

İSTANBUL TEKNİK ÜNİVERSİTESİ
Bilgisayar ve Bilişim Fakültesi

WEB SERVİSLERİNE TEST GELİŞTİRME

STAJ
Tuğrul YATAĞAN
040100117

YAZ / 2013

İstanbul Teknik Üniversitesi
Bilgisayar ve Bilişim Fakültesi
STAJ RAPORU

Akademik Yıl: 2013

Staj yapılan dönem: ☒Yaz ☐Bahar ☐Güz

Öğrenci ile ilgili bilgiler

Adı ve Soyadı: Tuğrul YATAĞAN

Öğrenci Numarası: 040100117

Bölüm: Bilgisayar Mühendisliği

Program: Lisans %100 İngilizce

e-posta adresi: yatagan@itu.edu.tr

(Cep) Tel No: 0555 678 53 57

ÇAP öğrencisi misiniz?: ☐ Evet (ÇAP yaptığınız Fakülte/Bölüm: _____)

☒ Hayır

Mezuniyet ☐ Evet

durumunda mısınız: ☒ Hayır

Yaz okulunda ders ☐ Evet (Ders sayısı: __)

alıyor musunuz? ☒ Hayır

Öğrencinin çalıştığı kurum ile ilgili bilgiler

İsmi: Kartaca Bilişim

Birimi: Ürün ve Sistem Yönetimi

Web Adresi: www.kartaca.com

Kısa adresi: Yıldız Teknik Üniversitesi Davutpaşa Kampüsü Teknopark Bölgesi B Blok
Kat:2 No: 401 Esenler 34220 İstanbul

Yetkili kiři ile ilgili bilgiler

Bölümü: Ürün ve Sistem Yönetimi

Ünvanı: Bölüm Sorumlusu

Adı ve Soyadı: Bekir Doęan

(kurumsal) e-posta: bekir.dogan@kartaca.com

(kurumsal) Tel No: 0212 483 71 71

Yapılan iş ile ilgili bilgiler

Staj yeri ☒ Türkiye

☐ Yurtdışı

Staj başlangıç tarihi 05.08.2013

Staj bitiş tarihi 06.09.2013

Stajda çalışılan net **gün** sayısı 22

Staj süresinde sigortanız var mıydı? ☒ Evet, İTÜ tarafından sigortalandım

☐ Evet, kurum tarafından sigortalandım

☐ Hayır, yurtdışı stajı yaptım

☐ Hayır

1. Giriş

Kartaca Bilişimde sistem yönetici olarak staja başladım. Bana verilen yeni stajyer, Kartaca çalışma düzeni ve kişisel bilgisayar prosedürlerini okudum ve onayladım. Staja başlamadan önce imzaladığım sözleşme gereğince benden şirket dışına kod çıkarmam istenmişti. Bu yüzden staj raporu içerisinde şirket ile ilgili doğrudan kod paylaşmadım. Fakat staj süresince yaptığım şeylere dair genel kodları ve açıklamalarını rapor içerisinde paylaştım.

Staja başladığımda insan kaynakları sorumlusu beni diğer stajyerler ve takımındaki çalışanlar ile tanıştırdı. Şirket içi sistemler için kaydımı yaptırdı (kullanıcı kaydı, parmak izi sistemi kaydı, sunucu erişim yetkileri vb.)

Aşağıdaki eğitim videolarını izledim:

- Teknik Destek Süreci
- SVN, Fisheye, Crucible
- Kartaca Organizasyon Yapısı
- E-posta Kullanımı
- Jira Kullanımı ve İş Akışlarımız
- Takvim ve Çalışma Saatleri
- Masaüstü Çalışma Ortamı
- Confluence Kullanımı ve Kuralları
- Santral ve Telefon Sistemi
- Maliyet Muhasebesi ve Önemi

Diğer yeni stajyer ile beraber toplantıya alındık ve bize şirket ile ilgili genel bilgilerden bahsedildi. Staj sürecimiz anlatıldı. Daha sonra diğer sistem yönetimi stajyerleri ile birlikte LDAP Eğitimine katıldım LDAP nedir, neler yapılır, nasıl kullanılır anlatıldı ve Kartaca'da ne amaçla kullanıldığı anlatıldı.

Staj süresince benden istenen işleri staj projesinin tanımı başlığı altında daha detaylı olarak açıkladım. Stajın tamamı boyunca benden web servisleri için test geliştirmem istendi ve bu yönde bana iki proje verildi. İlk Turkcell Mesajlarım servisi için Selenium testlerinin geliştirilmesi, ikincisi ise Turkcell Müzik servisi yönetici panel için Apache Jmeter testlerinin geliştirilmesi. Bu kararlar doğrultusunda stajıma başladım.

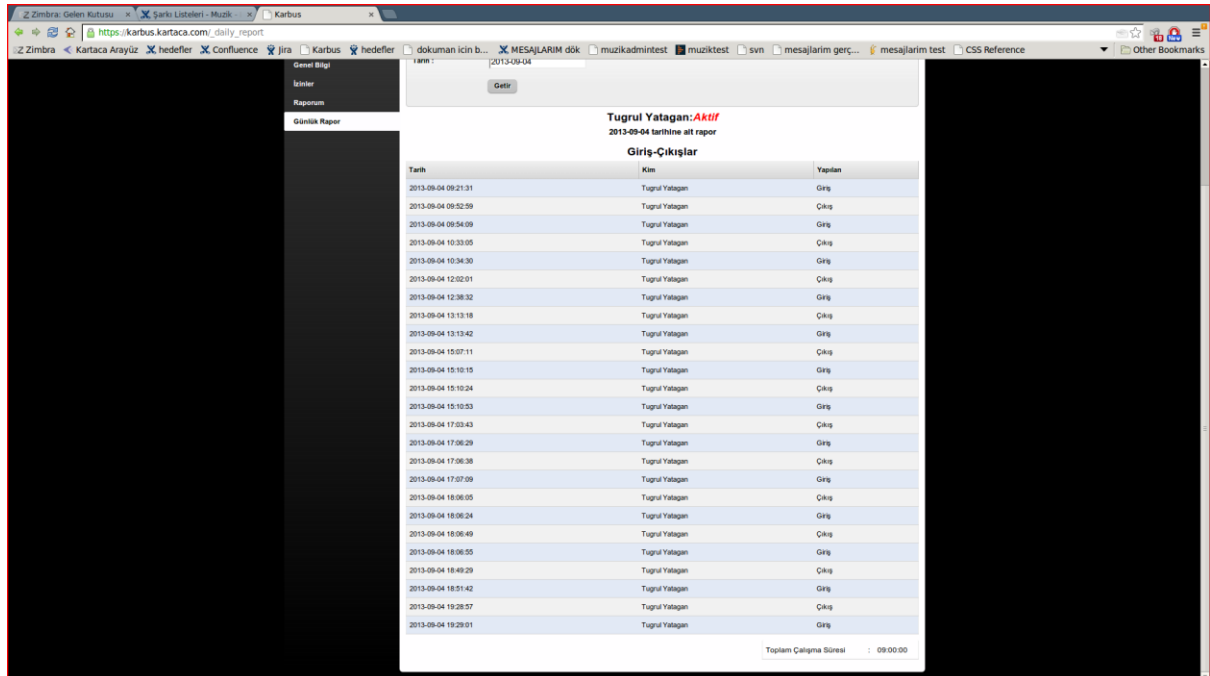
2. Kurumla İlgili Bilgi

Kartaca yaklaşık 50 kişinin çalıştığı bir yazılım şirketi. Çoğunlukla Web teknolojileri üzerine ürünler üretiyor. Parkyeri isimli şirketin bölünmesiyle oluşan şirketlerden birisi ve toplamda 12 yıllık geçmişi var. Kartaca ismi ile ise 4 yıldır aktif. Şirkette 4 ana bölüm ve 4 takım var. Bölümler: uygulama ve servis geliştirme (usg), ürün ve servis yönetimi (üsy), müşteri ilişkileri yönetim (miy), insan kaynakları ve idari işler (ik). Uygulama ve servis geliştirme bölümü tamamen geliştiricilerden oluşuyor. Ürün ve servis yönetimi bölümü ise çoğunlukla sistem yöneticilerinden ve geliştiricilerden oluşuyor. Diğer bölümler yönetimsel işlerden sorumlu. Her bölümün, bir bölüm sorumlusu var ve bölümlerde çalışan kişiler takımlara dağıtılmış durumda. Her takım şirketin farklı projelerinden sorumlu. Her takım içerisinde genellikle 6-7 uygulama ve servis geliştiricisi (geliştirici) ve 2 tane ürün ve servis

yöneticisi (sistemci) bulunuyor. Her takımın bir takım sorumlusu ve 1 veya 2 müşteri ilişkileri yöneticisi bulunuyor. Şirkette bir tane genel müdür var ve takım ve bölüm sorumluları genel müdüre karşı sorumlu. Genel müdür aynı zamanda şirketin en büyük hissedarı yani patronu. Sonuç olarak bir çalışanın bir tane teknik anlamda bağlı olduğu bölüm sorumlusu ve idari anlamda bağlı olduğu takım sorumlusu var. Ben ürün ve servis yönetimi bölümünde staj yaptım ve takımım da Turkcell'e ait projeler üzerinde çalışan son kullanıcı servisleri (sks) takımındı.

Şirket çoğunlukla özgür yazılım teknolojilerini kullanmakta tüm sunucular ve geliştirme ortamları Linux. Çoğu Debian, bir kısmı Red Hat. Geliştirme ortamı ise Debian ve beraberinde birçok masaüstü ortamı kullanılabiliyor. Geliştirme ortamı haricinde şirket tüm geliştirmelerin, geliştirme için kullanılan programlama diline has kullanılan sunucularında yapılmasını istiyor. Örneğin Python ile geliştirme yapılacak ise şirketin Python sunucusuna bağlanıp orada geliştirme yapılmasını isteniyor. Masalarda bulunan bilgisayarlar üzerinde sabit disk bulunmuyor ve masalarda bulunan bilgisayarlar üzerinde geliştirme yapılması istenilmiyor. Ben stajım süresince geliştirme sunucusu olarak sadece Python sunucusunu kullandım. Kodları yazarken editör olarak ise Emacs kullandım. Şirketteki her şey sunucular üzerinden işliyor. Bu sunucuların işletilmesi Kartaca (krt) Takımının sorumluluğunda, şirket içi tim sistemleri Kartaca takımı yönetiyor. Örneğin eğer bir iş için sunucuya erişim izni verilmesini isterseniz Kartaca takımına müşteriymişiz gibi e-posta atmak gerekiyor.

