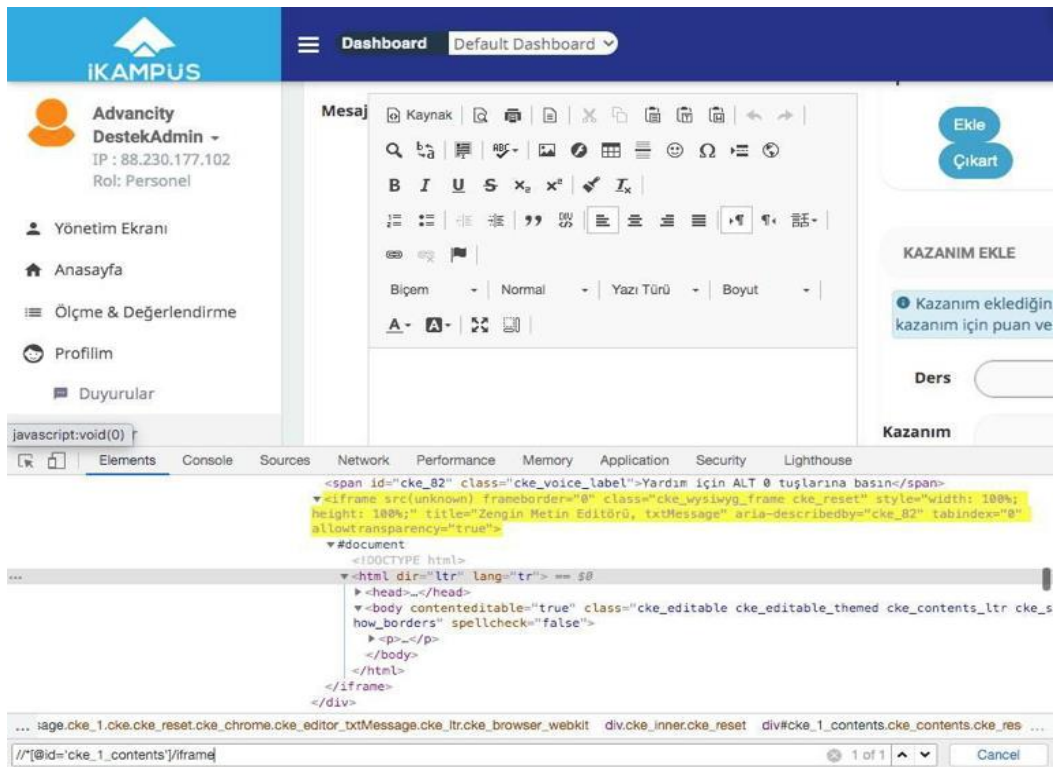
Örnek: `//a/span[contains(text(), ‘Eğitim Kurumu Ekle’)]`



Şekil 11.1. “Eğitim Kurumu Ekle” butonunun seçicisi text() ile bulundu.

BÖLÜM 12. SELENIUM'DA İFRAME KULLANIMI

İframe, bir sayfa içine dökümanlar, videolar ve interaktif medya yerleştirmeyi sağlayan bir HTML elementidir. İframe' i kullanarak ana sayfada ikinci bir web sayfası gösterilebiliyor. Eğer ulaşmak istediğimiz alanlar iframe tag' leri içerisinde bulunuyorsa buna ulaşmak için iframe' de odaklama işlemi yapılması gerekmektedir.



Şekil 12.1. Selenium' da iframe kullanımı

```
protected String HomeworkMessageXpath="//*[@id='cke_1_contents']/iframe";
```

Şekil 12.2. İframe Xpath kodu

```
public void fillHomeworkMessage(String iframeClass){
    driver.switchTo().frame(find(By.xpath(HomeworkMessageXPath)));
    sendKeys(By.xpath("html/body"), iframeClass);
    switchTab();
}
```

Şekil 12.3. İframe Page class kodu

```
@And("^Fill message with (.*) in the add new homework page$")
public void fillHomeworkMessage(String iframeClass) { addFileHomeworkPage.fillHomeworkMessage(iframeClass); }
```

Şekil 12.4. İframe PageSteps class kodu

```
And I upload homework with odev_test.pdf in the add new homework page
```

Şekil 12.5. İframe feature dosyası kodu

BÖLÜM 13. HATA OKUMA

Senaryolarda failed çıktısı alındığında, hata mesajını okumak hatanın ne olduğunu bulabilmek için çok önemli. Görselde görüldüğü üzere, hatanın 'AddPreRegistrationInterviewPage.java class' ında olduğu ve 138. Satırda olduğunu belirtmiş.

