1. KURULUŞ HAKKINDA BİLGİLER

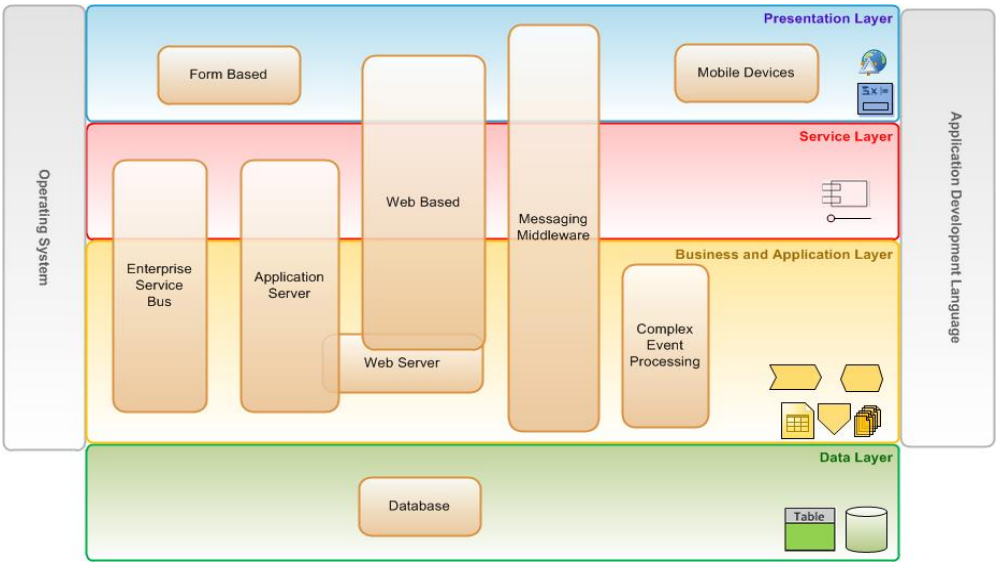
Bir Garanti bankası iştiraki olan Garanti Teknoloji, 1983'te bilgi işlem merkezi olarak faaliyete geçtiği günden beri bankacılık ve finansal hizmet alanlarında faaliyet gösteren şirketlere teknoloji altyapısı, yazılım geliştirme, internet uygulamaları, entegrasyon, sistem ve güvenlik yönetimi, proje yönetimi ve ofis uygulamaları alanlarında çeşitli hizmetler sunmakta ve danışmanlık vermektedir. Bünyesinde 1500'ün üzerinde çalışan bulunduran şirket, Güneşli'deki kampüsünden, Pendik kampüsüne en yakın zamanda geçişini tamamlamaya çalışmaktadır. Çalışanların iş tanımlarının net olması büyük çaplı bir firma için kritik olduğundan, birimler olabildiğince daraltılmış olmakla beraber, 8 departman, 46 birim ve 100'ün üstünde ekip halinde bir organizasyon düzeni bulunur. Stajım süresince içinde bulunduğum Mimari ve Bilişim Teknolojileri Güvenlik Yönetimi biriminde ise, müşteri ihtiyaçları; yeni ürün ve teknolojiler, sektör, rakip araştırmaları ve ürün analizleri yapılarak değerlendirilir, yol haritası oluşturulur, mevcut teknolojik mimari, kapasite ve uygulama yetkinlikleri gözetilerek, iş planları, bilişim teknolojisi projelerine dönüştürülür. Genel güvenlik politikası uluslararası standartlara uygun olarak oluşturulur, tüm projelerin risk değerlendirmeleri yapılır, bileşenleri kontrol edilir, izlenir ve kurulan uyarı mekanizmaları tarafından güvenlik kıstaslarına uygunlukları sürekli olarak sorgulanır.