Şirkete giriş ve çıkış saatleri oldukça esnek. Şirketin girişinde parmak okuma sistemi bulunuyor. Giriş ve çıkışlarda parmak izini okutmak gerekiyor dolayısıyla şirket içi geçirilen zaman belirlenebiliyor. Bunun takibi için şirket içinde geliştirilen bir Kartaca Bulunma Sistemi (Karbus) var. Sabah 8-10 arası gelmeniz ve günlük 8 saat, haftalık 40 saati doldurmanız bekleniyor, bunlara uyduğunuz takdirde ne zaman öğle arasına çıktığınız veya akşam kaçta çıktığınız ile kimse ilgilenmiyor. Stajyerler de aynı sisteme tabi.



| Tugrul Yatağan: Aktif | | |
|-------------------------------|----------------|---------|
| 2013-09-04 tarihine ait rapor | | |
| Giriş-Çıkışlar | | |
| Tarih | Kim | Yapılan |
| 2013-09-04 09:21:31 | Tugrul Yatağan | Giriş |
| 2013-09-04 09:52:59 | Tugrul Yatağan | Çıkış |
| 2013-09-04 09:54:09 | Tugrul Yatağan | Giriş |
| 2013-09-04 10:33:05 | Tugrul Yatağan | Çıkış |
| 2013-09-04 10:34:30 | Tugrul Yatağan | Giriş |
| 2013-09-04 12:02:01 | Tugrul Yatağan | Çıkış |
| 2013-09-04 12:38:32 | Tugrul Yatağan | Giriş |
| 2013-09-04 13:13:18 | Tugrul Yatağan | Çıkış |
| 2013-09-04 13:13:42 | Tugrul Yatağan | Giriş |
| 2013-09-04 15:07:11 | Tugrul Yatağan | Çıkış |
| 2013-09-04 15:10:15 | Tugrul Yatağan | Giriş |
| 2013-09-04 15:10:24 | Tugrul Yatağan | Çıkış |
| 2013-09-04 15:10:53 | Tugrul Yatağan | Giriş |
| 2013-09-04 17:03:43 | Tugrul Yatağan | Çıkış |
| 2013-09-04 17:06:29 | Tugrul Yatağan | Giriş |
| 2013-09-04 17:06:38 | Tugrul Yatağan | Çıkış |
| 2013-09-04 17:07:09 | Tugrul Yatağan | Giriş |
| 2013-09-04 18:06:05 | Tugrul Yatağan | Çıkış |
| 2013-09-04 18:06:24 | Tugrul Yatağan | Giriş |
| 2013-09-04 18:06:49 | Tugrul Yatağan | Çıkış |
| 2013-09-04 18:06:55 | Tugrul Yatağan | Giriş |
| 2013-09-04 18:49:29 | Tugrul Yatağan | Çıkış |
| 2013-09-04 18:51:42 | Tugrul Yatağan | Giriş |
| 2013-09-04 19:28:57 | Tugrul Yatağan | Çıkış |
| 2013-09-04 19:29:01 | Tugrul Yatağan | Giriş |

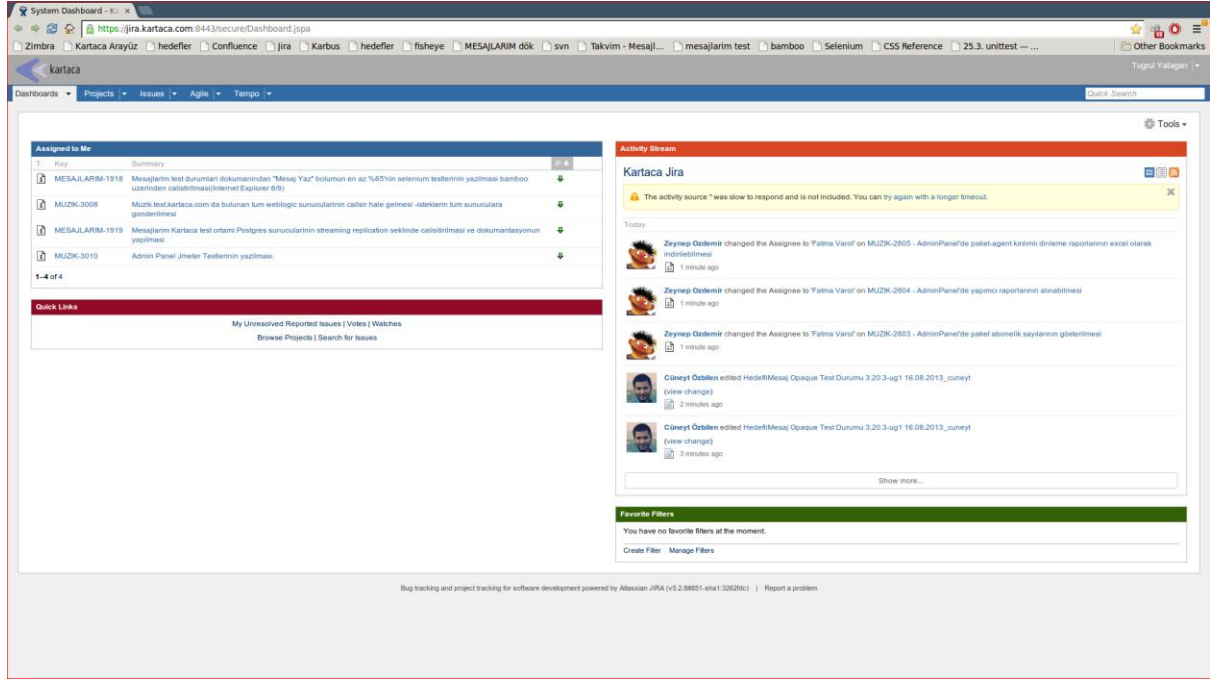
Toplam Çalışma Süresi : 09:00:00

Resim 1 Karbus sistemindeki giriş çıkış kayıtlarım

Şirkette beni kapıya en yakın masaya yerleştirdiler ve bazen dışarıdan gelenler, yeni stajyerler veya yeni çalışanlar geldiğinde parmak kayıtlarının alınması birkaç günü bulabiliyordu. Kapı da sadece parmak iziyle açıldığı için parmak izi kayıtları alınmamış bu kişilerin giriş çıkışlarını genellikle ben sağlıyordum, böyle günlerden birisinde benim giriş-

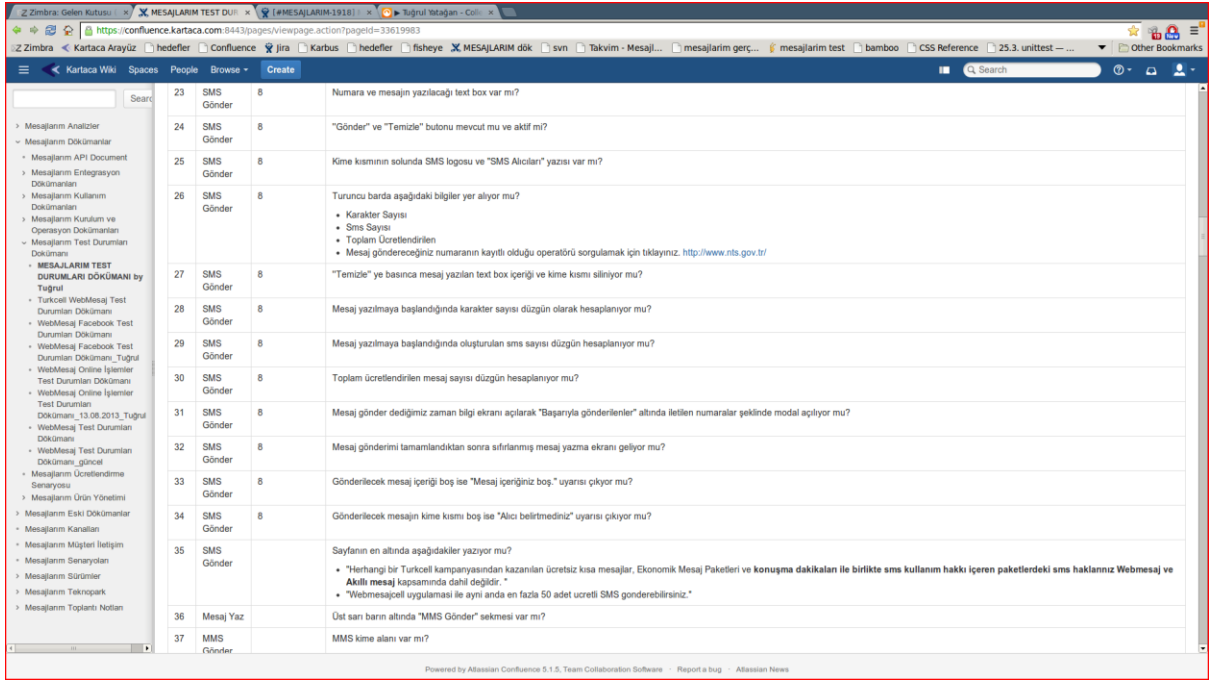
ıkış kayıtlarım sayfalarca olabiliyordu. Resim-1 de grlebileceęi gibi 5-10 saniyelik giriř ıkışlarının sebebi budur. Kartaca kk-orta byklkte bir řirket olmasına raęmen iře bařlayan iřten ayrılan personel sayısı bir hayli fazla, kısa staj sresince bile bu belli oluyordu.

řirket, iřlerin takibi iin Jira adlı yazılımı kullanmakta. Yapılan her iř ve alıřma Jira zerinden kayıt altına alınıyor. Herkesin yapması gereken iřler o kiři zerine atanıyor ve herkes ne yaptıysa Jira zerinde worklog olarak giriyor. Bu staj raporunu yazarken Jira logları olduka iřime yaradı.