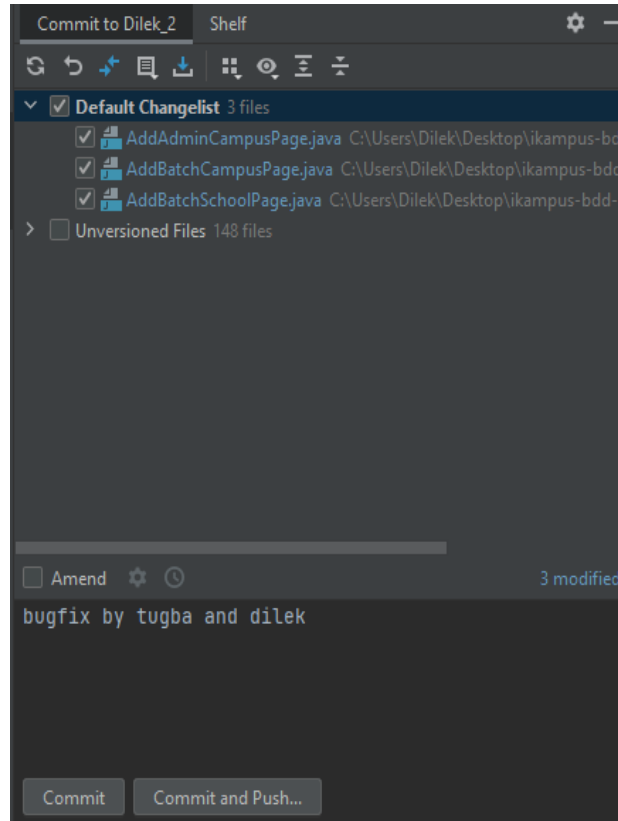


Şekil 13.1. Mavi altı çizili class ismine basıldığında hatanın olduğu yere hızlı şekilde gidilebilir.

BÖLÜM 14. GİTHUB'DA PROJEYİ MERGE EDEBİLMEK İÇİN GEREKLİ ALT ADIMLAR

14.1. Commit And Push

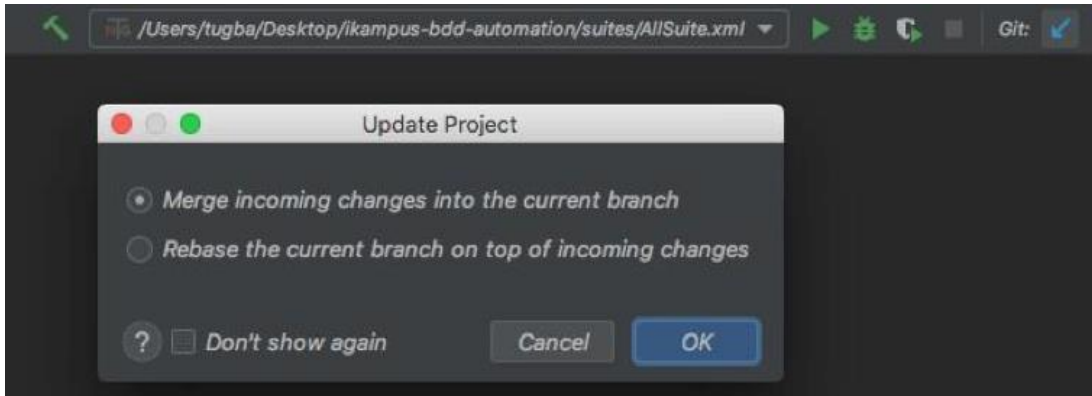
Senaryo bitirildiğinde herkes kendi branch' inde yazdığı senaryoları commit' liyor. Görselde gözüktüğü üzere, oluşturulan Dilek_ branch inde AddFileHomework senaryosunu tamamladıktan sonra commit and push işlemi gerçekleştiriliyor. Diğer dosyalarda da değişiklikler yapılmış fakat onlar genelde bir satır boş bırakma gibi fark edilmeden yapılmış oluyor. Bu yüzden onları seçmiyoruz. Senaryonun kodunu yazarak dört farklı dosyada yapılan değişikliği yolluyoruz.



Şekil 14.1. Commit ve Push işlemi

14.2. Pull

Proje ekip halinde yürütüldüğü için; “Commit and Push” işleminden sonra diğer takım arkadaşlarının yaptığı değişiklikler, senaryolar, classlar ve dosyalar kendi projemize gelebilmesi için Merge ve Pull işlemleri yapılıyor. Sağ üst köşedeki mavi ok “Pull” işlemi yapıyor. Mavi ok a tıkladıktan sonra çıkan pop-up’ ta “OK” butonuna tıklanıyor.



Şekil 14.2. Pull işlemi

Not: Merge işlemi şu anlık proje yöneticisi yapıyor. Daha sonra bize de nasıl yapıldığının gösterileceği söylendi.

BÖLÜM 15. LİSTE KONTROL İŞLEMİ

Kayıt işlemini yaptıktan sonra eklenen kaydın listede görüntülenmesi gerekiyor. Bu sebeple listeye kontrol eklenmeli.