2. GİRİŞ

Garanti Teknoloji firmasının bünyesi altında 2016-2017 eğitim döneminin sonuna denk gelen kısa dönem yaz stajım, hem öğrenim hayatımda hem de bireysel olarak edindiğim bilgileri pratikte kullanma, yeni teknolojileri kullanmaya adapte olma, bir yazılım geliştirme yaşam döngüsüne dahil olma ve kurumsal çalışma hayatını deneyimleme açısından oldukça geniş perspektifte imkanları tecrübe etmeme yardımcı oldu. Proje müdürlerinin görevleri ve sorumlulukları hakkında fikirler edindim.

3. STAJ PROJESİNİN TANIMI VE ANALİZİ

GT'de projeler Garanti Teknoloji Hizmet Yönetim Sistemi üzerinden yapıldığı için projeye dahil olan kişilerin bu sisteme kayıtlı kişiler olması gerekmekte olduğundan, GT'de devam eden aktif projelerden birinde görev almak yerine, Çözüm mimari ekibindeki yetkili uzmanlar tarafından gidişatı denetlenmek üzere farklı proje ve araştırma görevleri verildi. İlk günler kullanılan teknolojiler(araçlar, çatılar, derleyiciler, diller vs.), mimari katmanlar(servis katmanı, veri katmanı, iş ve uygulama katmanı, sunum katmanı) ve firma içerisindeki yazılım geliştirme hayat döngüsü ve iş akışını kavramaya yönelik incelemeler ile geçti.



Şekil 1- Mimari bileşen katmanları

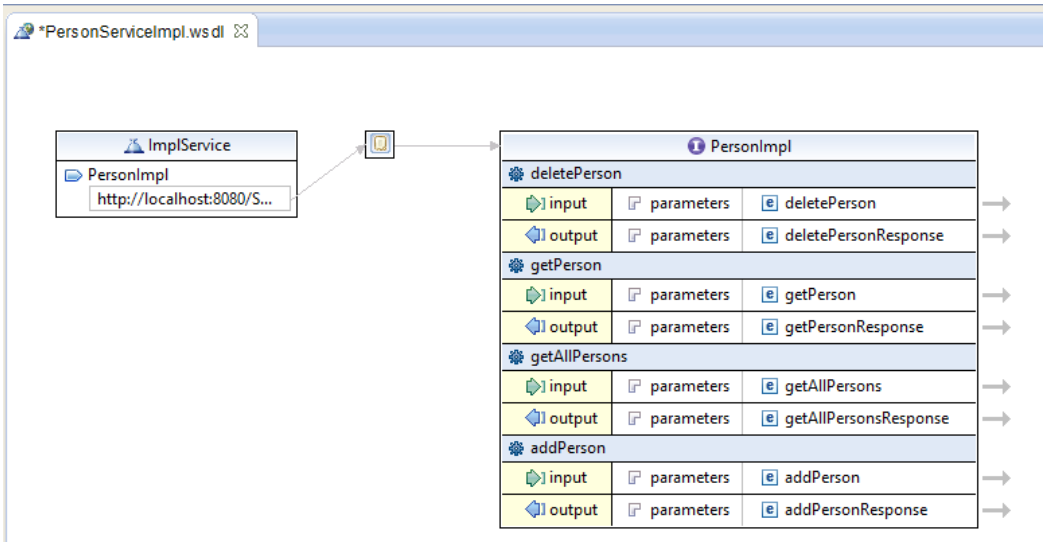
Daha sonrasında ise esas görevim olan SOA(Servis odaklı mimari)'nin araştırılması, Web servisleri geliştirmede kullanılacak bazı çatıların ve platformların incelenmesi, kıyaslanması, ve gerçekleme etabında hangilerinin tercih edilmesi ve neden tercih edilmeleri gerektiğine yönelik fikirleri, bu çatılarla uygulamalar yazarak analiz etmek ile geçti.