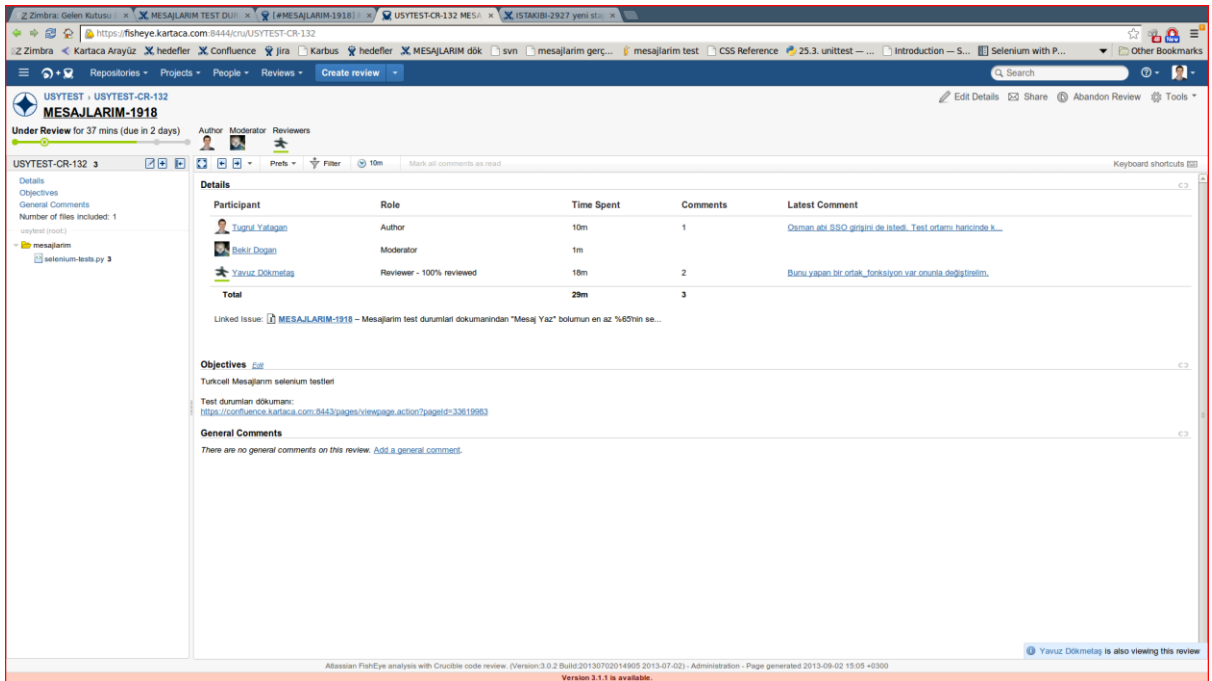
Resim 2 Jira iř takip sistemi

Belgelerin depolanması iin Confluence adlı yazılım kullanılıyor ve tm belgeler bu sistem zerinde toplanıyor. Tm projelere, řirketin iřleyiřine ve řirket ii prosedrlere dair belgeler Confluence ierisinde bulunmakta. Confluence, řirkete adapte olma srecini bir hayli kolaylařtırıyor.



Resim 3 Confluence belgeleme sistemi

Şirkette yapılan tüm geliştirmeler SVN sürüm takip sistemine yükleniyor, tüm kodlar SVN üzerinde tutuluyor. Yazılan tüm kodlar Fisheye gözden geçirme sistemi üzerinden kodu yazan kişi dışındaki biri veya birileri tarafından gözden geçiriliyor ve onaylanırsa SVN içindeki ana depoya ekleniyor.



Resim 4 Fisheye kod gözden geçirme sistemi

SVN içine atılan bazı projelerin kodları Bamboo sürekli bütünleştirme ve kurulum sistemi tarafından otomatik olarak derlenip test sunucularına kuruluyor ve projelere ait testler otomatik olarak çalışmaya başlıyor. Yani bir geliştirici bir projenin kodunda herhangi bir değişiklik yapıp SVN'e kodu attığı anda tüm sistem otomatik olarak çalışıyor ve test sonuçları Bamboo üzerinden takip edilebiliyor. Tabi ki eğer testleri etkileyecek bir değişiklik yapılmış ise testlerin de güncellenmesi gerekiyor.

Şirkette tarayıcı tabanlı testler için Python ve Selenium framework'ü kullanılıyor. Yük ve api testleri için ise Apache Jmeter kullanılıyor. Selenium testleri Bamboo ortamı üzerinde çalıştırılabilir. Selenium testleri daha detaylı ve tarayıcı tabanlı olduğu için test sonuçları daha güvenilir oluyor fakat testlerin geliştirilme süresi uzun oluyor. Jmeter ise sadece temel http metodlarını kullanarak servislere istek yapıp ve cevapları kontrol ediyor ve Selenium testlerine göre geliştirme süresi daha kısa sürüyor.

Benim içerisinde bulunduğum SKS takımı iki projeden sorumlu idi. İlki [Turkcell Müzik](#) servisi diğeri ise [Turkcell Mesajlarım](#) servisi. Ben stajımın çoğunu Turkcell Mesajlarım servisinin web ara yüzü için tarayıcı testlerini yazmakla geçirdim. Mesajlarım servisi kullanıcıların internet üzerinden SMS/MMS atabileceği, okuyabileceği ve adres defteri oluşturabileceği bir servis. Turkcell Müzik ise kullanıcıların Türkçe müzik dinleyebileceği bir servis. Diğer takımların hangi projelerden sorumlu olduğu hakkında hemen hemen hiçbir fikrim yok.

3. Staj Projesinin Tanımı ve Analizi

Staj süresince benden resmi olarak istenen işler:

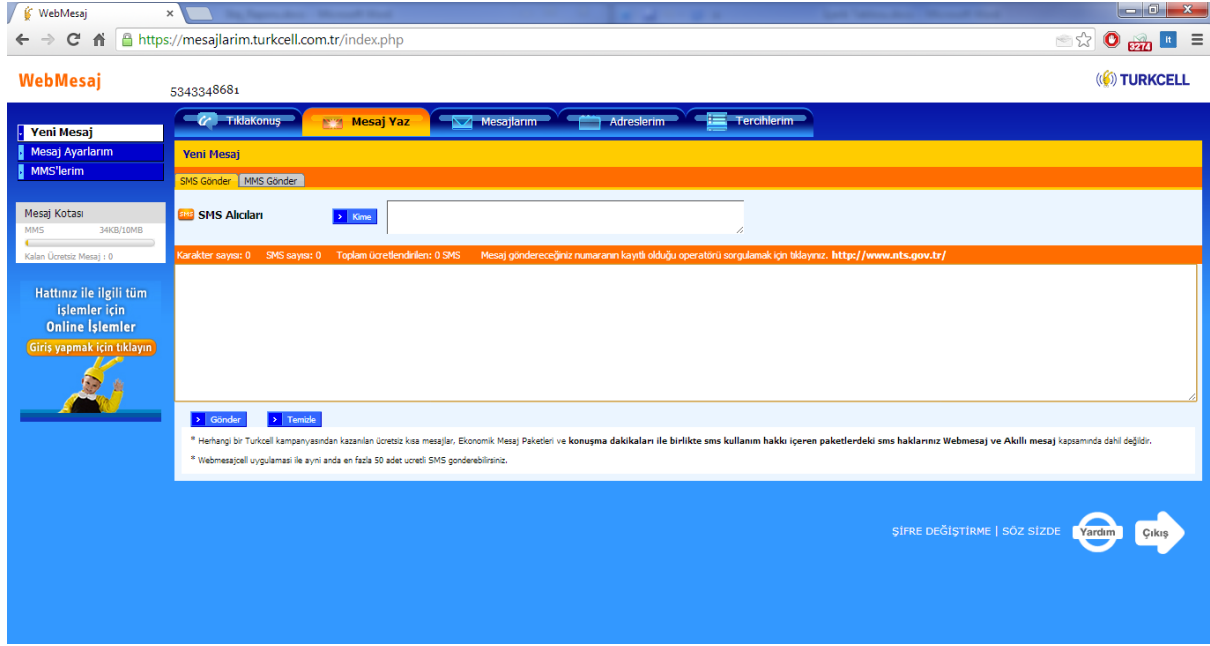
- Turkcell Mesajlarım test durumları dokümanının tamamlanması.
- Turkcell Mesajlarım test durumları dokümanından "Mesaj Yaz" bölümünün en az %65'nin Selenium testlerinin yazılması, Bamboo üzerinden Internet Explorer 8/9 için çalıştırılması.
- Turkcell Müzik yönetici panel test durumları dokümanının tamamlanması.
- Turkcell Müzik yönetici panel Jmeter testlerinin yazılması.

Stajın tamamı boyunca test yazılması ile ilgilendim. Stajın büyük bir bölümünde Mesajlarım servisinin Selenium testlerini yazmakla geçirdim. Testleri yazmaya başlamadan önce her zaman test durumları dokümanını baştan oluşturmam veya var olan dokümanı güncellemem istendi. Testlerin geliştirirken de bu test durumları dokümanlarına göre ilerledim.

Mesajlarım test durumları dokümanının yazılması

Test durumları dokümanını yazmadan önce Mesajlarım ürünü hakkında Confluence'daki tüm dokümanları okudum. Mesajlarım test durumları dokümanının yeni Confluence dokümanını oluşturdum.

Doküman; Giriş, Header, Footer, Tıklakonuş, Mesaj Yaz, Mesajlarım, Adreslerim başlıklarından oluşuyor. Resim-5 de de görüleceği üzere Giriş, Header ve Footer ana sayfada bulunan ve tüm bölümlerde ortak bulunan elemanların kontrolü için bulunan test durumlarını belirtiyor. Tıklakonuş, Mesaj Yaz, Mesajlarım ve Adreslerim başlıkları ise ana başlıklar.

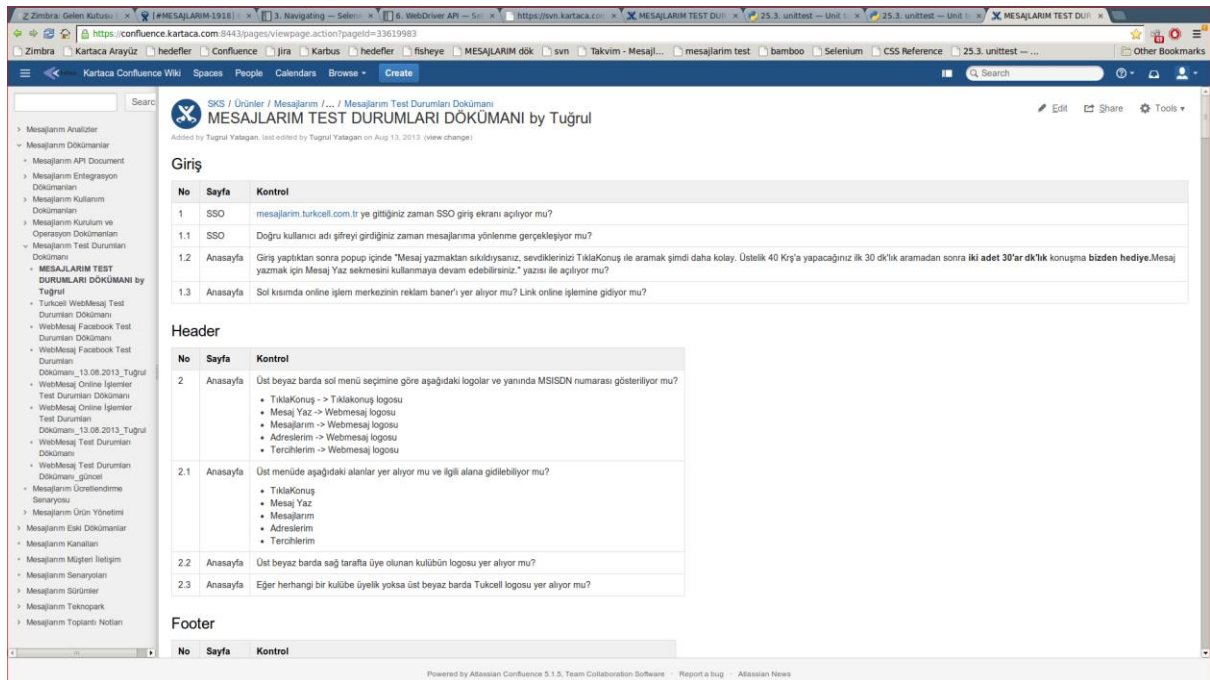


Resim 5 Turkcell Mesajlarım servisi

Dokümanda bulunan birkaç test durumu örneği:

- Sol kısımda Turkcell Online İşlemler Merkezinin reklamı yer alıyor mu? Bağlantı online işlemler merkezine gidiyor mu?
- Soldaki gri bar içinde Kalan Ücretsiz Mesaj sayısını gösteriyor mu?