Örnek olarak “Add Admin School” senaryosuna bakıldığında;

```
Scenario: Add Admin School
And I click Okul Listesi button second sub menu
And I get the old number in the admin add school page
And I click Okul Ekle button second sub menu
And I click campus area button in the admin add school page
And I click school type area button in the admin add school page
And Fill school name area with Test Otomasyon Okul Adı in the admin add school page
And Fill MEB code area with Test Otomasyon MEB Kodu in the admin add school page
And Fill authorized person area with Test Otomasyon Yetkili Kişi in the admin add school page
And Fill authorized person phone area with 5554443322 in the admin add school page
And Fill authorized person mail area with Test Otomasyon Yetkili Kişi E-Posta in the admin add school page
And I click save button in the admin add school page
And I control admin add school page
And I click Okul Listesi button second sub menu
And I get the new number then check old and new count in the admin add school page
```

Şekil 15.1. “Add Admin School” senaryosunda eklenen kontrol cümleleri gösterilmektedir.

İlk olarak “Okul Listesi” sayfasına giderek son kayıt numarası alınmalı. Kayıt numarası String değer geldiğinden veri tipini Integer yaparak değiştirilmeli.

```
public int GetCountOfSchoolList(){
    clickJS(By.xpath(RegistrationNumberTitleXpath));
    clickJS(By.xpath(RegistrationNumberTitleXpath));
    return Integer.parseInt(driver.findElement(By.xpath(RegistrationNumberXpath)).getText());
}
```

Şekil 15.2. Son kayıt numarasının alındığı ve Integer veri tipine dönüştürüldüğü fonksiyon gösterilmektedir.

Alınan kayıt numarasını oldNumber değişkenine atıldı.

```
@And("^I get the old number in the admin add school page$")
public void IGetTheOldNumberInTheAdminAddSchoolPage(){
    oldNumber=adminAddSchoolPage.GetCountOfSchoolList();
}
```

Şekil 15.3. Eski kayıt numarasının alındığı fonksiyon gösterilmektedir.

“Okul Ekle” senaryosuna geldiğinde ise gerekli alanlar doldurularak okul ekleme işlemi tamamlandı. Okul ekleme işlemi bitirdikten sonra tekrardan listeye gidilerek son kayıt numarasını alındı. Alınan kayıt numarası da newNumber değişkenine atıldı. Assertion ile, oldNumber ve newNumber arasında 1 kayıt fark mı var kontrol edildi.

```
@And("^I get the new number then check old and new count in the admin add school page$")
public void IGetTheNewNumberThenCheckOldAndNewCountInTheAdminAddSchoolPage(){
    newNumber=adminAddSchoolPage.GetCountOfSchoolList();
    Assert.assertEquals(++oldNumber,newNumber);
}
```

Şekil 15.4. Assertion ile, oldNumber ve newNumber sayıları arasındaki farkı gösterilen kontrol fonksiyonu gösterilmektedir.

BÖLÜM 16. AÇILAN PENCEREYİ KAPATMA (MOVEBYOFFSET)

Açılan pencerede seçim yapıldığında pencerenin kapanması gerekmektedir. Çünkü otomasyon bazı durumlarda, pencerenin altında kalan alanların yolunu bulamıyor.

16.1. Örnek

Doğum tarihi seçimi yapıldıktan sonra açılan tarih penceresi kapanmamaktadır. Bu da görselde gözüktüğü gibi diğer alanların (Öğrenci Şu anki Sınıfı, Öğrenci Şu anki Okulu, Kaydı Planlanan Okul, Kaydı Planlanan Sınıf) gözükmesini engellemektedir. Bu engel otomasyonun diğer alanlar için seçim yapılamamasına yol açmaktadır.

Doğum Tarihi

Öğrenci Doğum Tarihi

« Mayıs 2021 »

Pt	Sa	Ça	Pe	Cu	Ct	Pa
17	26	27	28	29	30	1
18	3	4	5	6	7	8
19	10	11	12	13	14	15
20	17	18	19	20	21	22
21	24	25	26	27	28	29
22	31	1	2	3	4	5

Bugün

Öğrenci Şuanki Sınıfı

Öğrenci Şuanki Okulu

Kayı Planlanan Okul

Kayı Planlanan Sınıf

Veli Adı

Şekil 16.1. Doğum tarihi alanına tıklandığında açılan pencere gösterilmektedir.

Çözüm

Çözüm için “fillStudentBirthDateArea” methodunu yazıldı. İlk olarak sendKeys ile doğum tarihini alındı. Actions sınıfından yeni bir action nesnesi oluşturuldu. Sonra fareyi bilinen son fare koordinatlarının belirtilen ofsetine hareket ettirildi.