3.1 JAX-WS & JAX-RS

Web servisler kısaca, farklı sistemler ve uygulamalar arasında veri alışverişini mümkün kılan standartlar ve protokoller bütünüdür. Farklı dillerde veya platformlarda yazılmış bir yazılım parçası, bu protokoller sayesinde bambaşka bir platformda kullanılabilir. Açılımı "Java API for XML Web Services" olan JAX-WS ise, XML ile konuşabilecek web servisleri ve istemcileri inşa etmeye yarayan bir teknolojidir. Bu teknolojide web servis işlem çağrıları XML tabanlı bir protokol aracılığıyla gösterilir. Bu protokol için en uygun örnek ise SOAP'tır. SOAP spesifikasyonu, içerisinde web servis çağrı ve istemlerini göstermek için zarf yapısı(envelope structure), kodlama/şifreleme kuralları ve konvansiyonlar gibi yapılar tanımlar.

Bu çağrı ve istemler SOAP mesajları halinde HTTP üzerinden iletilir. SOAP mesajları temelinde okumak için zor olsa da, JAX-WS ara yüzü bu zorluğu yazılımcı açısından hafifletir. Sunucu tarafında yazılımcı, metotları tanımlayacağı bir ara yüz(interface) sınıfı yaratır ve bunları gerçekleyecek başka sınıfları da dahil eder. İstemci tarafında ise bir proxy(servisi gösteren bir

yerel obje) oluşturulur ve metotlar bu proxy üzerinden kullanılır. JAX-WS ‘ in yazılımcının yükünü hafiflettiği noktanın ise, bu ara yüz metotlarının isimleri, parametreleri, dönüş türleri gibi bilgilerini XML haline çevirip SOAP mesaj haline getirmesi olduğunu bu standartla çalışırken deneyimledim.



Şekil 2- Yazdığım bir personel kayıt sistemi web servisinin WSDL dosyası şema örneği (WSDL: Web servisinin işleviyle ilgili bilgileri içeren XML tabanlı SOAP mesajının özel isimlendirmesi)

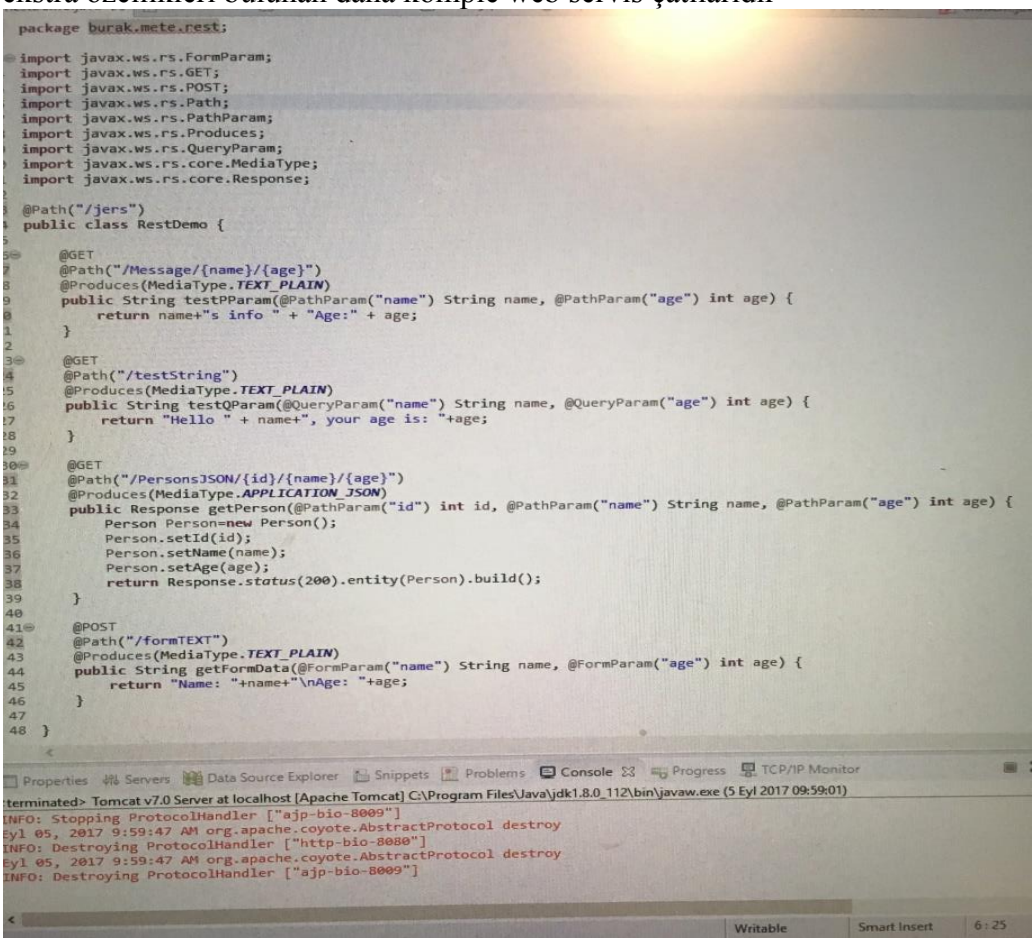
```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://service.jaxws.journaldev.com" xmlns:apachesoap="
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
 <wsdl:types>
  <schema elementFormDefault="qualified" targetNamespace="http://service.jaxws.journaldev.
   <import namespace="http://beans.jaxws.journaldev.com"/>
   <element name="deletePerson">
    <complexType>
     <sequence>
      <element name="id" type="xsd:int"/>
     </sequence>
    </complexType>
   </element>
   <element name="deletePersonResponse">
    <complexType>
     <sequence>
      <element name="deletePersonReturn" type="xsd:boolean"/>
     </sequence>
    </complexType>
   </element>
   <element name="getPerson">
    <complexType>
     <sequence>
      <element name="id" type="xsd:int"/>
```