Daha sonradan test durumları dokümanına test programındaki fonksiyon isimlerini ve test durumları dokümanında hangi maddeye karşılık geldiğini de ekledim.



Resim 6 Mesajlarım test durumları dokümanı

Mesajlarım Selenium testlerinin Python geliştirilmesi

Takımdaki diğer sistemciden Mesajlarım servisi hakkında bilgi aldım. Bana mesajlarım ürününün kurulum ve entegrasyonunu anlattı. Kartaca test ortamını ve sunucuları gösterdi. Nelere dikkat etmem gerektiğini anlattı. SKS takımının tüm test sunucuları üzerinde “sudo” yetkim olduğu için dikkatli olmam gerektiğini söyledi. Mesajlarım kurulum dokümanı üzerinden test ortamında uygulamanın nasıl çalıştığını ve nasıl yapılandırıldığını anlattı, test servisinin entegrasyon noktalarından bahsetti. Test servisi test amaçlı olmasına rağmen yine de Turkcell’in sunucularına bağlı idi. Rehber bilgileri ve ücretlendirme işlemleri için test ortamı Turkcell sunucularına bağlanıyordu. Gönderilen SMS/MMS başına cüzi bir miktar (1 kuruş) olarak şirketin test hattındaki hesaptan düşürülüyordu.

Selenium ile ilgili internetten dokümanları inceledim ve önceki projelerin kodlarını gözden geçirdim, denemeler yaptım. Şirkette sadece testler ile ilgilenen çalışan bana ve diğer birkaç stajyere Selenium, unittest ve Bamboo eğitimi verdi. Selenium için var olan birkaç metottan bahsetti. Selenium IDE ve Selenese arasındaki farkı anlattı. Şirkette Selenium IDE kullanmayacağımızı sadece kod tabanlı olan Selenese ve Python ile testleri yazacağımızı söyledi. Testleri nasıl yazacağımızı ve nasıl çalıştıracağımızı gösterdi. Testleri Python sunucusunda yazıp çalıştıracağımızı ve bu betiklerin client tarafında ise Windows Server 2008 üzerinde komutları nasıl işleyeceğini gösterdi. Testleri Internet Explorer üzerinde çalıştırabilmek için Windows Server kullanıldığından bahsetti.

Makinalara erişimlerimi kontrol ettim, erişimim olması gereken fakat yetkim olmayan makinelerde yetkilerimin açılması için Kartaca ekibini bilgilendirdim. Şirkete ait SVN, Bamboo ve Selenium dokümanlarını inceledim. SVN içerisinde sistem yöneticilerin testlerinin bulunduğu klasörün altında Mesajlarım için yeni bir klasör oluşturup içine deneme amaçlı Selenium testlerini koydum ve Kartaca Windows sunucuları üzerinden deneme testlerini çalıştırdım.

The screenshot shows a Jira issue page for the project 'Mesajlarım'. The issue title is 'Mesajlarım test durumları dokümanından "Mesaj Yaz" bölümün en az %65'nin selenium testlerinin yazılması bamboo üzerinden çalıştırılması(Internet Explorer 8/9)'. The page includes a table of tasks with columns for date, time, description, and progress. The tasks are as follows:

| Date | Time | Description | Progress |
|-----------|-------|--|----------|
| 14/Aug/13 | 10:00 | Mesajlarım test ortamının kurulduğu veri tabanını tutan Populapower makinesi kapamış. kartaca ekibinden tekrar açtığım postgres sunucusunu çalıştırdım | 0.5 |
| 14/Aug/13 | 11:00 | bamboo dokümanlarını ve template betiklerini inceledim | 2 |
| 14/Aug/13 | 13:00 | gerekenler kısmındaki bilgileri bildirdim | 0.25 |
| 14/Aug/13 | 15:31 | Bamboo için otomatik sürüm kurulumu yapmak için | 1 |

The page also includes a comments section with two comments from 'Korhan Müderrisoğlu' dated 05/Aug/13 10:39 AM and 05/Aug/13 10:45 AM. The activity log shows the issue was created by 'Korhan Müderrisoğlu' on 05/Aug/13 10:39 AM.

Resim 7 Mesajlarım Selenium testleri için Jira sayfası

Mesajlarım servisinin testleri için daha önceden var olan bir kod olmadığı için Mesajlarım servisi için Bamboo planı oluşturmak üzere Kartaca ekibine e-posta attım.

Mesajlarım test ortamının kullandığı veri tabanını tutan sunucu kapalıydı bunu için Kartaca ekibinden tekrar açılmasını istedim ve Postgresql veri tabanını aşağıdaki komut ile çalıştırdım.

```
$sudo service postgresql-9.2 start  
$sudo service apache2 restart
```

Bamboo dökümanlarını ve şirket tarafından oluşturulmuş olan Bamboo betiklerini inceledim.

Bamboo’da otomatik sürüm kurulumu yapmak için SVN test kodları altında bamboo-create-release betiğinin yazılması gerektiğini fark ettim. SVN üzerinde kaynak kodda değişiklik yapıldığı anda Bamboo bu bamboo-create-release betiğini çalıştırıp gerekli kurulumu yapacaktı. Bu betik için Mesajlarım servisinin geliştiricilerinden kendilerinin kaynak kodu derleyip, sürümü oluşturan betiklerini istedim. Geliştiricilerin betiği sürümü oluştururken SVN üzerinden tüm kodu çekip (svn checkout) sürümü oluşturduklarını fark ettim. Bölüm sorumluma bu konu hakkında danıştığımda ise Bamboo sunucusunun çalıştığı ortamın SVN’e erişiminin olmadığını, Bamboo ortamının kendine has bir kopya SVN deposunun olduğunu söyledi. Bu yüzden bu betiği olduğu gibi Bamboo ile kullanamayacağımı eğer Bamboo ile kullanırsak yetki hatası nedeniyle betiklerin çalışamayacağını söyledi. Ayrıca mesajlarım servisi ioncube adlı yazılımın kullanılarak PHP kodlarını gizlendiğini anlattı. ioncube çalıştıracak sunucuda bunun için gerekli sertifikanın yüklü olması gerektiğini ve bu sertifikanın ellerinde sadece bir sunucuda yüklü olduğunu söyledi. Yani ioncube sertifikası bizim kullanacağımız Mesajlarım test sunucusunda kurulu değilmiş.

Tüm bu sorunlar için geliştiricilerin kullandığı sürüm oluşturma betiğinde değişiklikler yapmak gerekiyordu fakat benim Turkcell Mesajlarım kod tabanına erişimim olmadığı için kod tabından sürümü oluşturan bir betiği değiştirmem mümkün olmadı. Bu yüzden işin Bamboo üzerinde çalıştırma kısmını kod tabanına erişimi olan başka bir geliştiriciye atanmasına karar verdik.

Selenium ile sayfayı açan ve sayfanın açılışını, Ana sayfadaki logoları ve diğer servislere yönlendirme yapan linkleri kontrol eden testleri yazdım.

Aşağıdaki kod Selenium ile uzak komut işleyici üzerinde Chrome ile istenilen web sayfasını açar. Bu betiği çalıştırmadan önce “rdesktop” uygulaması ile komut işleyici olarak çalışacak Windows Server 2008 ortamına bağlanılıp, Selenium komutlarını işleyecek Java programı çalıştırılır ve belirlenen portu dinlemesi için ayarlanır. Ayrıca komut işleyici üzerindeki tarayıcılarda Selenium komutlarını işlemek için gereken eklentilerin yüklü olması gerekir.

```
from selenium import webdriver
from selenium import DesiredCapabilities

tarayici = webdriver.DesiredCapabilities.CHROME
komut_isleyici = "http://win.kartaca.com:4495/wd/hub"
url = "http://mesajlarim.turkcell.com.tr"
try:
    sel = webdriver.Remote(
        desired_capabilities = tarayici,
        command_executor = komut_isleyici
    )
    sel.implicitly_wait(5)
    sel.maximize_window()
    driver.get(url)
except Exception as e:
    raise e
```

Ana sayfadaki genel başlıkları ve SMS/MMS kotasını kontrol eden testleri yazdım.

The image shows a Selenium test execution environment. On the left, a terminal window displays the test code for the WebMesaj page. The code includes imports for Selenium, a base URL, and several test functions (test_000 to test_005) that perform actions like clicking, waiting, and asserting. On the right, a browser window shows the WebMesaj page, which is the messaging interface for Turkcell. The page includes fields for sending messages, a 'Yeni Mesaj' button, and a list of recent messages. The terminal output shows the test results, indicating that all tests passed successfully.

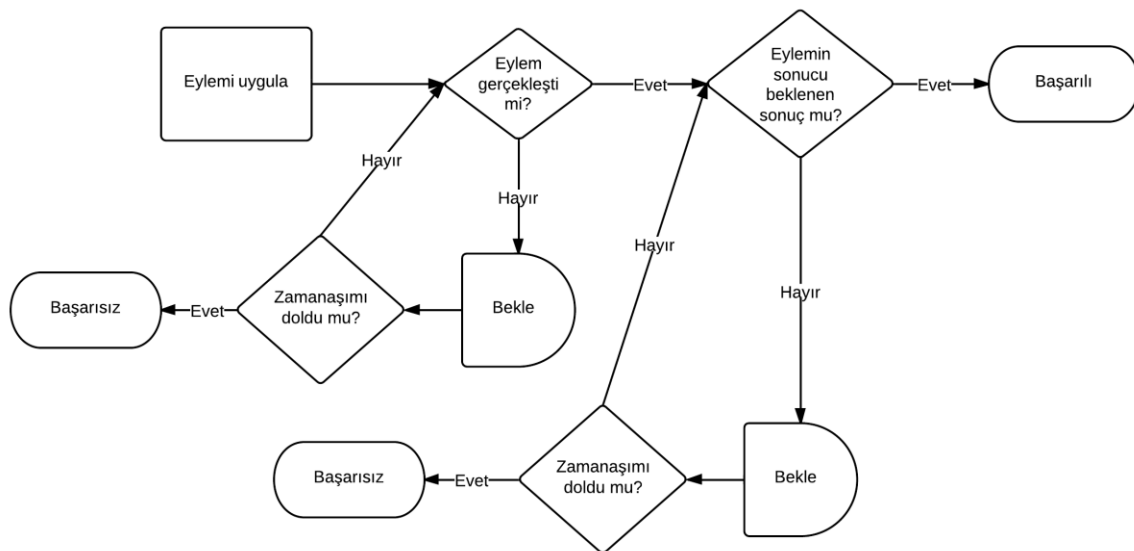
Resim 8 Ana sayfa kontrolleri

Test sunucularının da bulunduğu fiziksel sunucunun sabit disk ışığı sönünce Kartaca ekibi sunucuları bakıma aldı. Şirkete ait yüzlerce sunucu var fakat hepsi sanal, fiziksel olarak sadece 5-10 tane sunucu bulunuyor, yani fiziksel bir cihaz bakıma alındığı zaman birçok sanal sunucu devre dışı kalıyor. Bu süre içerisinde test servisine erişemedim sunucular geri gelince servisi yeniden başlattım. Apache ve PostgreSQL servislerini başlatmak yeterli oldu.