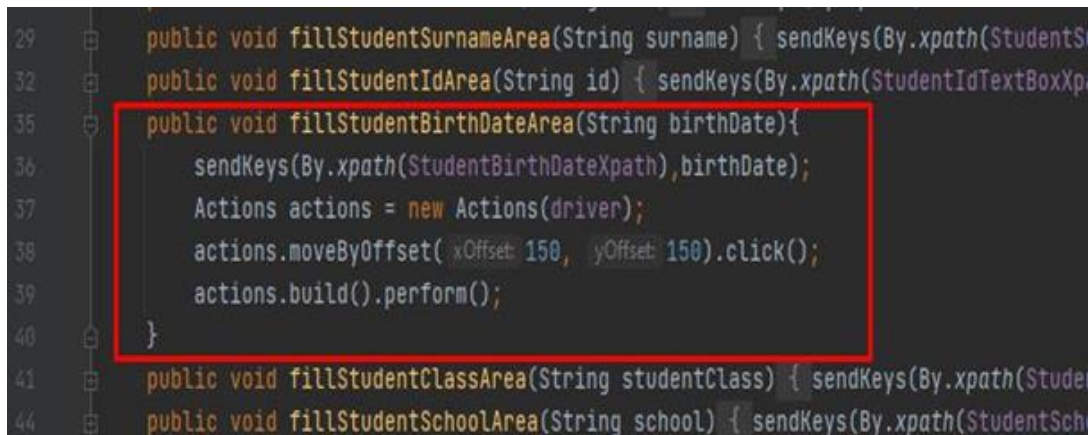
16.2. MoveByOffset Methodunun İncelenmesi

moveByOffset (x-ofset, y-ofset) : Verilen ofset ile fareyi mevcut konumundan kaydırır.

xOffset: Farenin hareket ettirileceği yatay uzaklık. Negatif bir değer ve fareyi sola hareket ettirmek anlamına gelir.

yOffset: Farenin taşınacağı dikey uzaklık. Negatif bir değer ve fareyi aşağı hareket ettirmek anlamına gelir.

Build().Perform() : Eylemler sınıfını derlemek ve yürütmek için kullanılır.



```

29 public void fillStudentSurnameArea(String surname) { sendKeys(By.xpath(StudentS
32 public void fillStudentIdArea(String id) { sendKeys(By.xpath(StudentIdTextBoxXp
35 public void fillStudentBirthDateArea(String birthDate){
36     sendKeys(By.xpath(StudentBirthDateXPath), birthDate);
37     Actions actions = new Actions(driver);
38     actions.moveByOffset(xOffset: 150, yOffset: 150).click();
39     actions.build().perform();
40 }
41 public void fillStudentClassArea(String studentClass) { sendKeys(By.xpath(Stude
44 public void fillStudentSchoolArea(String school) { sendKeys(By.xpath(StudentSch

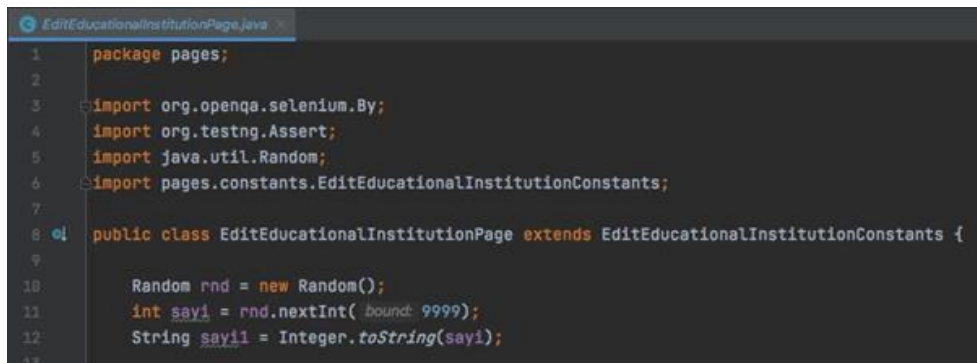
```

Şekil 16.2. moveByOffset methodunun kullanımı gösterilmektedir.

BÖLÜM 17. RANDOM METHODU

Edit işlemi yapılan sayfalarda “Random Methodu” kullanıldı.