Şekil 3- (Aynı şemanın XML tabanlı halinden bir kesit)

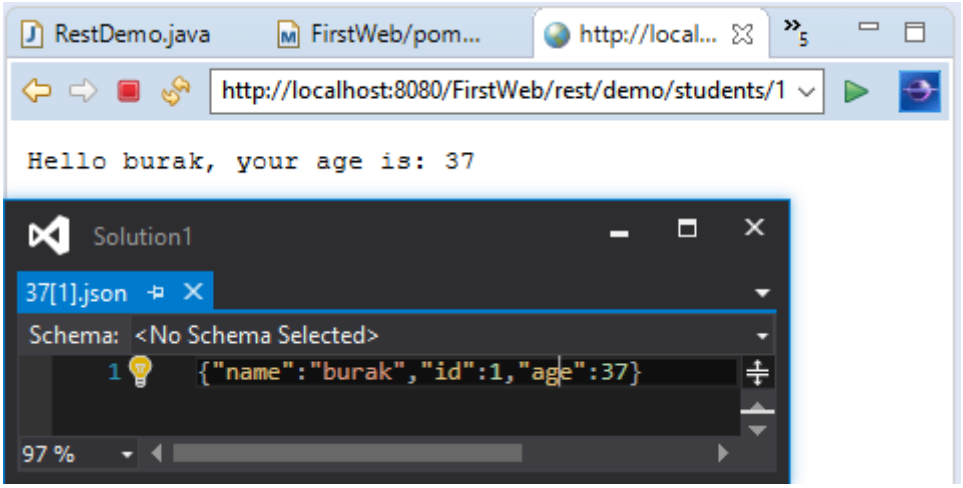
JAX-RS ise, RESTful(Representational State Transfer) web servisleri için geliştirilmiş spesifikasyonlardır. REST ise, web standartları tabanlı, iletişim için HTTP protokolünü kullanan bir mimaridir. Bu mimaride sunucu gerekli kaynakları sağlarken, istemci de URI’da saklanan bu kaynaklara ulaşır ve kullanıcıya sunar. SOAP’tan farklı olarak bu mimaride kaynaklar yalnızca XML formatında değil, XML, JSON veya düz metin olarak gösterilebilir. JSON ise web servisleri için en çok kullanılan formattır.

3.2 Metro & Jersey & Apache Axis2 & Apache CXF

Metro, JAX-WS ara yüz standardını, Jersey ise JAX-RS standardını gerçekleyen(implementasyon) çatılardır. Apache CXF ve Axis2 ise gerçeklemelerin yanı sıra ekstra özellikleri bulunan daha komple web servis çatılarıdır



Şekil 3- Jersey çatısı ile gerçeklediğim, JSON ve metin mesajı döndürebilen bir personel bilgileri uygulaması



Şekil 4-
Jersey ile yazılan RESTful Web servisin GET metotlarıyla döndürdüğü iki farklı veri tipi, düz metin ve JSON.

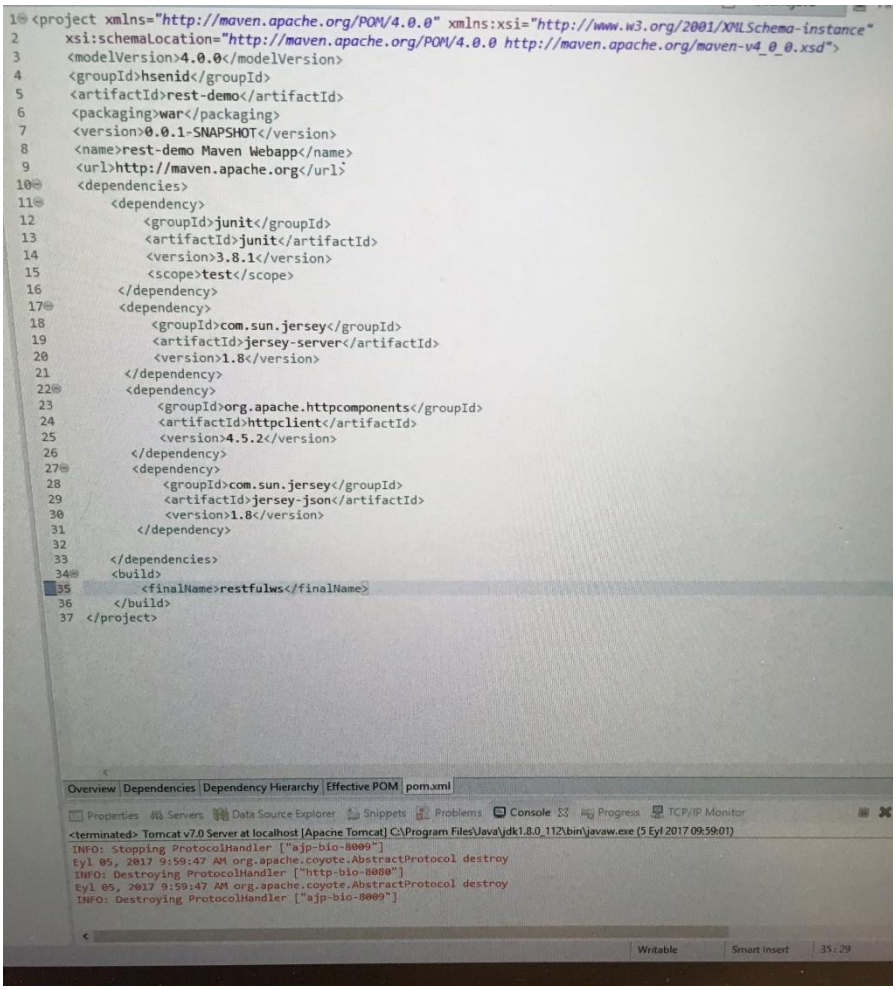
Hem SOAP web servislerde, hem de diğer çatılarda yazdığım web servis örneklerinde kullandığım uygulama sunucusu Apache Tomcat v7.0, kullandığım IDE ise Eclipse Kepler'dir.

Jersey çatısı ile RESTful web servisi geliştirirken karşılaştığım zorluklardan birisi, Maven kullanmaya alışmak oldu. Java Maven bir JDT yani bir Java Geliştirme Aracı'dır. Bize sağladığı kolaylıklardan birisi projemizin bağımlılıklarını (kütüphaneler) efektif bir şekilde yönetebilmek. Bu araç, POM adı verilen Proje nesne modeli isimli xml dosyası sayesinde, projede kullanılacak kütüphaneleri yerel dizinlerde ve kendi çevrimiçi sunucuları içerisinden arayıp bulur ve projeye dahil eder.

Şekil 3'te kullanılan

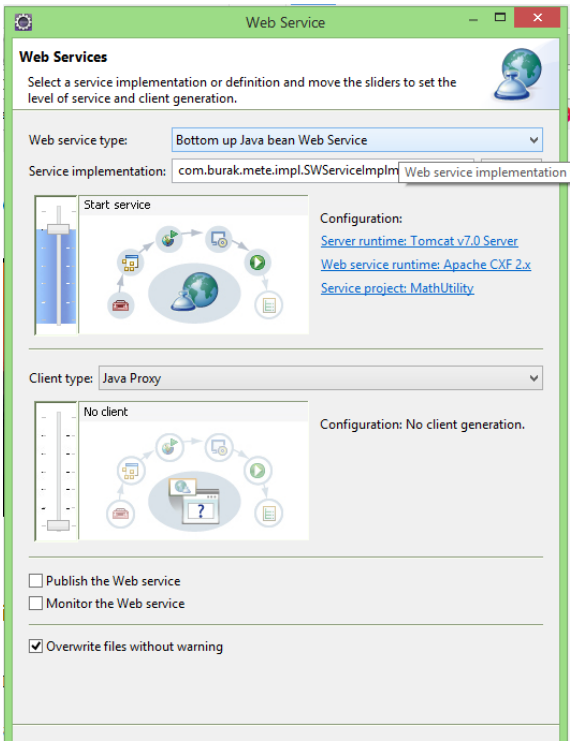
- Javax.ws.rs.FormParam
- Javax.ws.rs.Path
- Javax.ws.rs.QueryParam
- Javax.ws.rs.Produces
- Javax.ws.rs.GET

gibi atıflar (Annotation) Şekil 4'te görülen pom dosyasının içindeki bağımlılıkların projeye dahil edilmesiyle kullanılabilir hale gelmiştir.

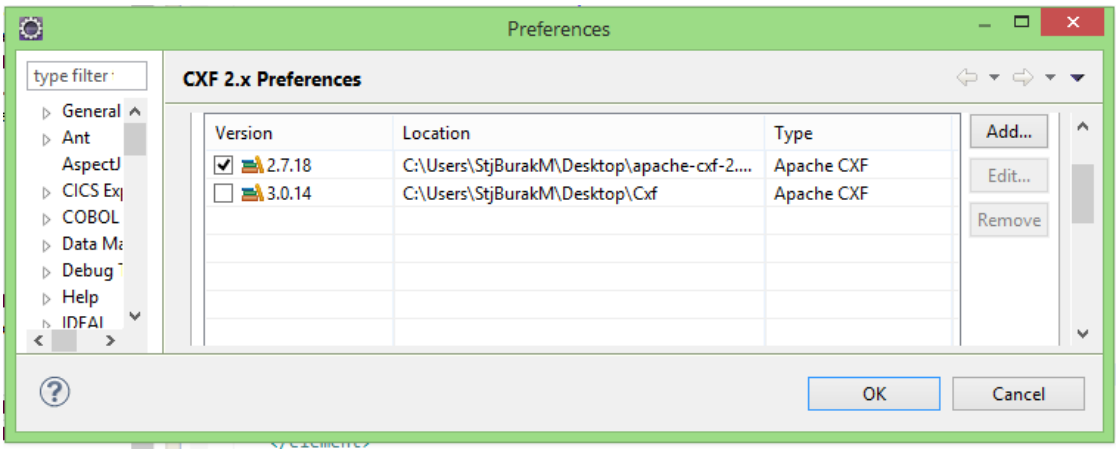


Şekil 5- POM.xml

Bu etapta biraz da Apache Cxf'ten bahsetmek istiyorum. CXF, hem JAX-WS hem de JAX-RS standartlarını destekleyen bir web servis çatısı olarak göze çarpıyor. Bu web servis standartların gerçeklemesinin yanı sıra, WS-Security, WS-Logging gibi WS-* uzantılı bir çok kütüphaneyi içinde bulundurması ile güvenlik ve loglama konusunda yazılımcıya birçok fırsat sunması, ayrıca kurumsal uygulamalar için vazgeçilmez olan ve yazılım geliştiricinin yükünü oldukça hafifleten Spring çatısı ile kolay entegre olması CXF'i oldukça kapsamlı bir çatı haline getiriyor. CXF'in artılarından birisi de WSDL'a bağımlı kalmayıp, WADL (Web Uygulaması Tanım Dili) ve Swagger çatıları ile de kullanılabiliyor olması. Çünkü WSDL'ın, RESTful servislerde sıklıkla kullanılan HTTP metotlarından yalnızca ikisini kullanıyor olması(GET, POST),REST mimarisi ile çalışmasının önünde bir engel.

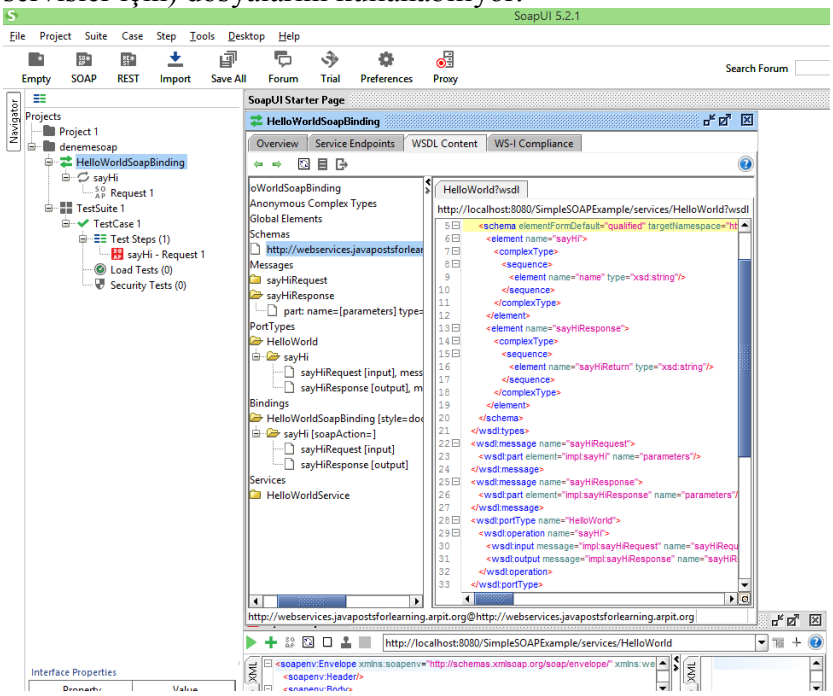


Şekil 6- Axis2 ve CXF web servis tanımlama



Şekil 7- Eclipse CXF konfigürasyonu

Web servis geliştirirken, test ayağında kullandığım bir diğer uygulama ise SoapUI oldu. Uygulama SOAP Web servisleri ile olduğu gibi RESTful web servisleri ile de çalışabiliyor. Açık kaynaklı ve ücretsiz olduğu için kullanımı oldukça yaygın ve arayüz servisleri testleri için kullanılan en popüler uygulama. Fonksiyonel Test(işlev), performans testi, eş güdümlü çalışabilirlik(interoperability) testleri gibi imkanları sunan uygulama, web servislerini simüle etmek için WSDL ve WADL(RESTful servisler için) dosyalarını kullanabiliyor.



Şekil 8- SoapUI arayüzü, ve test edilmek için hazırlanmış basit bir program

3.3 Çatıların kıyaslamasal değerlendirmeleri

AXİS 2

Temel Artıları:

- Çoklu dil desteği (Java, C++)
- Yukarıdan aşağıya ve aşağıdan yukarıya yönelimlerinin(Contract First-Contract Last) ikisini de desteklemesi(Contract Last yönteminde önce Java kodu üretilip daha sonra WSDL oluşturulurken, Contract-First methodunda ise önce WSDL yaratılır ve java kodu buna göre düzenlenir)
- WS-* spesifikasyonlarından birçoğunu destekler

Temel Eksileri:

- CXF'e oranla daha uzun kod gereksinimi duyar
- JAX-WS & JAX-RS ile yüzde yüz uyuşma yoktur

CXF

Temel Artıları:

- En yaygın olarak kullanılan Web Servis çatısıdır.
- Daha az kod gerektirir ve kullanımı daha kolaydır.
- JAX-WS ve JAX-RS ile tamamen uyuşur (fully compliant)
- Tüm web servis çatıları arasında en yüksek performansa sahip
- Çeşitli ön-yüz modellerini destekler
- Spring Çatısı ile uyumludur

Temel Eksileri:

- WSDL 2.0 ile uyumlu değildir

4. SONUÇ

Garanti Teknoloji' de gerçekleştirdiğim 20 günlük stajımın ardından, web servislerinin işlevleri, istemci ve sunucu tarafında yazılım geliştirme hakkında birinci elden tecrübeler edinmiş oldum. Bir firmanın sistem ekibinde çalışmak, mimari dizayn hakkında bilgiler edinmenin benim için oldukça kıymetli olduğunu düşünüyorum. Projemin doğrudan amaçlarından olmadığından raporda bahsetmediğim, z/OS işletim sistemi ve mainframe üzerinde kullanılan COBOL programları, ESB(Kurumsal Veri Yolu) ve mikroservisler hakkında da bilgi sahibi olmuş oldum.Analiz, araştırma ve yazılım geliştirme sürecinde kendime önemli edinimler kazandırdığımı düşünüyorum. Teknik kazanımlarımın yanısıra ilk staj deneyimimin büyük bir kurumsal firmada geçmesinin de oldukça değerli olduğunu düşünüyorum. Bu sayede firma içi dinamikler, gerektiği anlarda farklı ekipler ile entegre olup çalışma, toplantı süreçleri, insan yönetimi gibi iş hayatımın gelecekte yadsınamayacak bir bölümünü oluşturacağına inandığım detayları da gözlemlemiş, gerektiği yerlede sorular sorarak öğrenmiş bulunmaktayım.

Son olarak İstanbul Teknik Üniversi'nin gelecek dönemlerde staj yapacak öğrencilerine, büyük bir firma olmasına rağmen stajyerlerine ve onların gelişimlerine verdikleri önem ve hem teknik hem de sosyal olarak kişiye önemli tecrübeler kazandırdığını gördüğüm Garanti Teknoloji'yi seçmelerini tavsiye ederim.

5. REFERANSLAR

https://www.dzone.com/ (Ankur Kumar, Mayıs 2013)
https://www.soapui.org/
http://cxf.apache.org/

6. EKLER

Criteria	Weight	AXIS2	CXF
Development Features	10.0%	2.6	2.8
Security	15.0%	2.3	2.7
Performance	20.0%	2	3
Reusability	5.0%	3	3
Data Bindings Support	5.0%	3	3
TCO (Total Cost of Ownership)	10.0%	3	2
Web Services Standards Support	10.0%	2.85	2.39
Transport Mechanism Support	5.0%	3	3
Vendor Alignment	5.0%	3	3
JSR Standards Alignment	5.0%	3	2
Required Skill sets	5.0%	2	2
Governance	5.0%	3	2

3.3 ***Çatıların kıyaslamasal değerlendirmeleri (www.dzone.com , Mart 2013)***