Boş kalan zamanımda (yarım gün) o ana kadar olan kodda fazla kullanılan kısımları fonksiyon haline getirdim, hata mesajlarını ekledim, yorum satırlarını ekledim.

Selenium ile test yazarken sürekli karşılaştığım zorluk zamanlama ile ilgili sorunlardı. Örneğin; Selenium ile rehbera bir kayıt ekleyip ekle butona tıkladığınızda ve o butona karşılık bir uyarı çıkması gerektiğinde, servis ile ilgili sorunlardan dolayı kaydın eklenmesi ve uyarının çıkması uzun sürebiliyor. Kaydın eklenip eklenmediğinin kontrolü ve uyarının çıkıp çıkmadığının kontrolünü işlemler tamamlanmadan önce yapılırsa testler başarısız sonuç verecektir. Hâlbuki kontrol daha geç yapılsaydı testler başarılı sonuç verebilecek durumda olabiliyor. Servisin o anki durumuna göre bu işlemlerin süresi değişebileceği için her seferinde işlemin ne kadar sürebileceğini bilmek zor oluyor ve bir şekilde bunun üstesinden gelmek gerekiyor. Genel yöntem bir zaman aşımı belirleyip döngü içerisinde kontrolleri yapmak ve eğer zaman aşımı içerisinde beklenen içerik oluşmaz ise teste başarısız sonuç verdirmek. Fakat bu yöntem de her zaman çözüm olamayabiliyor. Bazı durumlarda kayıt eklendikten sonra ve uyarı mesajı çıktıktan sonra bile eklenen kaydın tarayıcıda görülebilmesi için servis veri tabanından sonuç dönmesini bekleyebiliyor. Arada geçen bu zaman içerisinde kayıt eklenmiş bile olsa sayfa üzerinde görünür olmuyor. Eğer kaydın eklenip eklenemediğinin kontrolü bu görünür olmadığı aralıkta yapılırsa test yine başarısız sonuç dönüyor. Fakat kontrolleri birkaç saniye sonra yapılırsa kayıt görünür oluyor ve test başarılı sonuç verebiliyor. Testlerdeki kontrolleri sürekli geciktirmek testlerin çok yavaş çalışmasına denenebileceği ve bu gecikmelerin uzunluğu tahmin edilemez olduğu için sürekli beklenen duruma has kodu yazmak gerekiyor. Yani tüm durumları karşılayacak altın bir fonksiyon yazılamıyor.

Testlerin genel çalışma işleyişi yaklaşık olarak şöyle:



Test servisi gerçek ortama göre daha yavaş çalışması oldukça sıkıntı çıkarıyor. Bazı Selenium fonksiyonları işlem yapılmak istenen elemanın yüklenip yüklenmediğini düzgün algılayamıyor bu nedenle sorun çıkartan yerlerdeki işlemleri döngü içerisinde işlem yapılana kadar tekrarlatmak zorunda kalıyorum.

Selenium ile sayfada html elementi (buton, link, yazı kutusu, logo, menü seçimi vb.) bulmanın birkaç metodu bulunuyor. Bunlardan en güvenilir olan yöntemler html id ile veya css seçici ile element bulma. Html id'leri sayfanın kaynak kodu içerisinde rahatlıkla bulunabiliyor. Eğer elemente ait id yok ise o zaman css seçici ile elementi bulmak en iyi yöntem. css seçici kullanarak bulmak istediğimiz elemente en yakın id'ye veya class'a sahip element üzerinden giderek bulmak istediğimiz elemente ulaşabiliyoruz. (css seçiciler için referans: http://www.w3schools.com/cssref/css_selectors.asp)

```
html_element = sel.find_element_by_id("element_id")
html_element = sel.find_element_by_css_selector("css_secicisi")
```

Örneğin okulun web sayfası (<http://www.itu.edu.tr>) üzerindeki hızlı bağlantılar menusunun html kodu:

```
<dl id="sample" class="dropdown">
  <dt>
    <a href="#">
      <span>HIZLI BAĞLANTILAR</span>
    </a>
  </dt>
</dl>
```

Burada doğrudan id üzerinden butona erişebilir, id “sample” ise:

```
html_element = sel.find_element_by_id("sample")
```

Yine okulun sayfasında İLETİŞİM tabı için html kodu:

```
<div class="menu">
  <ul class="menu-ana">
    <li>
      <a target="_self" href="/?iletisim"> İLETİŞİM </a>
    </li>
    ...
  </ul>
</div>
```

Burada ise İLETİŞİM tabı menü sınıfının içerisindeki liste elemanlarından bir tanesi yani listenin tamamına sınıf üzerinden erişebiliriz fakat listenin bir elemanına ulaşmak için listeyi bir obje dizisi olarak alıp içerisinde aradığımız tabı bulmamız gerekir bunun için “menu-ana” sınıfının altındaki “li” elementlerini bulmak gerekir:

```
html_elementleri = sel.find_elements_by_css_selector(".menu-ana>li")
```

İLETİŞİM tabı listenin ilk elementi olduğu için objemiz listenin sıfırıncı elementi olacaktır:

```
html_element = html_elementleri[0]
```

Yukarıdaki fonksiyonlar sonucunda elimize obje veya obje dizisi geçiyor. Bu objeler üzerinden çeşitli işlemler yapılabilir. En çok kullanılan fonksiyonlar:

```
html_element.click()
html_element.send_keys("yazi_veya_deger")
html_element.clear()
html_element.text
html_element.get_attribute("ozellik")
html_element.value_of_css_property("css_ozellik")
html_element.is_displayed()
html_element.is_enabled()
```

Yukarıdaki fonksiyonların ne yaptığı isimlerinden de rahatça anlaşılıyor:

click() : objeye tıklıyor

send_keys() : objeye değer yolluyor, yazı kutucukları bunun ile dolduruluyor

clear() : obje üzerindeki var olan değeri temizliyor

text : objenin yazısını dönüyor

get_attribute() : objedeki istenen özelliği geri dönüyor. Örn: colour

is_displayed() : objenin görünür olup olmadığını kontrol ediyor

is_enebled() : objenin etkin olup olmadığını kontrol ediyor

value_of_css_property() : get_attribute ile bulunamayan css özelliğini geri dönüyor

Testlerde eylemi yapan ve sonucunu kontrol eden sıkça kullanılan fonksiyonlar bunlardı.

Test fonksiyonlarının sırayla ve otomatik olarak çalışmasını ise unittest adındaki modül sağlıyordu.

```
import unittest

class testler(unittest.TestCase): # testler unittest'i miras alır
    @classmethod
    def setUpClass(self):
        # burası testler başlamadan önce çalışır
    def test_001(self):
        ...
    def test_002(self):
        ...
    @classmethod
    def tearDownClass(self):
        # burası testler bittikten sonra çalışır

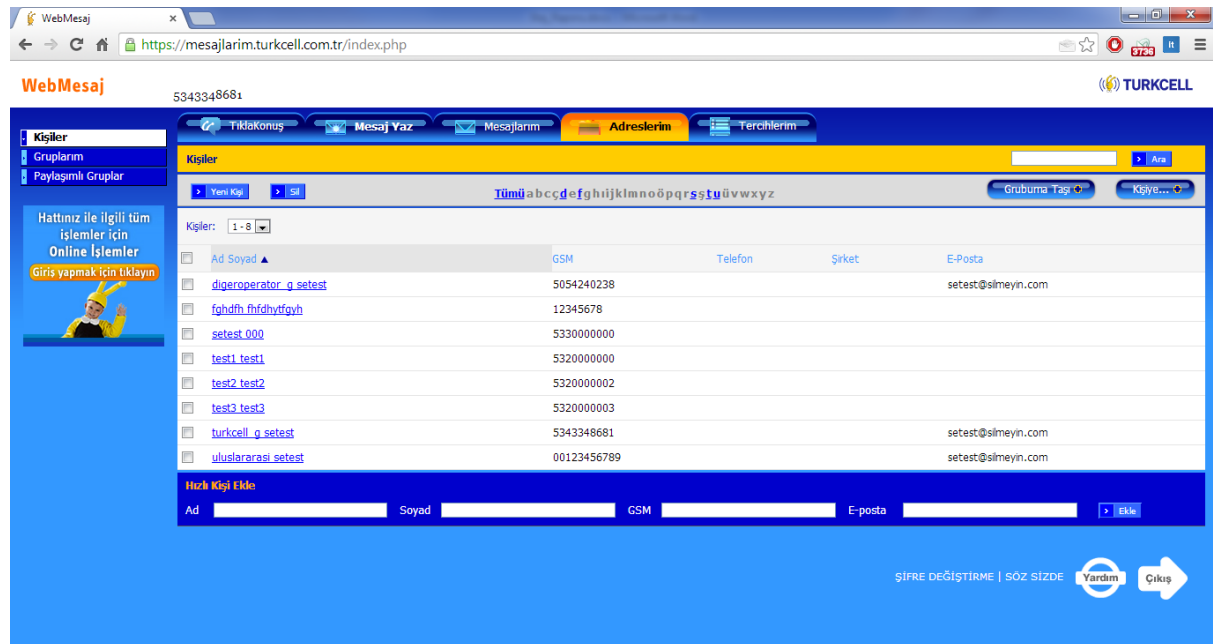
if __name__ == '__main__':
    unittest.main()
```


Unittest için yapılması gereken tek şey testlerin bulunduğu sınıfın unittest sınıfını miras alması ve main fonksiyonun unittest'in main fonksiyonunu çağırması. Setup fonksiyonu testlerin başlangıcında yapılması gereken işler için kullanılır, teardown fonksiyonu ise tüm testler sona erdikten sonra çalışır. Testler hata ile karşılaşsa bile bu fonksiyonlar çalışır. Geri kalan tüm test fonksiyonları alfabetik olarak otomatik olarak çalıştırılır. Test fonksiyonlarının ayrıca başka bir yerde çağırılmasına gerek yoktur.

Mesaj yaz bölümünde (Resim 11) kontrol işlemleri yapmak üzere Adreslerim kısmında kişi ekleme, kişi silme, grup ekleme, grup silme, paylaşımlı grup ekleme işlemlerini yapan ve kontrol eden testleri yazdım.

Mesajlarım servisinde adreslerim bölümüne aynı isim, soy isim ve numarada kayıtların eklenebilmesi test için sorun oluşturdu. Çünkü belirli bir test kişisi eklenip daha sonra eklenip eklenmediğinin kontrolünü yapmak gerekiyor ve eğer önceden aynı isimler bir kullanıcı varsa kişinin eklenip eklenemediğini bilmek mümkün olmuyor. Test esnasında adres defterine eklenecek kullanıcılar eğer tüm test bitmeden yarıda kesilir ise silinmeden kalıyorlar ve bu kullanıcılar bir sonraki test için hata oluşmasına neden oluyor. Bence mesajlarım servisinin aynı kullanıcının birden fazla eklendiğinde uyarı vermesi gerekir fakat vermediği için bu durumu test kodlarında kontrol etmek zorunda kaldım. Test kişilerini testin başında ekleyip sonunda silmek sorun çıkartıyordu. Eğer bir şekilde test yarıda kalırsa(programın bir hatayla karşılaşması durumu veya servis ile ilgili problemler) silinemeyen test kişileri bir sonraki test içinde hatalara neden oluyordu. Bu yüzden test kişilerini testlerin başında ekleyip sonunda silmekten vazgeçtim ve test kişilerini testin başında silip sonra yeniden ekleyerek çözdüm. Eğer test başta varsa silinecekler yoksa da bir problem yaratmayacaklar.

Her test başlangıcında, yavaş olduğu için adreslerim kısmındaki daha önceden var olan tüm kişileri silmedim. Şirkette bu servis için tek test hattı olduğu için deneme yapmak isteyen herkes bu hat üzerinde ekleme çıkartma yapabiliyor. Rehberde aynı isimde olmadığı müddetçe ilave kişi girilebilir ve testler süresince silinmez hatta bazı testler için farklı isimde kişilerin bulunması şart.



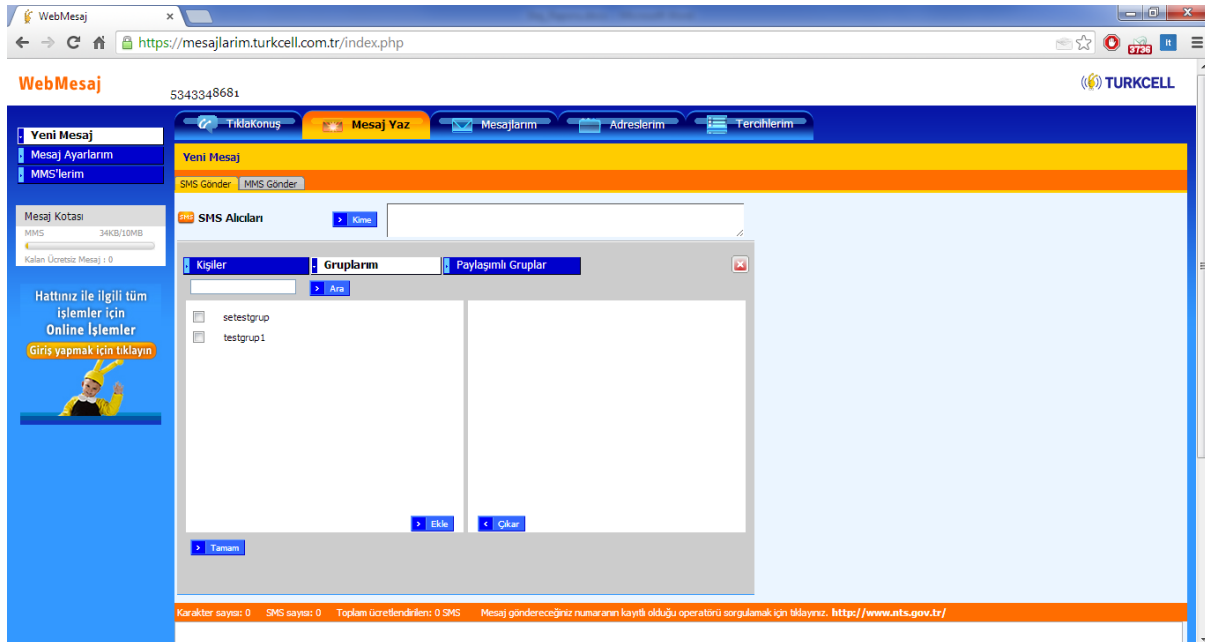
Resim 9 Adreslerim sayfası

Adreslerim kısmında hızlı kişi ekleme işleminden sonra sonra Firefox ve Chrome da çıkmayan bir uyarı mesajı IE8 de çıkması testler için sorun çıkardı. Durumu geliştiricilere sorduğumda bu sıkıntının sadece test ortamında oluştuğunu bunun nedeninin de IE8 de konsol olmadığı için geliştiriciler debug mesajlarını konsol yerine uyarı mesajı olarak ekrana bastırdıklarını söylediler.

Test servisi IE8 üzerinde çalışınca sürekli uyarı mesajı veriyordu. Bunun nedeni geliştiricilerin uyarı mesajlarını debug amaçlı kullanmaları idi. Test servisinin ayar dosyalarından debug modunu kapatmayı denedim fakat debug modu Turkcell SSO geçişini sağladığı için test servisi debug mod kapatılınca çalışmadı. Bu durumu takımımdeki sistemciyle ve şirketin testlerden sorumlu çalışanıyla danıştım. Test sorumlusu sorunun servisin geliştirme seviyesinden kaynaklandığını bunun için servisi uyarı mesajı vermeyecek şekilde değiştirilmesi gerektiğini söyledi aksi takdirde Selenium'un sürekli uyarı mesajı için kontrol etmesi gerektiğini ve bunun çok yavaş çalışacağını söyledi. Takımımdeki sistemci ise test servisinden uyarı mesajlarının temizlenmesinin şimdilik mümkün olmadığını söyledi. Bunun için Selenium testlerini IE8 uyumluluğu olmadan yazmaya devam ettim.

Benim takımım olan son kullanıcı servisleri (sks) takım toplantısına katıldım ilerleyen projeler hakkında konuşuldu Çalarken Dinlet projesinin yeni taslaklarının Turkcell'den onay alındığı söylendi. Sitenin taslak tasarımı anlatıldı. Stajyer olmama rağmen konular hassas olmadığı sürece takım toplantılarına ben de katıldım. Toplantılarda genelde yönetsel konular konuşulduğu için bazen toplantılara niye katıldığımı dahi anlamadım.

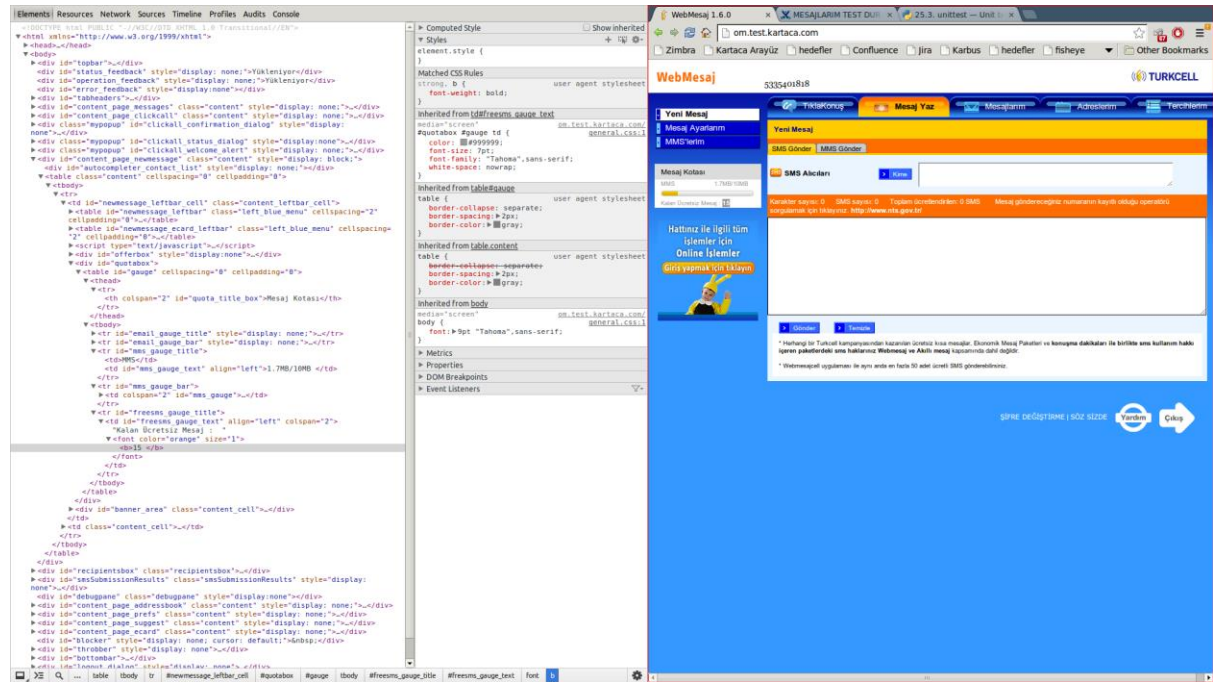
Kime butonu (Resim 10) içerisinde kişiler, gruplarım ve paylaşımlı gruplarım kısımlarının testlerini yazdım. Paylaşımlı grup ekleyemediğimiz için var olan paylaşımlı gruplar üzerinden testleri yazdım. Birileri var olan paylaşımlı grubu silmez ise çok iyi olur çünkü paylaşımlı grup eklemenin tek yolu paylaşımlı grup oluşturmak istenen kişilere giden mesaja karşılık, bu kişilerin onay mesajı yollaması. Bunu da test prosedürü içerisine alıp, sürekli tekrarlamak mümkün değil.



Resim 10 Kime butonu içeriği

Mesajlarım servisindeki SMS/MMS ücret hesaplarını yapabilmek için günlük kotanın her test başlangıcında sıfırlanması gerekiyordu. Takımımdaki sistemci Mesajlarım test ortamının kullandığı Postgresql veri tabanına dışarıdan nasıl bağlanabileceğini anlattı ve pgAdmin'i gösterdi. Ücretlendirme ile ilgili bilgi verdi. Site üzerinde ücretli mesaj sayacının nasıl çalıştığını anlattı. Test ortamında telefonun şebekesini öğrenmek için regex kullandıklarını söyledi. Daha sonra Kartaca test ortamındaki ücretlendirme bilgileri ile gerçek ortam ücretlendirme bilgilerini test servisinin ayar dosyalarından aynı yaptık.

Bölümümdeki sistemciden SMS/MMS ücret hesaplamalar ile ilgili tekrar bilgi aldım. SMS gönder kısmında SMS karakter sayısı, SMS sayısı ve toplam ücretlendirilen SMS sayısı kısımlarının testlerini yazdım.



Resim 11 SMS gönder sayfası

SMS gönderme servisinin SMS ücretlerini hesaplama bölümünde, gün içerisinde Turkcell'iler ile en fazla 15 ücretsiz SMS vermesi test için büyük sıkıntı çıkardı. Testler bu haliyle gün içerisinde en fazla 3 kere çalıştırılabilirdi. Bölümümdeki sistemci ile konuştuğumda bu durumu test servisinin veri tabanına bağlanmadan aşamayacağımızı anladık. Bunun için Kartaca ekibinden Postgresql ile bağlantı için gereken Psycopg2 modülünü Python sunucularına yüklemelerini istedim. Daha sonra her test başlangıcında veritabanındaki gönderilen SMS sayısını sıfırlamak için gerekenleri yazdım.

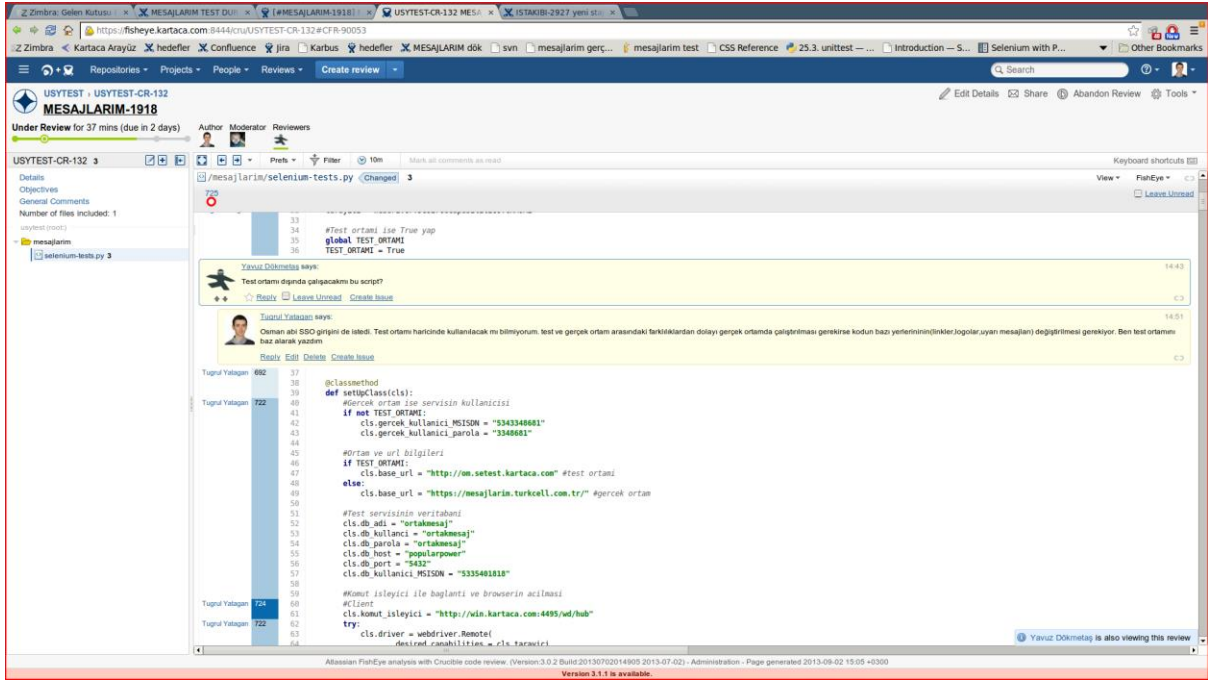
Mesajlarım SMS gönder alanındaki gün içerisinde ilk Turkcell dışı mesaj atılırken verilen “Turkcell içi ilk mesaj ücretli sonraki 15 mesaj ücretsizdir.” uyarısı kararsız çalıştığını fark ettim. Verilmesi gereken uyarı bazen hiç verilmemekte veya bazen Firefox içerisinde birden fazla(5 kere bile olabiliyor) verildiğini fark ettim, fakat sorunu tekrar etmeye çalıştığımız zaman sorunu oluşturan belirli bir patern bulamadık. Birden fazla kez uyarı çıkma sorununu sadece Firefox'ta yaşıyorduk. Hiç uyarı verilmediği takdirde ise sitede kullanıcıları bu konuda uyaracak başka bir uyarı yazısı bulunmuyor. Konu hakkında geliştiricileri bilgilendirdim. Sorununun düzeltilmesi için Jira da iş açıldı fakat ne zaman çözülebileceği belirsizdi çünkü sorun tekrar edilemiyor. Firefox için birden fazla uyarı verme durumu testleri çalışmaz kılıyor. Eğer bir uyarı beklenirken verilmez ise o test başarısız oluyor ve kalan testler çalışmaya devam ediyor. Ama eğer bir uyarı beklenirken iki uyarı verilirse Selenium ilk uyarıyı kapatıyor ve testlere devam etmeye çalışıyor haliyle kapatılamayan diğer uyarı geri kalan tüm testlerin başarısız sonuç değil hata vermesine neden oluyor. Bu sorun çözülene kadar ona ait testi devre dışı bıraktım. Staj raporunu yazarken tekrar kontrol ettim ve sorunun çözüldüğünü gördüm, sorun ben stajdan ayrıldıktan sonra çözülmüş.

Test fonksiyonlarının yazımı bittikten sonra test durumları dokümanını, koddaki test fonksiyonlarındaki numaralarını içerecek tekrar şekilde düzenledim. Böylece test durumları dokümanına bakan kişi testin kodda hangi fonksiyona karşılık geleceğini görebilecek hale geldi. Dokümanda bulunup testlerde eksik kalan ve testlerde bulunup dokümanda eksik kalan yerleri düzelttim.

Kodun gözden geçirilebilmesi için Kartaca takımından Fisheye içinde Mesajlarım Test klasörü açılmasını istedim. Kartaca ekibi klasörün zaten yaratılmış olduğunu söyledi, Fisheye'dan sorumlu kişi ile konu üzerinde konuşurken onun gördüğü dosyaları benim göremediğimi fark ettim. Yetki ile ilgili bir hata olduğunu düşünüp incelemeye başladılar daha sonra Fisheye ayarlarını karıştırırken “show deleted files/hide deleted files” seçeneğinin nedense Mesajlarım klasörünün görünüp görünmemesinde etkili olduğunu fark ettim. Silinmiş dosyalar ile alakalı bir seçeneğin silinmemiş olan bir dosyayı görünürlüğüne etkilediğini Kartaca takımına anlattığımda önce inanamadılar fakat sonra bunun sistemdeki çok acayip bir bug olduğu anlaşıldı. Bu basit olay benim yarım günümü harcadı.

Şirkette testlerden sorumlu personeli gözden geçirici olarak seçip kodu gözden geçirmeye yolladım, daha sonra beni yanına çağırdı ve gözden geçirme üzerine konuştuk. Gözden geçirenin istekleri üzerine;

- Benim yazdığım yardımcı fonksiyonları şirkette kullanılan ortak fonksiyonlarla değiştirmemi istedi. Benim şirkette kullanılan ortak fonksiyonlardan haberim yoktu çünkü ben staja başladıktan sonra geliştirilmeye başlanmış yani ortak fonksiyonlar daha çok yeniydi. Ortak kullanılan fonksiyonları gözden geçirmeden sonra istenen yerlerdekiyle değiştirdim.
- Koddaki fonksiyonları daha önceden test durumları dokümanı içerisinde karşılık gelen yerlere yazmıştım fakat buna ilaveten test durumları dokümanındaki maddeleri de kod içerisinde karşılık gelen yerlere yorum satırı olarak eklemem istendi.



Resim 14 Selenium testleri gözden geçirmedeyken

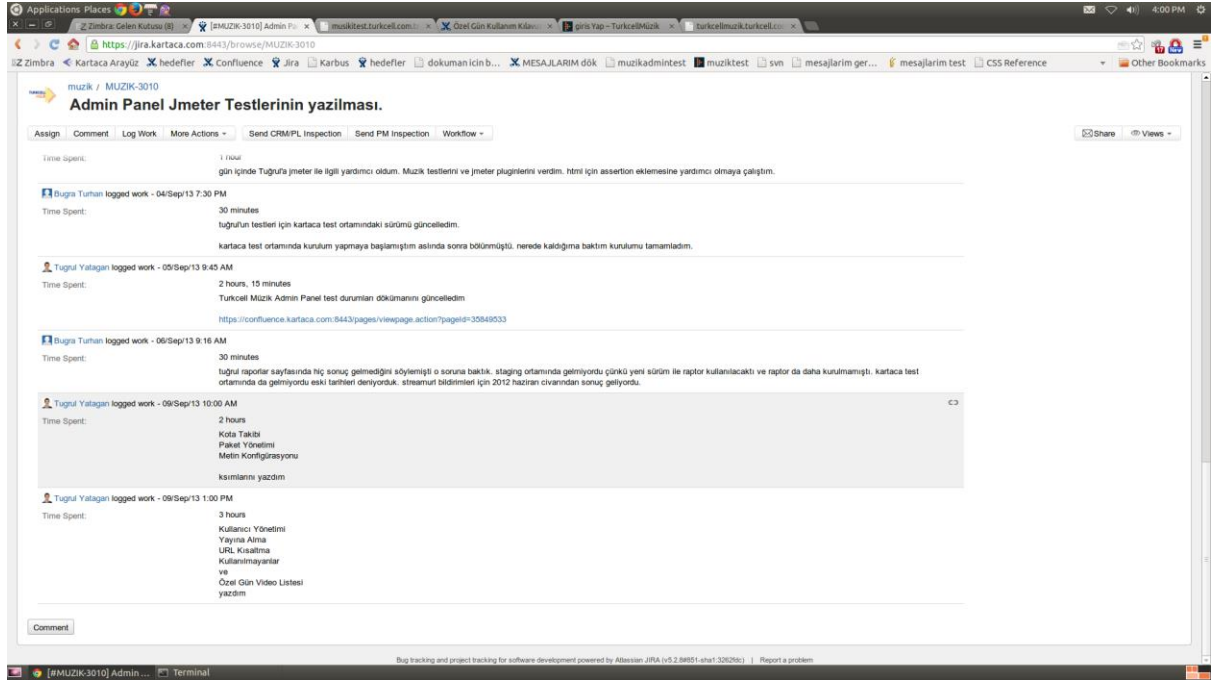
Kodu tekrar gözden geçirmeye yolladım. İkinci gözden geçirmeden sonra istenen ufak tefek değişiklikleri de yaptım. Tekrar gözden geçirmeye gerek olmadığını söylediler. Son kullanıcı servisleri takım toplantısına katıldım. Takımla ilgili şu anki projelerin ilerleyişi ve gelebilecek projeler ile ilgili konuşmalar yapıldı. Yeni projeye (Çalarken Dinlet) başlamak için Turkcell den bütçe onayı beklendiği söylendi. Bu toplantıda da beni ilgilendiren fazla bir konu konuşulmadı.

İşin tamamını kontrol etmesi için takım sorumluma yolladım. Kodların gözden geçirilmesinin tamamlandığını söyledim. Testlerin Bamboo üzerinde çalıştırılabilmesi için Mesajlarım kod tabanına erişimi olan birisinin sürüm oluşturma betiklerini yazması gerektiğini tekrar anlattım. Benim kod tabanına erişimim olmadığı için yazamadığımı söyledim. Selenium testlerini test ortamını temel alarak yazdım. IE8'deki debug mesajlarını uyarı mesajı olarak verdiği için testler IE8 test ortamında çalışmayacağını söyledim. Ayrıca Firefox'taki birden fazla uyarı verme sorunu çözülene kadar 8. test çalıştırılmayacağını söyledim. Sorunun düzeltilmesi için iş açıldı ve daha sonra da düzeltilmiş.

Staj boyunca yapmam beklenen en büyük iş bitti

Turkcell Muzik yönetici panel Apache Jmeter testlerinin geliştirilmesi

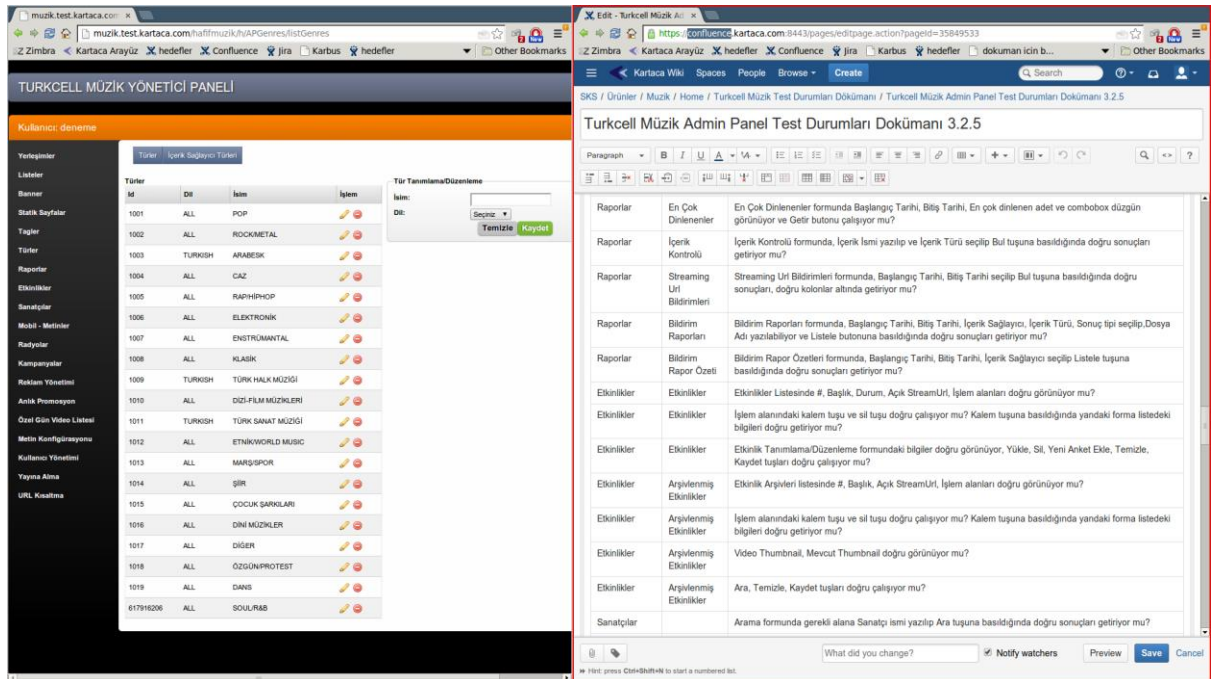
Jmeter ile ilgili dokümanları okudum denemeler yaptım. Takımındaki sistemci bana Jmeter ile yönetici panele isteklerinin nasıl yollanacağını ve gelen cevapların kontrollerinin nasıl yapılacağını gösterdi. Genel olarak testlerin nasıl yazılacağından bahsetti. Eski jmeter testlerini ve Jmeter pluginlerini verdi. Html cevapları için nasıl assertion ekleneceğini gösterdi. Eski regexleri gösterdi, benim güncel olarak regexleri tekrar yazmam gerektiğinden bahsetti.



Resim 15 Turkcell yönetici panel Jmeter testleri Jira iş sayfası

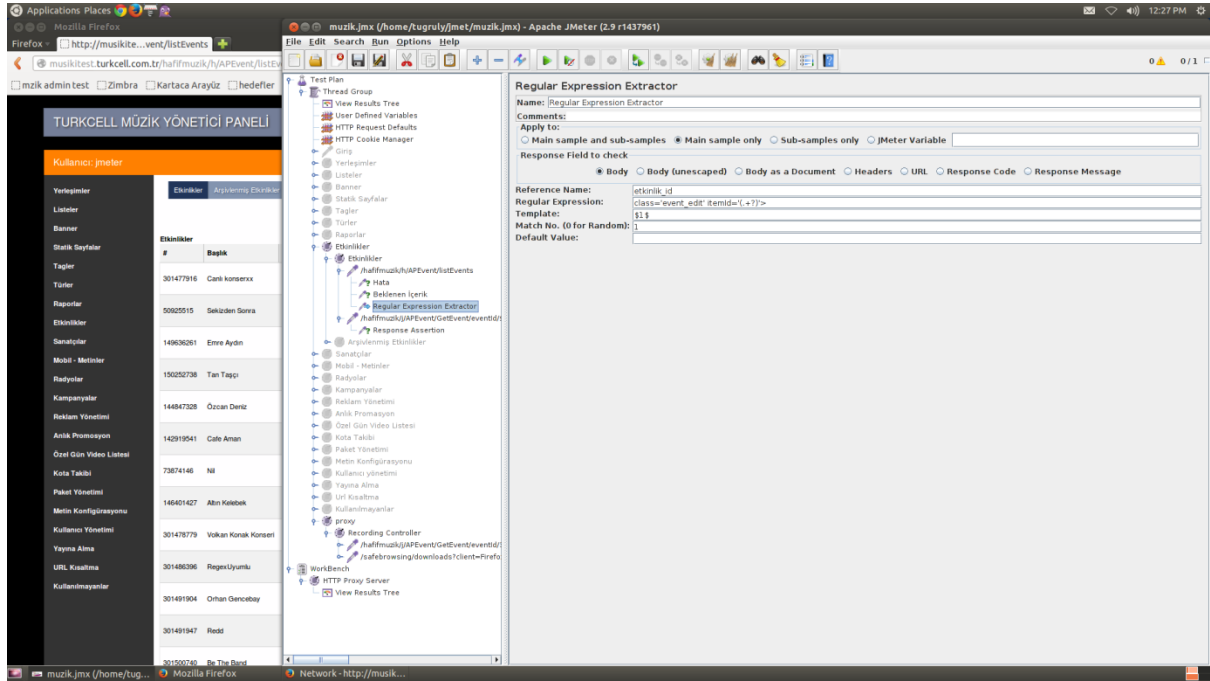
Jmeter testlerini geliştirmeye başlamadan önce, Turkcell Müzik Admin Panel test durumları dokümanını güncelledim.

Her zaman kullanmaya daha alışık olduğum xmonad masaüstü ortamı Jmeter ile düzgün çalışmayınca Ubuntu için varsayılan olan Unity masaüstü ortamına geçtim.



Resim 16 Turkcell Müzik yönetici paneli test durumları dokümanı

Turkcell Müzik yönetici panele girişi yapılırken kullanıcı adı ve parolasının yanında captcha bulunması testlerin otomatik giriş yapmasını engelliyordu. Geliştiriciler ile konu hakkında konuştuğumda, Jmeter testleri için test ortamının belli ip'lerden captcha olmadan geçiş yapmasını sağlamak üzere bir geliştirme yapılmasının en mantıklı çözüm olduğunu söylediler. Konu hakkında bir iş açıldı. Sorun çözümlenene kadar ben her oturum için tarayıcıdan giriş yapıp o oturum için saklanan cookie'yi Jmeter'a elle giriyordum. Cookie'lerin zaman aşımı ne kadar bilmiyordum ama bu işlemi günde bir kere yapmam yeterli oluyordu.



Resim 17 Turkcell Müzik yönetici panel Jmeter testleri

Yönetici panel Jmeter testlerini tamamlamak yaklaşık 1.5 günümü aldı. Selenium testlerine kıyasla çok daha kolay ve hızlı geliştirilebiliyor. Fakat çalışma mekanizmaları oldukça farklı olduğu için birbirlerinin yerini tutamıyorlar. İşi bitirdikten sonra jmx dosyasını SVN'e yükledim ve gözden geçirme için işi takımımdaki sistemciye yolladım. Jmeter koda tabanlı olmadığı için Fisheye benzeri bir gözden geçirme sistemi de yok. Gözden geçirme esnasında takımımdaki sistemci ile konuştuğumda gözümünden kaçan yerler olduğunu fark ettim ve eksik kısımları tamamladım. Tekrar gözden geçirmeye yolladım ve stajım sona erdi.

4. Staj Deneyimiyle İlgili İzlenimler ve Değerlendirmeler

Kartaca'da geçirdiğim staj süresince hayati kısımları olmasa da gerçek projeler üzerinde çalışma fırsatı buldum. Programlama becerilerimi geliştirdim ve daha önceden hiç duymadığım iş yaşamına ait birçok terim ve kavramı öğrendim. İşler bir şirket içerisinde nasıl yapılıyor, projeler nasıl ilerliyor, şirket içerisinde iletişim nasıl sağlanıyor gibi birçok şeyi yerinde öğrenme fırsatı buldum. Gerçek projeler üzerinde çalışırken nelere dikkat etmek gerektiğini ve yapılması gereken işlerin zaman baskısı altında nasıl düzgünce dağıtılarak yapıldığını görme fırsatı buldum. Profesyonel olarak idare edilen bir takım içerisinde çalışmanın nasıl bir şey olduğunu gördüm.

5. Sonuç

Staja başladığım ilk günden itibaren benden istenen işleri stajımın son gününde fazlasıyla tamamlayarak sonuçlandırdım. Bana, birçok deneyim ve bilgi katan bir staj dönemi geçirmiş oldum.

6. Tavsiyeler

Staj yapacak arkadaşlara en büyük tavsiyem staj süresince öğrenebildikleri kadar pratik bilgiyi öğrenmeleri ve şirket içindeki kavramları olabildiğince öğrensinler. Çünkü bir şirket içerisinde kazanılmış bir deneyim var ve dışarıdan ilginç görünen bir işleyişin temelinde mutlaka bir tecrübe bulunuyor. Bunların haricinde staj ciddiyetle yapılır ise önemli kazanımlar elde etmeye yarayan bir araç olduğunu düşünüyorum.