Çünkü projedeki tüm senaryoları çalıştırıp hata var mı gözlemlemek istediğimizde yani “Bugfix Çalışmaları” mızda bu method sayesinde sistem hata vermemektedir. Random methodu sayesinde her çalıştırmada fonksiyona farklı bir parametre vermiş olmaktadır ve bunu test etmiş olmaktadır.



```

1 package pages;
2
3 import org.openqa.selenium.By;
4 import org.testng.Assert;
5 import java.util.Random;
6 import pages.constants.EditEducationalInstitutionConstants;
7
8 public class EditEducationalInstitutionPage extends EditEducationalInstitutionConstants {
9
10     Random rnd = new Random();
11     int sayi = rnd.nextInt( bound: 9999);
12     String sayi1 = Integer.toString(sayi);
13

```

Şekil 17.1. Page class’ ında “Random” tanımlaması



```

public void editEducationalInstitutionName(String editName){
    String oldName = getAttribute(By.xpath(EducationalInstitutionNameXpath), attribute: "value");
    sendKeysClear(By.xpath(EducationalInstitutionNameXpath));
    String sayi2 = editName + ""+ sayi1;
    sendKeys(By.xpath(EducationalInstitutionNameXpath), sayi2);
    clickJS(By.xpath(EditEducationalInstitutionSaveButtonXpath));
    wait( second: 1);
    String newName = getAttribute(By.xpath(EducationalInstitutionNameXpath), attribute: "value");
    System.out.println(oldName);
    System.out.println(newName);
    Assert.assertNotEquals(newName, oldName);
}

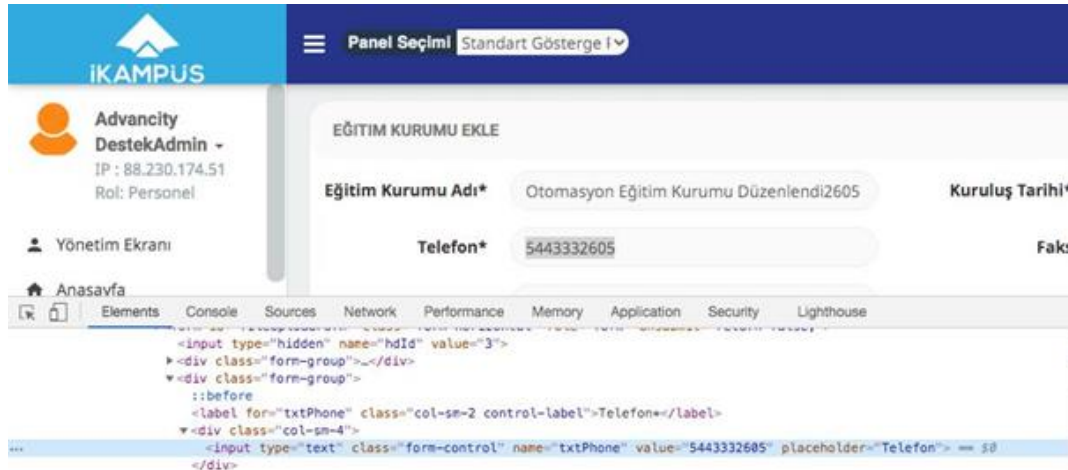
```

Şekil 17.2. Random methodunun fonksiyondaki işleyişi

NOT: “clickJS(By.xpath(EditEducationalInstitutionSaveButtonXpath));”ten sonra “ wait(second: 1); “ yazıldı.

Kaydet işlemlerinden sonra sayfa yenileme yapılıyorsa bunu yaparak sistemin değişiklikleri algılamasını sağlamaktadır.

BÖLÜM 18. ATTRIBUTE



Şekil 18.1. XPath' lerde Attribute' lar

Yukarıdaki şekilde “Elements” altındaki mavi şeritli kısma bakacak olursak input içerisindeki; “ type, class, name, value, placeholder “ bunların her biri Attribute olarak adlandırılmaktadır.

```
public void editTaxNumber(String editTaxNumber) {
    String oldTaxNumber = getAttribute(By.xpath(TaxNumberXPath), attribute: "value");
    sendKeysClear(By.xpath(TaxNumberXPath));
    String sayi2 = editTaxNumber + " " + sayi1;
    sendKeys(By.xpath(TaxNumberXPath), sayi2);
    clickJS(By.xpath(EditEducationalInstitutionSaveButtonXPath));
    wait(second: 1);
    String newTaxNumber = getAttribute(By.xpath(TaxAdministrationAreaXPath), attribute: "value");
    System.out.println(oldTaxNumber);
    System.out.println(newTaxNumber);
    Assert.assertNotEquals(newTaxNumber, oldTaxNumber);
}
```

Şekil 18.2. Attribute' lerin senaryoda kodlaması yapılırken getAttribute fonksiyonu ile kullanıldı.

Özellikle de “Edit” sayfalarında çok işe yaradığı söylenebilir. Eski verilerle yeni girilen verilerin aynı olup olmadığını karşılaştırma yaparken bundan yararlanıldı.

BÖLÜM 19. RUNNER

Projede birden fazla ekip arkadaşıyla çalışabiliriz. Sadece kendi senaryolarımızı çalıştırmak istediğimizde yeni bir Runner eklememiz gerekiyor.

19.1. Birden Çok Test Senaryosunu Çalıştırma

Tags = "not @DeveloperByNaime" yazılarak, Naime adlı ekip arkadaşının yazdığı feature' lar dışındakilerin çalıştırılmak istenildiği belirtildi.

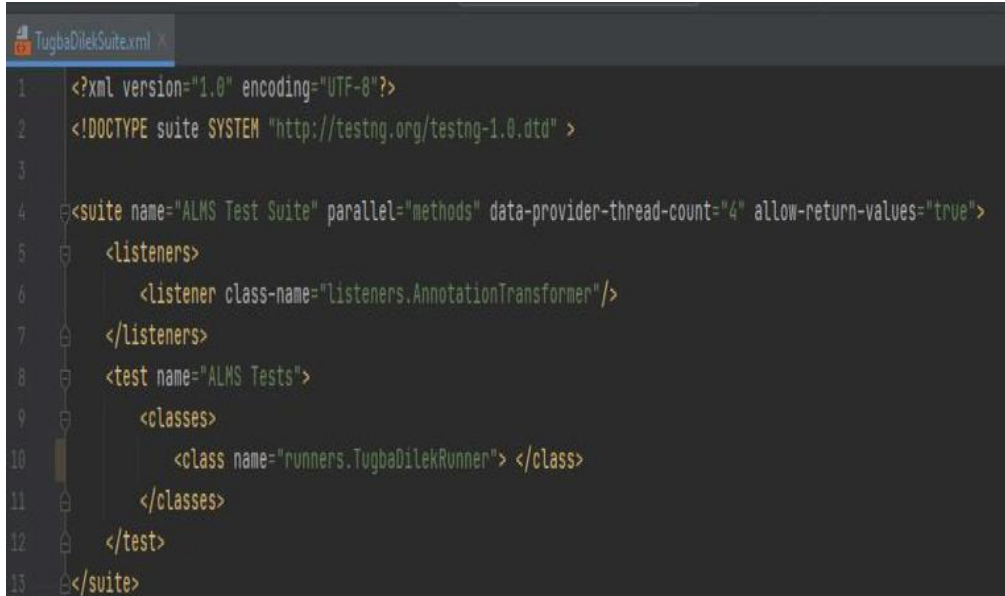
Eğer yalnızca Naime adlı ekip arkadaşının yazdığı feature' ları çalıştırmak isteniyor olsaydı Tags = " @DeveloperByNaime" yazılacaktı.

```

1 package runners;
2
3 import ...
4
5
6
7 @CucumberOptions(
8     features = {"src/test/java/features"},
9     glue = {"stepDefinitions", "hooks"},
10    strict = true,
11    tags = "not @DeveloperByNaime",
12    plugin = {
13        "pretty",
14        "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:",
15        // "timeline:test-output/test-output-thread"
16    }
17 )
18 public class TugbaDilekRunner extends AbstractTestNGCucumberTests {
19
20     @Override
21     @DataProvider(parallel = true)
22     public Object[][] scenarios() { return super.scenarios(); }
23
24
25
26 }

```

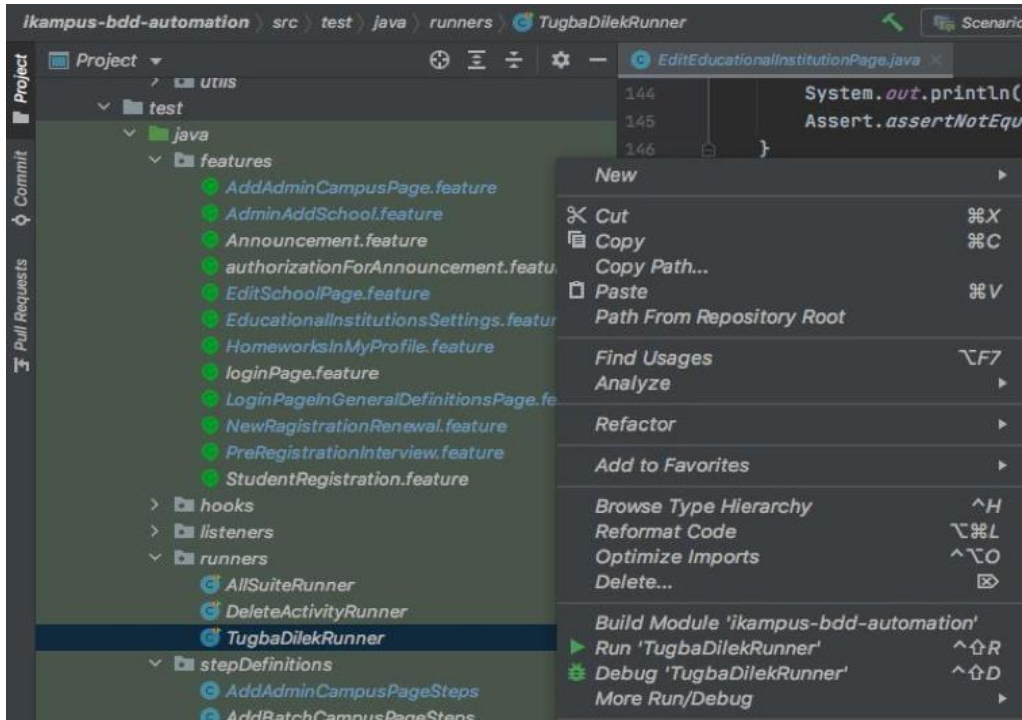
Şekil 19.1. İstemediğimiz feature' ların çalışmasını önleme



Şekil 19.2. Suites klasöründe yeni oluşturulan TugbaDilekRunner.java tanımlandı.

Data - provider-thread-count: Projedeki senaryoların eş zamanlı olarak kaç tanesinin çalışacağını belirtildiği alan. Örnekte 4 adet senaryonun eş zamanlı olarak çalıştırılması sağlandı.

Avantajı: Büyük projelerde senaryoları tek tek çalıştırmak vakit kaybı yaratacaktır.



Şekil 19.3. Runner klasörü altındaki TugbaDilekRunner.java çalıştırılması.


```

✓ runScenario["new registration renewal", "Optional[New registration renewal page cases feature]"] (26)
✗ runScenario["admin my homework", "Optional[File Homework]"] (27)
✓ runScenario["New registration interviews", "Optional[New registration renewal page cases feature]"] (28)
✓ runScenario["Pre-registration interview list must be displayed with successfully", "Optional[pre-registration interview feature]"] (29)
✓ runScenario["edit interview", "Optional[pre-registration interview feature]"] (30)
✗ runScenario["Admin add educational institution", "Optional[Educational Institutions Settings]"] (31)
✓ runScenario["pre-registration interview conclusion", "Optional[pre-registration interview feature]"] (32)

```

Şekil 19.4. Run işleminden sonra testte çıkan hatalar görüntüleniyor.

```

✗ runScenario["Admin add educational institution", "Optional[Educational Institutions Settings]"] (14)
✗ runScenario["Admin educational institution list", "Optional[Educational Institutions Settings]"] (15)
✗ runScenario["Admin edit educational institution", "Optional[Educational Institutions Settings]"] (16)
✗ runScenario["Admin delete educational institution", "Optional[Educational Institutions Settings]"] (17)
✗ runScenario["Admin add educational institution", "Optional[Educational Institutions Settings]"] (18)
✓ runScenario["Add file homework", "Optional[File Homework]"] (19)
✓ runScenario["adding homework in text format", "Optional[File Homework]"] (20)
✓ runScenario["homeworks that I have given", "Optional[File Homework]"] (21)
✗ runScenario["Admin Has Edited That Given Homeworks", "Optional[File Homework]"] (22)
✓ runScenario["Admin Has Deleted That Given Homeworks", "Optional[File Homework]"] (23)
✗ runScenario["admin my homework", "Optional[File Homework]"] (24)
✓ runScenario["Admin Past Homeworks", "Optional[File Homework]"] (25)
✓ runScenario["Login page in the general definitions", "Optional[Login page in general definitions feature]"] (26)
✓ runScenario["Delete image in the login page", "Optional[Login page in general definitions feature]"] (27)
✗ runScenario["new registration renewal", "Optional[New registration renewal page cases feature]"] (28)

```

Şekil 19.5. Run işleminden sonra testte çıkan hatalar görüntüleniyor.

19.2.TestNG

Selenium WebDriver: Test Durumu Yönetimi ve Rapor Oluşturma için TestNG Bu Selenium WebDriver öğreticisi, test senaryolarını yönetmek ve ayrıntılı test raporları oluşturmak için TestNG'yi Selenium ile kullanma ihtiyacını anlamanıza yardımcı olacaktır [10].

```

AllSuite.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
3
4  <suite name="ALMS Test Suite" parallel="methods" data-provider-thread-count="4" allow-return-values="true">
5      <listeners>
6          <listener class-name="listeners.AnnotationTransformer"/>
7      </listeners>
8      <test name="ALMS Tests">
9          <classes>
10             <class name="runners.AllSuiteRunner"/></class>
11          </classes>
12      </test>
13  </suite>
14

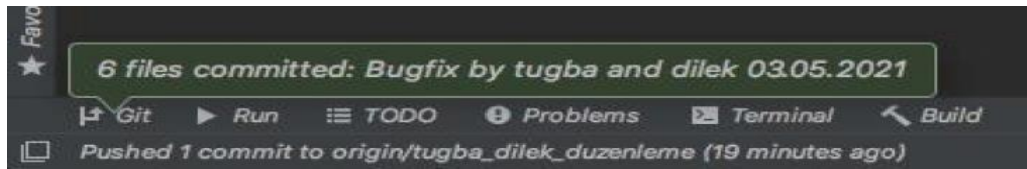
```

Şekil 19.6. AllSuite.xml içerisine TestNG eklenmesi.

BÖLÜM 20. BUGFIX

Proje Run edildikten sonra çıkan hatalarda yeni çözümler üreterek hataları düzeltme işlemi yapıldı. Bu işleme “Bugfix” deniliyor. Bugfix ortalama 1-2 haftada bir düzenli olarak yapıldı. Senaryolar bittikçe genel çalışmada problem varsa önce onlar düzeltildi daha sonra senaryolar yazılıp kodlamaya devam edildi.

“Bugfix” yapıldıktan sonra Merge işlemi yapabilmek için yapılan değişiklikler commit and push yapıldı.



Şekil 20.1. Bugfix commit edildi.

BÖLÜM 21. SONUÇLAR VE ÖNERİLER

Hazırlanan bu dökümandan yola çıkarak, Selenium’da işimize yarayacak faydalı bilgilere ulaşabiliriz. Selenium’a özgü fonksiyonların işleyişi hakkında bilgi edinebiliriz. JUnit Assertions mesajlarının otomasyonda kullanımını öğrenebiliriz. Cucumber kullanımının test senaryolarını kodlarken, cümleler halinde yazılmasının kodun anlaşılabilirliğini arttırılmasındaki faydasını öğrenebiliriz. Takım arkadaşlarıyla çalışırken GitHub kullanımının faydasını gözlemleyebiliriz.

Yazılım testi, geliştirme aşamalarında yapılan kusurları ve hataları belirtmek için gereklidir. Müşterinin kurumu güvenilir bulması ve uygulamadaki memnuniyetinin sürdürülmesini sağlamak için yazılımları test etmek önerilir. Müşterilere teslim edilen ürün kaliteli ise, güvenlerini kazanmaya yardımcı olacaktır. Düşük kaliteli yazılım, ürünün benimsenmemesine neden olabilir ve bu, işletmenin telafi edemeyeceği kayıplarla sonuçlanabilir. Ayrıca yazılım uygulamasının veya ürünün etkin performansı için test yapılması önerilir.

KAYNAKLAR

- [1] <https://junit.org/junit5/docs/current/user-guide/>
- [2] <https://github.com/mertcanyuksekkaya/ikampus-bdd-automation>
- [3] <https://cucumber.io/docs/gherkin/reference/>
- [4] <https://testng.org/doc/>
- [5] <https://chrome.google.com/webstore/detail/xpath-helper/hgimnogjllphhhkhlmebbmlgjoedpjl?&hl=tr>
- [6] <https://www.selenium.dev/documentation/en/>

ÖZGEÇMİŞ

Tuğba ULUSOY, 18.08.1998 de Isparta’da doğdu. İlkokul eğitimini Isparta’da, ortaokul ve lise eğitimini ise Ankara’da tamamladı. 2017 yılında Ankara Kalaba Anadolu Lisesi’nden mezun oldu. 2017 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü kazandı. 2019 yılında TÜRKSAT’ ta “Kurumsal Bilgi Ve Siber Güvenlik Yönetimi Direktörlüğü” nde zorunlu donanım stajını başarıyla tamamladı. 2021 yılında SAÜ Teknokent Köprü Projesi kapsamında CAZ Bilişim A.Ş. (Advancity) ile “Test Otomasyonu” projesinde çalışarak bitirme projesini yaptı. 2021 yılında CAZ Bilişim A.Ş.’ de (Advancity) zorunlu yazılım stajını yapacak. SAÜ Bilgisayar Mühendisliği Bölümü’nden 2021 yılında mezun olacaktır.

Dilek Ayşe GÜN, 01.01.2000 de İstanbul’da doğdu. İlk, orta ve lise eğitimini Kocaeli/Gebze de tamamladı. 2017 yılında Sarkuysan Adalolu Lisesi’nde okul birinciliği ile mezun oldu. 2017 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü kazandı. 2019 yılında Gebze Belediyesi’ nde donanım stajını ve 2020 yılında Cavo Otomotiv’de yazılım stajını yaptı. 2021 yılında SAÜ Teknokent Köprü Projesi kapsamında CAZ Bilişim A.Ş. (Advancity) ile “Test Otomasyonu” projesinde çalışarak bitirme projesini yaptı. Firmayla gönüllü staj şeklinde halen devam etmektedir. SAÜ Bilgisayar Mühendisliği Bölümü’ nden 2021 yılında mezun olacaktır.

BSM 498 BİTİRME ÇALIŞMASI

DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : TEST OTOMASYON PROJESİ

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): G171210017/TUĞBA/ULUSOY

G171210113/DİLEK AYŞE/GÜN

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			

Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN(JÜRİ ADINA): Doç. Dr. Ünal ÇAVUŞOĞLU

DANIŞMAN İMZASI: