



YILDIZ TEKNİK ÜNİVERSİTESİ STAJ DEFTERİ

ÖĞRENCİNİN

Bölümü: Matematik Mühendisliği

Adı Soyadı: Fethi Fırat TÜLÜ

No: 13052068

İÇİNDEKİLER

1. HAFTALIK ÇALIŞMA ÇİZELGESİ

2. GİRİŞ

3. KURUM TANITIMI

4. BİRİM TANITIMI

5. HAFTALIK ÇALIŞMA RAPORU

6. PROJE

6.1.1. PROJENİN TANITIMI

6.1.2. PROJENİN PLANLANMASI

6.1.3. PROJE AYRINTILARI

6.1.4. PROJENİN EKKRAN ÇIKTILARI

6.1.5. PROJENİN DEĞERLENDİRİLMESİ

7. STAJ DEĞERLENDİRİLMESİ

8. KAYNAKLAR

1. HAFTALIK ÇALIŞMA ÇİZELGESİ

30/07/2018 TARİHİNDEN 03/08/2018 TARİHİNE KADAR OLAN HAFTALIK ÇALIŞMA			
GÜN	YAPILAN İŞLER	Sayfa No	Çalışılan Saat
PAZARTESİ	ORYANTASYON, BİLGİ VE İLETİŞİM TEKNOLOJİLERİ DEPARTMANI İLE TANIŞMA	8	8
SALI	KURULUM VE UYGULAMALARI ÖĞRENME	8	8
ÇARŞAMBA	UYGULAMALAR VE ACTIVE DIRECTORY	9-15	8
PERŞEMBE	ANALİSTLERLE TANIŞMA	16-17	8
CUMA	YAZILIM GELİŞTİRME MODELLERİ ARAŞTIRMASI	18-23	8
Çalıştığı işyeri ve kısım: Kontrol edenin unvanı: Adı ve Soyadı: İmzası:			
06/08/2018 TARİHİNDEN 10/08/2018 TARİHİNE KADAR OLAN HAFTALIK ÇALIŞMA			
GÜN	YAPILAN İŞLER	Sayfa No	Çalışılan Saat
PAZARTESİ	MICROSOFT.NET ARAŞTIRMASI	24-26	8
SALI	ASP.NET ARAŞTIRMASI	27-28	8
ÇARŞAMBA	C# YAPISI ARAŞTIRMASI	29-31	8
PERŞEMBE	SQL ARAŞTIRMASI	32-38	8
CUMA	SQL ARAŞTIRMASI	32-38	8
Çalıştığı işyeri ve kısım: Kontrol edenin unvanı: Adı ve Soyadı: İmzası:			
LeasePlan Türkiye – Bilgi ve İletişim Departmanı Bilgi ve İletişim Departmanı Müdürü Esra YILMAZ			

13/08/2018 TARİHİNDEN 17/08/2018 TARİHİNE KADAR OLAN HAFTALIK ÇALIŞMA			
GÜN	YAPILAN İŞLER	Sayfa No	Çalışılan Saat
PAZARTESİ	ERİŞİM VE MS VISUAL STUDIO YAPISI	39-41	8
SALI	TFS KURULUMU VE FLEETNET	42-46	8
ÇARŞAMBA	PROJE KARARLAŞTIRILMASI	47-48	8
PERŞEMBE	PROJE PLANLAMASI	49-51	8
CUMA	PROJE		8
Çalıştığı işyeri ve kısım:	LeasePlan Türkiye – Bilgi ve İletişim Departmanı		
Kontrol edenin unvanı:	Bilgi ve İletişim Departmanı Müdürü		
Adı ve Soyadı:	Esra YILMAZ		
İmzası:			
27/08/2018 TARİHİNDEN 31/08/2018 TARİHİNE KADAR OLAN HAFTALIK ÇALIŞMA			
GÜN	YAPILAN İŞLER	Sayfa No	Çalışılan Saat
PAZARTESİ	PROJE		8
SALI	PROJE		8
ÇARŞAMBA	PROJE		8
PERŞEMBE	30 AĞUSTOS ZAFER BAYRAMI		
CUMA	PROJE		8
Çalıştığı işyeri ve kısım:	LeasePlan Türkiye – Bilgi ve İletişim Departmanı		
Kontrol edenin unvanı:	Bilgi ve İletişim Departmanı Müdürü		
Adı ve Soyadı:	Esra YILMAZ		
İmzası:			
03/09/2018 TARİHİNDE Kİ GÜNLÜK ÇALIŞMA			
GÜN	YAPILAN İŞLER	Sayfa No	Çalışılan Saat
PAZARTESİ	LEASEPLAN'A VEDA		8

2. GİRİŞ

LeasePlan, otomobil kiralama ve filo yönetiminde uzmanlaşmış uluslararası bir Hollanda kökenli şirkettir. Ürünleri operasyonel filo yönetim hizmetlerinden oluşmaktadır.

Staj yaptığım departman, şirketin verilerini ve bilgisini saklayan, işleyen ve ayarlayan bölümdür.

Günümüzde bilgi işlem daha çok bir kurumun teknik altyapısını sağlayan bölüm olarak düşünülse de, genel olarak üstlendiği roller bunun ötesindedir. Bilgi işlem, sağladığı teknik altyapı ile tüm verilerin saklanması, işlenmesinde, bunlardan istatistik ve raporların çıkarılmasında ve talepler doğrultusunda yazılım kullanarak yeni ürünler üretmede ya da geliştirmede görev alır.

LeasePlan Bilgi ve İletişim Teknolojileri Departmanı; gelişen bilgi teknolojilerini kullanıma sunarak, şirket çalışanları ile müşterilerin gereksinim duyacağı bilişim hizmetleri ve servislerinin karşılanmasında, ihtiyaç duyulan yazılımsal ve donanımsal çözümleri üretmek, temin etmek ve bu amaçla yeni bilişim teknolojilerini şirkete kazandırarak etkin ve verimli bir şekilde kullanımını ve devamlılığını sağlamayı hedeflemektedir.

Çalışacağım ekibi incelerken, yazılımın etkili bir iş bölümü ve entegrasyon gerektirdiğine kanaat getirdim. Gözlemlediğim kadarıyla ofis içerisindeki insanlar oldukça yoğun ve bir arada bir çalışma yürütmekteler.

Stajımı yaparken benimle ilgilenen Bilgi ve İletişim Teknolojileri Direktörü Esra Hanım'ın katkılarıyla stajımı daha verimli hale getirmeye çalıştım. Öğrendiğim bilgileri uygulayıp sonuçlarını görmeyi ve hatalara çözüm bulmaya çalışmayı bu perspektifte kendimi geliştirmeyi öğrenme yolunda bir adım attığımı düşünüyorum.

3. KURUM TANITIMI



LeasePlan, otomobil kiralama ve filo yönetiminde uzmanlaşmış uluslararası bir Hollanda kökenli şirkettir. 50 yılı aşkın deneyimiyle bugün dünyanın en büyük filo kiralama şirketi olarak 5 kıtada ve 32 ülkede 1,7 milyon adetlik dev bir araç filosunu yönetmekte.



Şirketin ürünleri operasyonel filo yönetim hizmetlerinden oluşmaktaymış ve 1963 yılında kuruluşundan bu yana şirket filo yönetiminin dünya lideri konumuna yükselmiş. Çalışanlarının yaklaşık % 85'inden fazlası Hollanda dışında olan şirkette yaklaşık 7400 kişi çalışıyormuş.

Bir banka ve sürücülere hizmet veren bir şirket arasındaki ortak girişimden sonra 1963'te Hollanda'da kurulan LeasePlan, başlangıçta açık-kitap yönetim modelinde (gerçek bir maliyet yönetim sistemi) uzmanlaşmış. LeasePlan, 1970'lerde Belçika, Almanya, Fransa ve Büyük Britanya'da operasyonlar kurarak ve otomotiv filo yönetiminde dünya lideri konumunu elde ederek uluslararası alanda genişlemiştir.

1990'larda, bankacılık grubu ABN-Amro, LeasePlan'daki hisselerin %100'ünü satın almış ve ABN-Amro Lease Holding adında bir holding şirketi kurmuş. LeasePlan, bir online filo yönetimi yazılım paketi (Plan8, şimdi FleetReporting ya da "FleetNet" olarak adlandırılan) sunarak tekliflerini çeşitlendirmiş. 21 Mart 2016 tarihinde

Hollandalı Stichting Pensioenfonds Zorg en Welzijn emeklilik fonu liderliğindeki uluslararası bir konsorsiyum şirketin devralınmasını tamamlamış. En son Şubat 2018'de şirket, 5 milyar sterlinlik ilk halka arz planlarını açıklamış.

LeasePlan, benzersiz bir iş profili ve güçlü bir sicile sahip. Okunduğu üzere kurulduğu günden bugüne küresel, yenilikçi ve girişimci bir şirket haline gelmiş. Global franchise olarak global filo yönetimi ve sürücü hareketliliği hizmetleri diğer ülkelerde olduğu gibi bizim ülkemizde de sunulmakta.

Bu büyük küresel şirketin ülkemizdeki kronolojik tarihi aşağıda ki gibidir.

LeasePlan Türkiye



Bu arada LeasePlan'ın Türkiye'de ve diğer ülkelerde sunduğu hizmet olan operasyonel araç kiralama ne anlama gelir kısaca ondan da bahsetmek istiyorum.

Operasyonel araç kiralama işletmelerin araç ihtiyaçlarının belirlenen süre ve hizmet kapsamında işletmelere kiralanmasıdır. Bu süreçte yaşanan araçla ilgili kaza, hırsızlık gibi maddi risklere ek olarak yatırım riski de LeasePlan'e aittir.

4. BİRİM TANITIMI

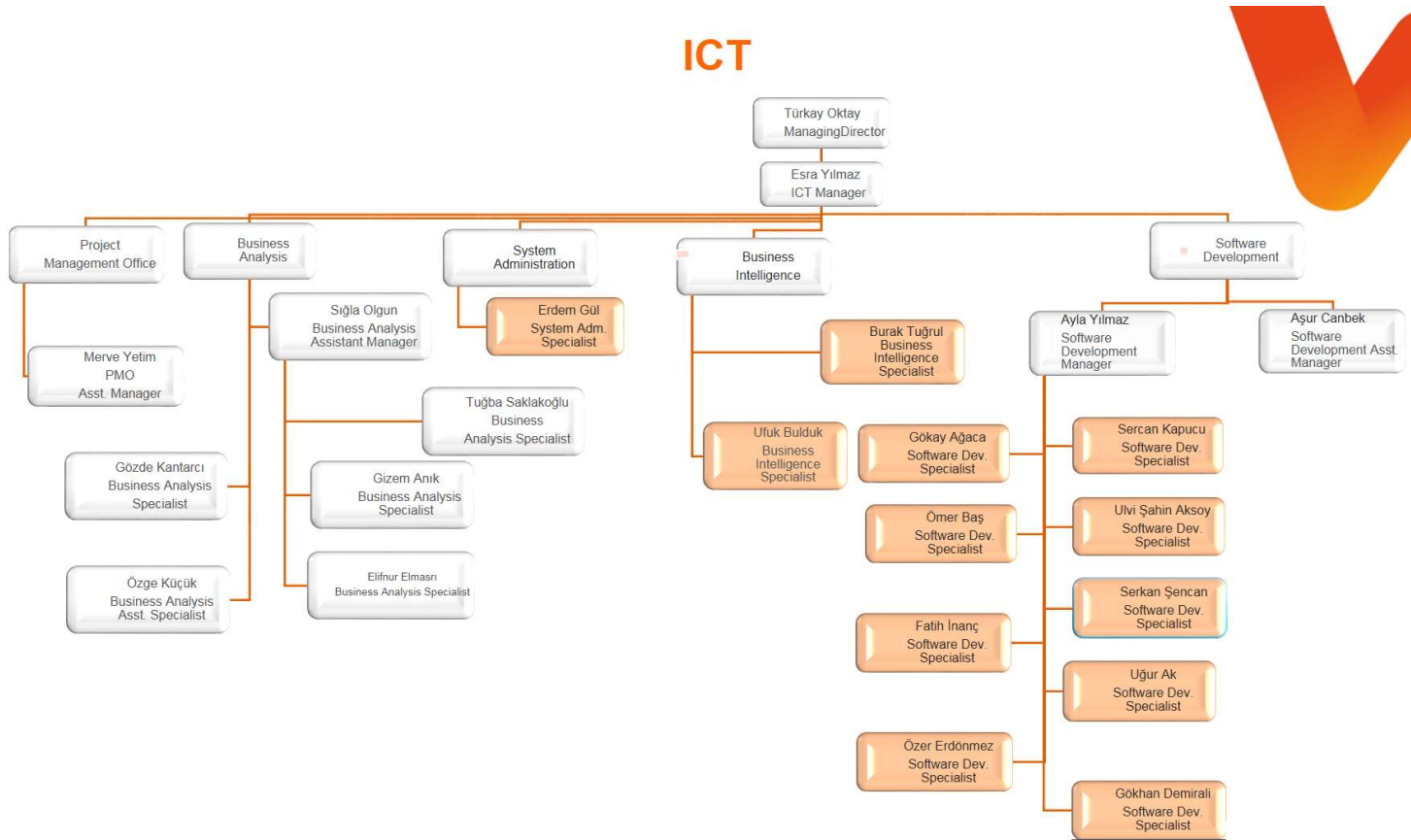
Şirket içindeki birimler;

- Satış ve Pazarlama
- Finans

- Operasyon
- Bilgi ve İletişim Teknolojileri
- İnsan Kaynakları

Bilgi ve İletişim Teknolojileri Departmanı Direktör Esra Yılmaz'a bağlı beş birimden oluşmaktadır;

- Proje Yönetimi Ofisi
- İş Analizi
- Sistem Yönetimi
- İş Zekası
- Yazılım Geliştirme
- Bilgi ve İletişim Teknolojileri Departmanı'nın Organizasyon Şeması;



İş Zekası:

İş Zekası Birimi, Bilgi ve İletişim Teknolojileri Direktörü Esra Yılmaz yönetimindedir. İş Zekası Birimi, tüm LeasePlan Türkiye çalışanlarının raporlama ihtiyaçlarının çözülmesinden sorumludur.

İş Zekası Birimi tarafından oluşturulan iç raporlama faaliyetleri, SAP Business Objects ve Qlikview üzerinden yapılmakta olup; müşterilere açılan raporlar, FleetReporting üzerinde hazırlanmaktadır.

Ayrıca LeasePlan Corporation tarafından yürütülen veri projelerinden de İş Zekası Birimi sorumludur.

Proje Yönetimi Ofisi, İş Analizi ve Sistem Yönetimi;

Proje Yönetim Ofisi, İş Analizi ve Sistem Yönetimi Birimi, Esra Yılmaz'ın liderliğinde olup, Proje Yönetimi Ofisi, İş Analizi ve Sistem Yönetimi olmak üzere, üç ayrı ekipten meydana gelmektedir.

Proje Yönetimi Ofisi'nin temel görevleri aşağıdaki gibidir;

- Projelerin portföyler bazında konsolide edilmesi, senelik "Master Plan" yapılması ile ilgili karar destek sağlaması
- Senelik planda yer alan projeler için statü takibinin, durum raporlamasının, risk ve açık konuların yönetimi
- İşgücü kaynak ve kapasite planlaması ve takibi
- Proje önceliklendirme komitesi için hazırlık ve raporlama
- Tüm proje hedeflerinde (zaman, efor, kapsam, ek bütçe, ek kaynak vb.) değişikliklerin kayıt altına alınması ve proje paydaşlarının bilgilendirilmesi
- Projelerinin maliyet çalışmasının yapılması
- Projelerin finansal getirilerinin hesaplanması
- İhtiyaç duyulduğunda planların revize edilmesi, önceliklendirmenin değiştirilmesi
- Projelerin başarısının değerlendirilmesi ve buna yönelik iyileştirme çalışmaları

İş Analizi Ekibinin temel görevleri arasında, Proje Önceliklendirme Komitesi (PPC) tarafından önceliklendirilen projeler ile ilgili gereksinimlerin ortaya çıkarılması, analiz edilmesi, çözüm kapsamının oluşturulması, yeni süreç / sistem tasarımının önerilmesi, proje paydaşları ile ilgili iletişim ve koordinasyonun sağlanması, projelerin yazılım geliştirme aşaması sonrası ortaya konulan çözümün belirlenen gereksinimleri karşıladığının doğrulanması amacıyla test faaliyetleri koordinasyonunun yapılması, mevcut sistem uygulamalarında meydana gelebilecek sorunların incelenmesi ve çözümler ile ilgili yazılım ekipleriyle koordineli bir şekilde çalışılması faaliyetleri bulunmaktadır.

Tüm bu görevleri gerçekleştirmenin yanı sıra "sürekli iyileştirme" yaklaşımını benimseyerek mevcut süreçler ve sistemlerde gelişim ve iyileştirme imkanlarını araştırıp, LeasePlan global stratejilerine katkı sağlamayı amaçlamaktadırlar.

Yazılım Geliştirme:

Yazılım Geliştirme Birimi, bilgi ve İletişim Teknolojileri Direktörü Esra Yılmaz yönetiminde, Ayla Yılmaz ve Aşur Canbek önderliğindedir.

LeasePlan çalışanlarının kullandığı FleetNet yönetimi ve geliştirmelerini, Microsoft Visual Studio programında Microsoft.NET ASP.NET ve C# programlama dilleri kullanarak yapıyorlar. Sadece FleetNet değil, "Tıkla Kirala", "İş Ortağım" vs. projelerin devamını sağlamaktadırlar. MS SQL Server da aynı zamanda kullandıkları uygulamalardan biridir.

Bu birimin amacı şirketin teknolojik alt yapısını üst seviyelere taşıırken aynı zamanda günlük hayatımızda kullandığımız sistemleri etkileyecek alt yapının da sorunsuz bir şekilde çalışıyor olmasını sağlamaktır.

5. HAFTALIK ÇALIŞMA RAPORU

30.07.2018

LeasePlan’de insanlar en büyük önceliktir. Bu nedenle her yıl çalışan memnuniyetini ölçümlerler ve LeasePlan’in herkesin çalışmak isteyeceği bir marka olması için neleri daha iyi yapabileceğini düşünürler. Stajımın ilk günü, kapıdan içeri girdiğim andan itibaren şirketin bu önceliğini hissettim. İnsan Kaynakları departmanı benimle birlikte başlayan diğer stajyerleri bir toplantı odasında toplayarak şirket hakkında genel bilgi verdi.

Kurum tanıtımında da belirttiğim gibi birçok departmana sahip olan şirketin, her departmana stajyer aldığı katıldığım toplu mülakatta farkına varmıştım. Stajyer olmaya hak kazanan diğer arkadaşlarla beraber tanışıp kaynaşarak o günü birlikte geçireceğimizi düşünüyorduk. Ancak böyle olmadı.

Şirket hakkında bilgi verildikten, tüm sunumlar bittikten sonra hepimiz tek tek şirketi gezerek oryantasyonun son aşamasına doğru yola çıktık. Her departmanın ne iş yaptığını yerinde gözlemledikten sonra her stajyer departmanlarına dağıldı.

Kendi departmanım olan Bilgi ve İletişim Teknolojileri Direktörü Esra Yılmaz ile tanıştıktan sonra, direktörümüz sistem yönetiminde tek başına çalışan Erdem Gül Bey ile beni tanıştırdı. Erdem Bey ile kısa bir tanışma konuşmasından sonra onunla birlikte neler yapacağımı, ona nasıl yardım edeceğimi anlattı. Gün sonunda ise Erdem Bey ile beraber departmanda ki geri kalan herkesle tek tek tanışıp ilerleyen günlerde kiminle neler yapacağımızın konuşmasını gerçekleştirdik. Şanslıydım ki tek bir mentör değil, tüm departmanda ki her meslekten çalışanlar mentörüm olacaktı.

31.07.2018

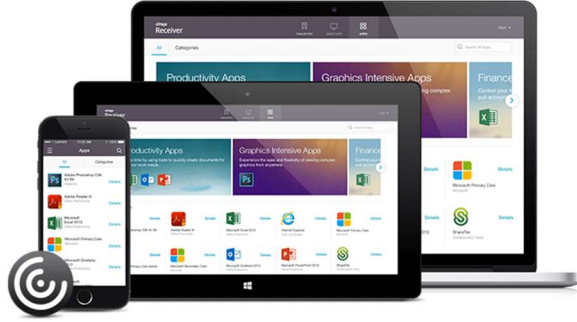
İlk günün ardından ikinci günün sabahı şirketten kendime ait bir bilgisayar, kullanıcı adı ve e-posta hesabımı Erdem Bey’den teslim aldım. Erdem Bey’in gösterdiği ve yapmamı istediği şeylerin başında tüm şirkette ki bilgisayarların Windows 10 işletim sistemine geçmesinde yardımcı olmamdı. İşletim sistemini değiştirdikten sonra kullanıcıların yetkilendirilmesi, kullanıcı kolaylığı ve güvenliği sağlayan uygulamaların yüklenmesi de yapmamı istediği işlerden bir kaçıydı. Bilgisayarlara Windows 10 kurulumu, kullanılacak uygulamaların yüklenmesi, kullanıcı yetkilendirme, Active Directory kullanımı, LanDesk Service Desk kullanımı hakkında detaylı bir bilgi aldım.

Bilgisayarlarda kullanılan Uygulamalar;

1. Citrix Receiver 4.11

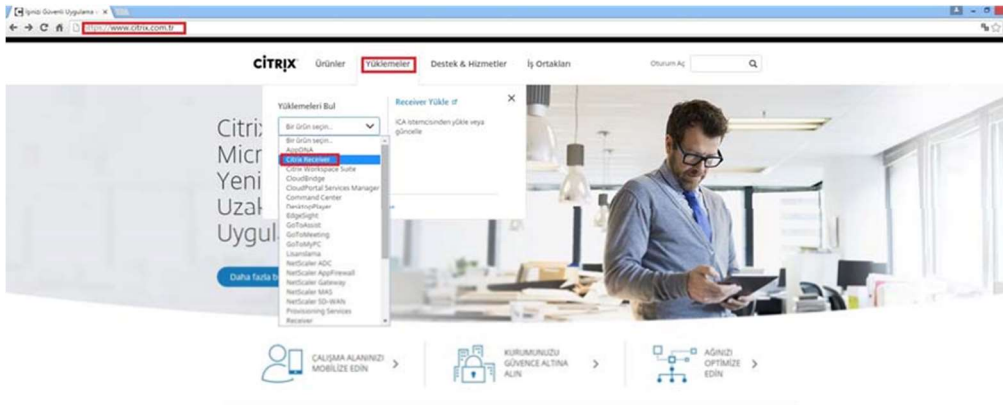
Citrix Receiver, XenDesktop ve XenApp kurulumlarınıza erişim sağlayan kurulumu kolay bir istemci yazılımıdır. Bu ücretsiz indirme ile uygulamalara, masaüstlerine ve verilere akıllı telefonlar, tabletler, PC'ler ve Mac'ler dahil tüm cihazlardan kolaylıkla ve güvenli bir şekilde erişebilirsiniz.

LeasePlan Türkiye şirketi aslında internet ve ağ üzerinden Hollanda'ya bağlı Remote şeklinde çalışmakta. Kişisel bilgisayarlar mevcut fakat her kullanıcı sistemine internet üzerinden bağlanıyor. Bu program sayesinde kullanıcılar diledikleri yerde ve diledikleri zaman hesaplarına remote olarak erişebiliyorlar.

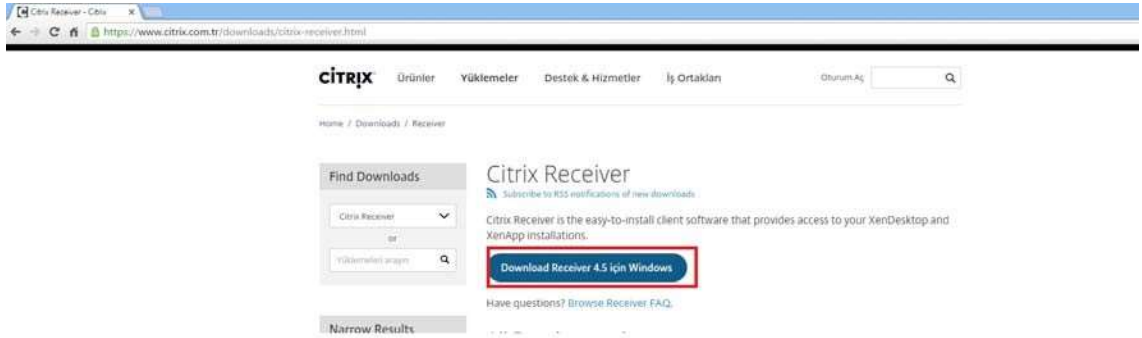


Citrix Receiver Kurulumu

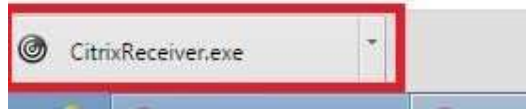
Öncelikle kullandığımız web tarayıcı üzerinden <https://www.citrix.com.tr/> adresine bağlanıp **Yüklemeler** sekmesinden **Citrix Receiver** seçilir.



Açılan sayfa kullandığımız işletim sistemine uygun versiyonu otomatik olarak bulup download etmemiz gereken linki gösterecektir, alttaki resimde gösterildiği gibi **download** sekmesine tıklayıp programı indirmeye başlayabiliriz



Kısa bir süre bekleyip programın indirme işlemi tamamlandıktan sonra **CitrixReceiver.exe** dosyasına çift tıklayıp kurulum işlemi başlatabiliriz.



Açılan kurulum penceresinde ilk öncelikle **Start** butonuna basıp kurulumu başlatırız



Karşımıza çıkan Lisans sözleşmesini onaylayıp, Next butonuyla işlemi devam ettiriyoruz.



Karşımıza çıkan pencerede **Install** sekmesine basıp işlemi devam ettiriyoruz



Kısa bir süre kurulumu bekledikten sonra altta resimde görüldüğü gibi kurulumun başarıyla tamamlandığına dair pencereyi göreceksiniz. **Finish** butonuna basıp hiçbir işlem yapmadan programı kapatabiliriz.



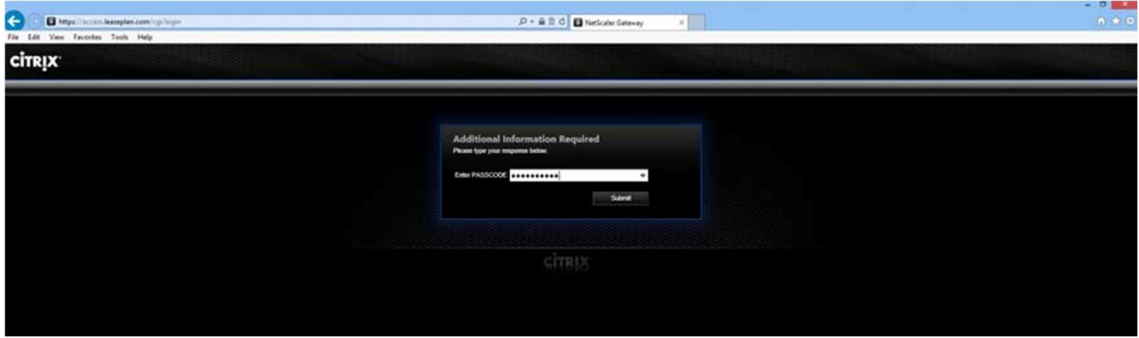
Citrix Receiver program kurulumunu tamamlandıktan sonra Internet Explorer üzerinden <https://access.leaseplan.com/> adresine girilir;

Açılan sayfada user name kısmına sisteme giriş yaptığımız

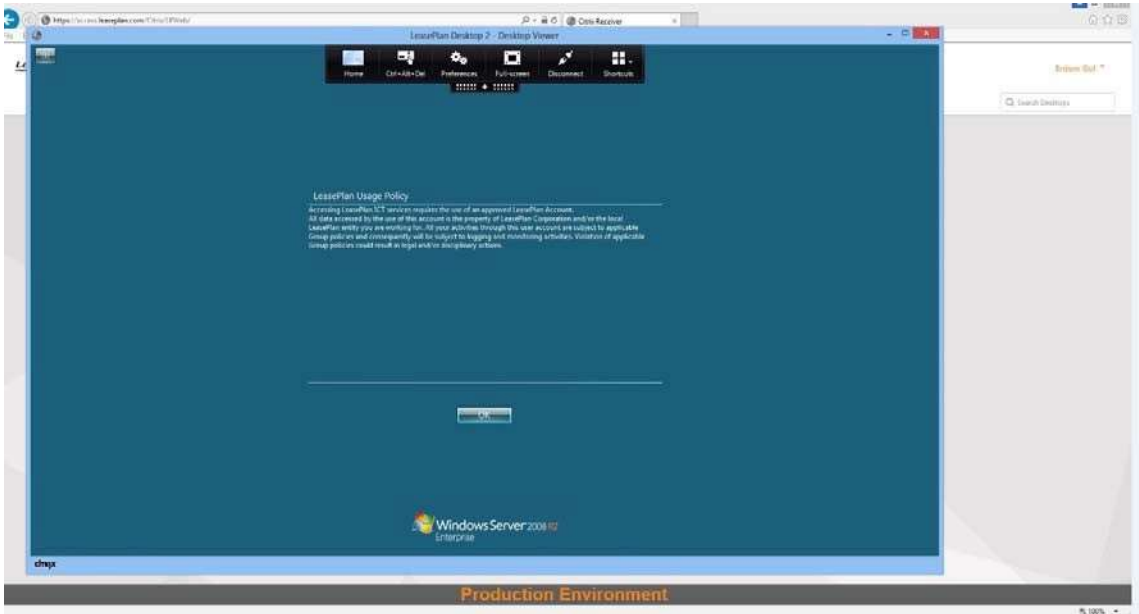
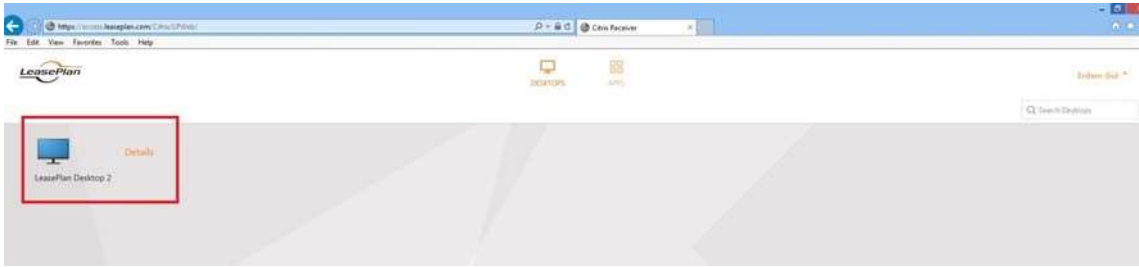
“username(kısa kullanıcı ismimiz)@emea.leaseplancorp.net” eklenerek sisteme bağlandığımız şifre girilir.



Açılan pencerede SMS Passcode Guide dökümanını kullanarak daha önce oluşturduğumuz 4 haneli PIN numarası ve SMS ile gelen şifre birleşik yazılarak girilir. Girilen şifreyi onayladıktan sonra Submit butonuna basılır.



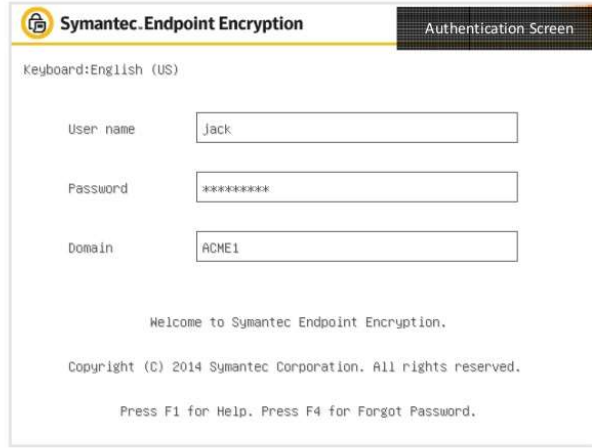
Açılan pencerede **LeasePlan Desktop 2** kısmına tıklayıp sisteme giriş yapabiliriz.



2. Symantec Endpoint Encryption Client

Kurum içindeki sistemlerde bulunan kritik verinin güvenliği için çeşitli önlemler alınıyor. LeasePlan Türkiye şirket çalışanları çoğu zaman remote ve mobil olarak çalışmaktadır. Dizüstü bilgisayarların yoğun iş temposunda bir yerlerde unutulması, dinlenme noktaları yada araç içinde bırakılması durumlarında çalınma vakaları oldukça fazladır.

Endpoint Encryption istemci(dizüstü,masaüstü), iş istasyonu ve sunucu sistemlerinde bulunan disk(ler)i komple şifreleyen bir yazılımdır. Aynı zamanda harici disk, çıkarılabilir cihazlarda(usb), CD/DVD/Blue-Ray medyalarında dosya şifrelemesi yapmaktadır.



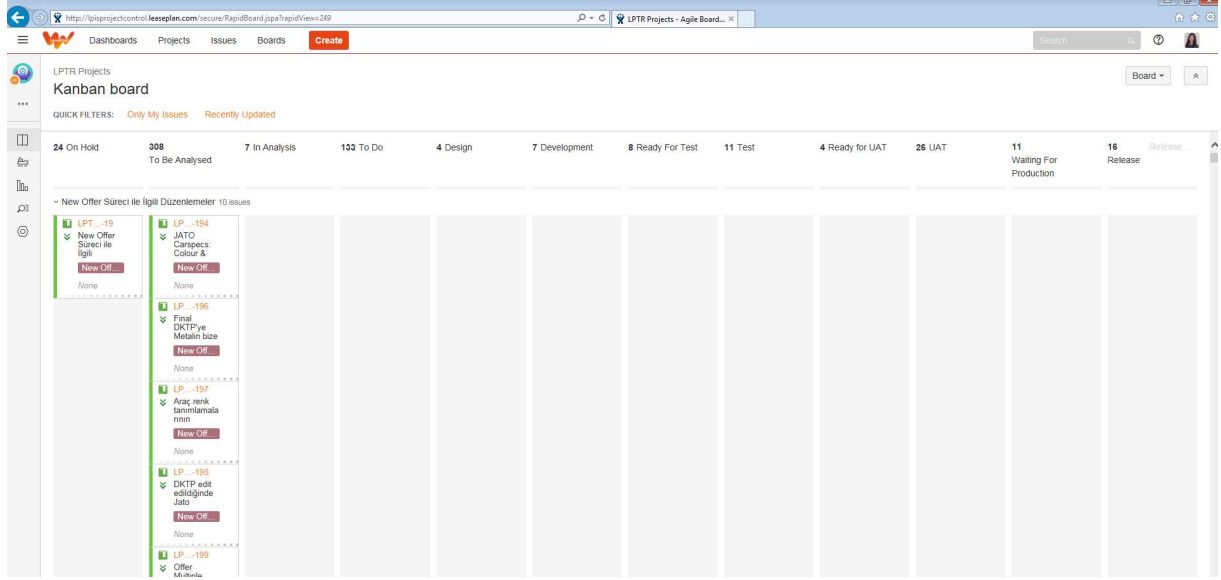
Disk şifreleme işleminden sonra kullanıcı, işletim sistemine erişim sağlarken ek güvenlik kontrolünden geçer. Ayrıca, tüm diskin şifreli olmasından dolayı yetkisiz kişiler tarafından işletim sistemine ve verilere erişimi mümkün değildir.

Active Directory

Active Directory, Microsoft tarafından özellikle Windows Server ve Client bilgisayar sistemleri için tasarlanmış olan içerisinde sunucu, client bilgisayar, kullanıcı ve yazıcı gibi bilgileri tutan bir dizin servsidir. Tabi bahsi geçen verileri tuttuğu için aynı bir veri tabanıdır. Bu servis içerisinde yer alan Group Policy yönetim aracı ile çeşitli kısıtlamalar yapabilir veya tek bir noktadan istediğimiz uygulamanın dağıtımını gerçekleştirebiliriz. Kaynakların kontrolü ve yönetiminin merkezleştirilmesi açısından büyük kolaylık sağladığı için çok tercih edilen bir servistir.

02.08.2018

Stajım dördüncü gününde iş analistlerinin yanında yerimi aldım. İlk olarak İş Analisti Elifnur Elmasrı ile tanıştık. Ona stajımın gerekliliklerini anlattım. Elifnur Hanım da benim için ne gibi bir proje verebilir araştıracağını söyledi. Onun dışında kullandıkları “JIRA” uygulamasını, yapımı devam eden projenin süreçlerini ve işleyişlerini anlattı.



Üstte bulunan ekran görüntüsü analistlerin kullandığı board. Toplamda 12 aşama bulunuyor. Sırasıyla,

On Hold: Oluşturulan issue'nun henüz beklemede olduğu aşama.

To Be Analysed: Issue'nun paylaşıldığı analiz edileceğini gösteren aşama.

In Analysis: Issue'nun analiz sürecinde olduğunu gösteren aşama.

To Do: Yapılacakların not edildiği aşama.

Desing: Issue yani talebin doğrultusunda yapılacaklar not edildikten sonra, yine analistler ürünün dizaynının yapıldığı aşama.

Development: Bu sekme ise yazılım geliştiricilerin devreye girerek kodlarla yeni ürünü oluşturma aşaması.

Ready For Test: Yazılım geliştiricilerin kodlamaları bittiğinde geçilen aşama.

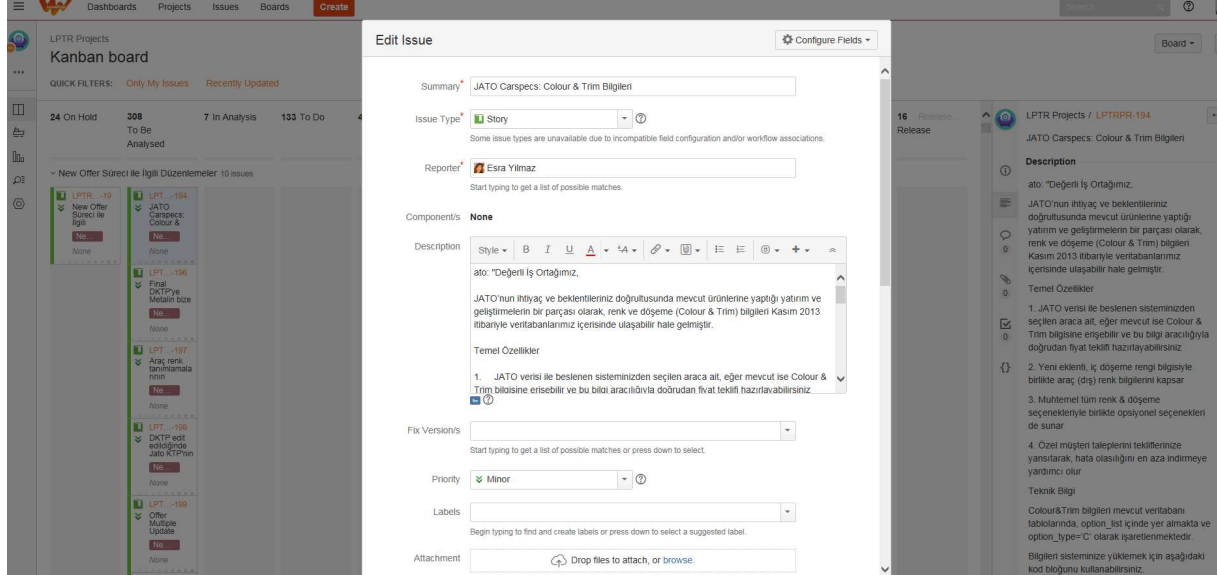
Test: Analistlerin ya da diğer şirketler de Test Analistlerin ürünün talepleri karşıladığını test ettiği aşama.

Ready for UAT: Taleplerin karşılandığını test edildikten sonra kullanıcı testine geçmeyi beklenen aşama.

UAT: Oluşturulan ürünün kullanıcılar tarafından test edildiği aşama.

Ready For Production: Ürünün tüm testleri bittikten sonra üretime geçmeye hazır olduğunu gösteren aşama.

Release: Onların tabiriyle ürünün ayağa kaldırılması aşaması. Üretim.



Üstte ki ekran görüntüsü bahsedilen Issue'ların editleme görüntüsüdür. Raporlar ve talepler burada sıralanır. Ürünün senaryosu ve öncelik sırası belirlenir.

Daha sonrasında İş Analisti Müdürü Sıgla Olgun beni bir toplantıya davet etti. Toplantıya katıldım. Toplantı sonrasında Sıgla Hanım beni yine bırakmadı. Analistlerin çokça kullandığı metotları anlattı. Yazılım geliştirme modelleri olan Waterfall, Scrum ve Agile Metotları ile ilgili bir ders verdi ve benden araştırmamı istedi. Günün devamında ve ertesi günü Sıgla Hanım'ın istediği araştırmaları yaptım.

YAZILIM GELİŞTİRME MODELLERİ ARAŞTIRMASI

Geliştirdiğimiz yazılımın, üretim aşaması ve kullanım süreci boyunca geçirdiği tüm aşamaları yazılım geliştirme yaşam döngüsü olarak tanımlıyoruz. Her yazılım, bu döngüde planlama-analiz-tasarım-üretim-bakım aşamalarından geçer. Yazılım yaşam döngüsündeki bu temel adımların nasıl gerçekleştirileceğine yönelik çeşitli modeller geliştirilmiş. Yazılım geliştirme modelleri, elimizdeki projenin 'hangi felsefe' doğrultusunda işi yürüteceğimize karar verdikten sonra kullanacağımız metodolojilerdir. Projenin hangi felsefeye göre yürütüleceğine ise iş ihtiyacına göre karar veririz.

Özetleyecek olursak; bir yazılım geliştirme metodolojisi aşağıdaki adımlardan meydana gelir:

- Yazılım geliştirme süreci yaklaşımıyla bir yazılım geliştirme felsefesi
- Yazılım geliştirme sürecine destek veren araçlar, modeller ve yöntemler

Çokça bilinen yazılım geliştirme modelleri:

- Waterfall (Şelale) Model
- Agile Software Development (Çevik Yazılım Geliştirme) Model
- Prototyping (Prototip) Model
- Incremental (Artırımsal) Model
- Spiral (Sarmal) Model
- Rapid Application Development (Hızlı Uygulama Geliştirme) Model
- Object-Oriented Analysis and Design (Nesne Yönelimli Analiz ve Tasarım) Model

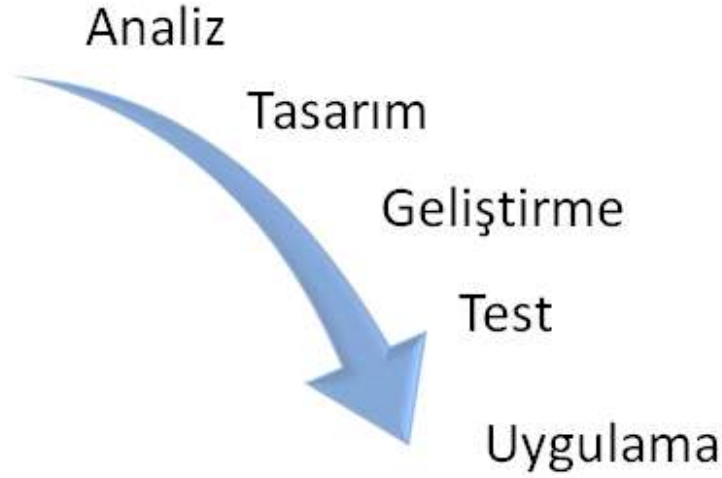
Geleneksel yazılım geliştirme süreçlerinde çoğunlukla Waterfall (şelale) modeli kullanılır. LeasePlan şirketi daha önceleri Waterfall modeli kullanıyorken. Şimdilerde ise Çevik modele geçmiş bulunuyorlar.

Waterfall (Şelale) Modeli

Waterfall modeli, yazılım geliřtirmenin gereksinim analizi, tasarım, geliřtirme, test ve bakım gibi fazları boyunca ařađıya dođru (řelale gibi) devam eder gibi görüldüđü sıralı geliřtirme sürecidir.

Genel Özellikler:

- Şelalenin her basamađında yer alan aktiviteler eksiksiz olarak yerine getirilir. Bu, sonraki basamađa geçmenin şartıdır.
- Her safhanın sonunda bir doküman oluşturulur. Bu yüzden waterfall modeli doküman güdümlüdür.
- Yazılım süreci doğrusaldır, yani bir sonraki safhaya geçebilmek için bir önceki safhada yer alan aktivitelerin tamamlanmış olması gerekir.
- Kullanıcı katılımı başlangıç safhasında mümkündür. Kullanıcı gereksinimleri bu safhada tespit edilir ve detaylandırılır. Daha sonra gelen tasarım ve kodlama safhalarında müşteri ve kullanıcılar ile diyalođa girilmez.



Teoride her ne kadar her řey mükemmel görünse de bu prensipler pratik hayata geçirildiđinde her řey bu kadar mükemmel olmayabiliyor.

Problemler:

- Bir safha bitmeden diğer safhaya geçilememesi müşterinin tüm gereksinimlerini ilk aşamada tanımlamayı gerektirir. Yazılımcıların büyük çoğunluğunun tecrübe ettiği gibi;
 - Proje başlangıcında müşteriler, tam olarak ne istediklerinden yüzde yüz emin değildirler.
 - Müşteriler yazılım teknolojilerinin içinde değillerse isteklerini net bir şekilde dile getirmekte zorlanabilirler.
 - Müşterilerin asıl istekleriyle ilgili detaylar proje başlangıcında fark edilebilir değildirler, süreç içerisinde yüzeye çıkarlar.
 - Müşteriler isteklerinin gerçekleştiğini gördükçe, proje gözle görülür sonuçlar üretmeye başladıkça, isteklerinde değişiklik yapma eğilimindedirler.
 - Dış koşullar değişim içerisindedir ve projelerin bunlardan etkilenecek değişime uğramaları kaçınılmazdır. Şelale yöntemi ile müşterinin istediği yazılım sistemi proje sonunda tamamlanır. Ancak bu safhada müşteri yazılım sistemini test edebilecektir. Müşteri tamamlanan yazılım sistemini tüm artı ve eksileriyle kabullenmek ve kullanmak zorundadır. Dolayısıyla tüm gereksinimlerin en başta tanımlanıp daha sonra geri dönülememesi prensibi ürün geliştirme sürecinde oluşabilecek her türlü değişikliği göz ardı etmek demektir.
- Müşterinin talebinin anlaşılması ve tüm gereksinimler için detaylı dokümanların oluşturulması, yazılımcının bunları okuyup uygulaması, test aşamasında oluşabilecek herhangi bir hatada her şeyin sil baştan yapılması; projenin hayata geçme süresini uzatacaktır.
- Başlangıçta yapılan hataların tespiti çok uzun zaman alabilir. Bu hataların giderilmesi maliyeti yükseltecektir.

Waterfall modelinde; bunun gibi belki 10 tane daha gereksinim yazıp bunları hangi program kullanılacaksa, o programın gerektirdikleriyle birlikte tasarımını yapıp uzun bir doküman hazırlayıp yazılımcıya teslim etmemiz gerekiyor, yazılım geliştirme aşamasından sonra test edilecek; müşterinin gereksinimlerinde bir değişiklik olduğunda herhangi bir hata olması halinde ya da test aşamasında herhangi bir hata olması durumunda tekrar en başa dönecek.

Agile modelinde ise; her bir gereksinim üzerinden bu süreç yürütülecek.

- Yani diyeceğiz ki; biz önce bir notepad şeklinde insanların sadece yapacakları şeyleri listeleyeceği bir ekran yapalım. Bu uygulamaya konduktan sonra son kullanıcıdan, yazılım ekibinden ya da bizden bu projeyi isteyen yatırımcıdan feedback alabileceğiz. Ve belki de diyecek ki son kullanıcı: "İş hayatımı ve özel hayatımı ayırmak istiyorum ben, notlarımı farklı renklerde yapabileyim", "Not aldığım ekrana .doc, .pdf uzantılı dokümanlar ekleyebileyim". Bu şekilde geri bildirim aldığımızda projemizde değişiklik yapabileceğiz.
- Belki de proje için fazla para harcıyoruz. Bunu projenin en başında farkedebileceğiz ve bundan sonraki süreci ona göre ayarlayabileceğiz. Bu da bizim maliyetimizi düşürecek.
- Waterfall gibi klasik yöntemlerle oluşabilecek herhangi bir hata için tüm sistemin sil baştan dönmesi gerekirken burada hatayı en baştan farkedebileceğiz. Bu da bizim sürecimizi kısaltacak.

Dolayısıyla Agile modeli sayesinde boşu boşuna para, zaman ve emek harcamamış olacağız.



Agile Modeli içerisinde en çok kullanılan metodolojileri (frameworkleri) şu şekilde sıralayabiliriz:

- Scrum
- Kanban
- Lean
- Extreme Programming (XP)
- Crystal
- Test-Driven Development
- Feature-Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)



Bu haftaya iş zekası ve yazılım geliştirme ekipleri ile tanışarak başladım. Sıgla Hanım beni tek tek birimdeki çalışma arkadaşları ile tanıştırdı. İş zekası bölümünden Burak Tuğrul ve yazılım geliştirme bölümünden Ayla Yılmaz bana çeşitli araştırma görevleri verdiler. Tüm haftam bu araştırmalarla geçecek.

.NET FRAMEWORK ARAŞTIRMASI

NET Framework, Microsoft tarafından geliştirilen, Windows, Windows Phone, Windows Store, Widows Server ve Azure uygulamaları geliştirmek için açık internet protokolleri ve standartları üzerine kurulmuş komple bir uygulama geliştirme platformudur. .NET Framework içinde yazdığımız kodları makine diline çevirmeye yarayan derleyiciler, program yazarken bazı işlerin otomatik yapılmasını sağlayan metotlar ve yardım dokümanları bulunmaktadır. Bu platform, işletim sisteminden ve donanımdan daha üst seviyede taşınabilir olarak tasarlanmıştır. .Net mimarisi, ortak bir yürütme ortamı (runtime environment), ortak bir değişken tür sistemi, ve birbirleriyle bağlantılı kütüphanelerden oluşur.

.NET'in Getirdiği Çözümler;

- Varolan kodlarla tam çalışabilirlik desteği.
- Tüm .NET dilleri tarafından paylaşılan ortak bir çalışma zamanı: Ortak Çalışma Zamanı (Common Language Runtime), daha önce uygulama geliştiricinin düşünmek zorunda olduğu birçok işin üstesinden gelir. Bellek yönetimi(Memory management), tip güvenliği (Type safety), istisna yönetimi (Exception handling) vb.).
- Çoklu dil desteği: Microsoft radikal bir karar alarak CLR ile uyumlu her .NET dilinin kullanılmasına olanak sağlıyor.
- Tüm .NET dilleri tarafından paylaşılan ortak temel sınıf kütüphanesi:.NET ile birlikte uygulama geliştiricinin hizmetine sunulan 3500'den fazla sınıftan oluşan zengin kütüphane, daha hızlı program geliştirme imkanı ve bütün .NET dilleri tarafından kullanılan tutarlı bir nesne modeli sunuyor.
- Programlama modelinden bağımsız uygulama geliştirme ortamı: Tek bir uygulama geliştirme ortamı uygulamaları geliştirilebilir.
- Basitleştirilmiş masaüstü uygulama geliştirme ve yayınlama modeli: .NET ortamında geliştirilen bir masaüstü uygulaması, herhangi bir windows işletim sisteminin kurulu olduğu makinede çalıştırılabilir, gereken tek şey .NET Framework'ünün kurulu olmasıdır.

.NET'in üç yapıtaşı CLR, CTS ve CLS'i inceleyelim;

Ortak Çalışma Zamanı (Common Language Runtime -CLR-)

CLR'in birincil rolü .NET tiplerinin yerini öğrenmek, bu tipleri kendi ortamına yüklemek ve yönetmektir. CLR ayrıca bellek yönetimi ve tip güvenlik kontrollerini yerine getirmek gibi birçok alt seviye ayrıntıdan da sorumludur.

Ortak Tip Sistemi (Common Type System -CTS-)

CTS spesifikasyonları, çalışma zamanı tarafından desteklenen bütün veri tipleri ve programlama yapılarını tanımlar, bu yapıların birbirleriyle nasıl etkileşeceklerini açıkça belirtir ve .NET metadata formatında nasıl temsil edileceklerinin ayrıntılarını belirler. Böylece .NET destekli tiplerin, aynı veri tiplerini kullanabilmesi sağlanabilmektedir.

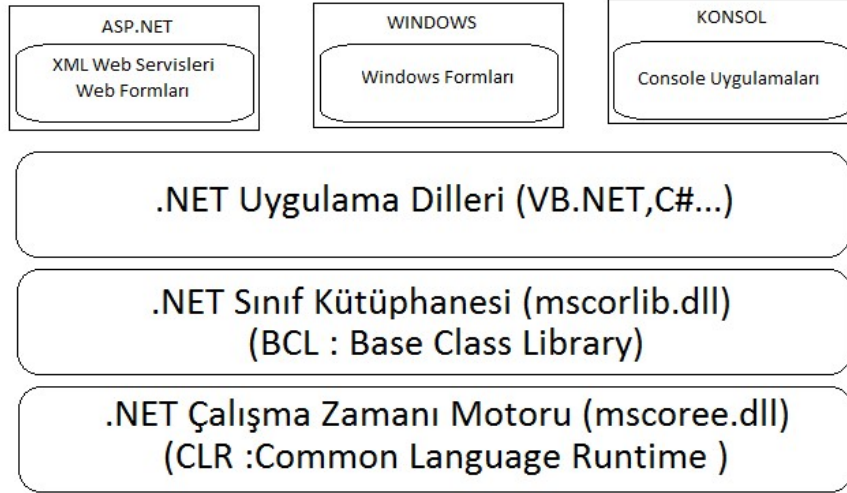
Ortak Dil Spesifikasyonları (Common Language Specification -CLS-)

.NET tabanlı bir programlama dili, ortak tip sistemi CTS tarafından tanımlanan her bir özelliği desteklemeyebilir. Common Language Specification (tüm .NET dillerinin orta noktada buluşabileceği ortak tip ve programlama altyapısını tanımlayan yönerge listesidir. Dolayısıyla CLS uyumlu özelliklere sahip bir .NET tipi geliştirilirse, bu tipi diğer bütün .NET dillerinin kullanabileceği garanti edilmiş olur. Tersine CLS sınırlarının dışında bir veri tipi ya da programlama yapısı oluşturulursa, bu kod kütüphanesi ile bütün .NET dillerinin sağlıklı bir şekilde etkileşebileceği garanti edilemez.

Temel Sınıf Kütüphanesi (Base Class Library)

CLR ve CTS/CLS spesifikasyonlarına ek olarak .NET platformu, tüm .NET programlama dillerinin kullanabileceği Base Class Library'i (temel sınıf kütüphanesi) sunar. Her biri belli bir görevi yerine getirmekle sorumlu olan sınıflardan oluşan bu kütüphane hem temel işler (thread -kanal-, dosya giriş çıkış,grafiksel görünüm...) için kullanılacak tipleri içerir hem de gerçek hayat uygulamalarının ihtiyaç duyacağı birçok servise destek sağlar. Örneğin temel sınıf kütüphanesinin bize sağladığı tipler veritabanı erişimi, xml etkileşimleri, programatik güvenlik konularını ele almayı ve web,masaüstü ya da konsol tabanlı kullanıcı ara yüzleri geliştirmeyi çok kolaylaştırır. .NET platformunun yazılım geliştiricilere sunduğu bu koleksiyonun güzel yanlarından biri de kullanılmasının son derece kolay olmasıdır. Kullanılan isimler o kadar açıklayıcıdır ki temel ingilizce bilgisine sahip birisi için sezgisel olarak hangi üyenin kullanacağını bulunması çok fazla zaman almamaktadır.

.NET UYGULAMALARI



Aşağıda örnek kütüphaneleri vereceğim;

#System.IO: Dosya, okuma, yazma gibi birçok işlem yapılmasını sağlayan sınıflar bulunmakta. **#System.Globalization :** Dil, takvim, para birimi işlemleri.

#System.Text: Encoding, decoding ve text işlemleri

#System: Bu assembly, generic koleksiyonları, işletim sistemi bazında takip eden işlemleri yapan sınıfları(dosya, mail, güvenlik vb.) içerir. Örnek; SmtClient, FileSystemWatcher.

#System.Security: Güvenlik ve yetkilendirme.

#System.Runtime.InteropServices: Eskiden yazılmış COM kütüphaneleri ile birlikte .NET uygulamaları geliştirmemizi sağlayan hazır yapıları kullanmamızı sağlar.

#System.Reflection : Assembly'lerin metadatalarını sorgulayarak sınıflar, metotlar alanlar gibi birçok özellik hakkında bilgi toplayabiliriz.

#System.Threading : Yazdığımız uygulamaların aynı zamanda paralel işlemler sağlamak için kullanılır.

Assembly : Bir ya daha fazla sınıfın içinde kayıtlı olduğu dosyalara assembly denir. Assembly dosyaları genelde .dll uzantısına sahiptir fakat .exe uzantısına sahiptir fakat .exe uzantısına da sahip olabilirler. .NET binary'leri, platformdan bağımsız Common Intermediate Language(CIL) adındaki ara dili içerirler. Assembly CIL dışında bir de metadata içerir.

Metadata : Assembly adı, versiyonu, kültür bilgisi, kısa bir açıklama, başka assembly'lere olan referanslar gibi bilgilerin tamamına assembly metadata denir.

Manifesto: Assembly'nin kendisine erişimini sağlayan ve kendisini tanımlayan bilgilerdir.

Kaynaklar: Assembly'deki üyelerin kullandığı dosya, resim gibi ek materyeller.

CIL : C# kaynak kodunun derlenmesi sonucu oluşan, intermediate language komutları.

APN.NET ARAŞTIRMASI

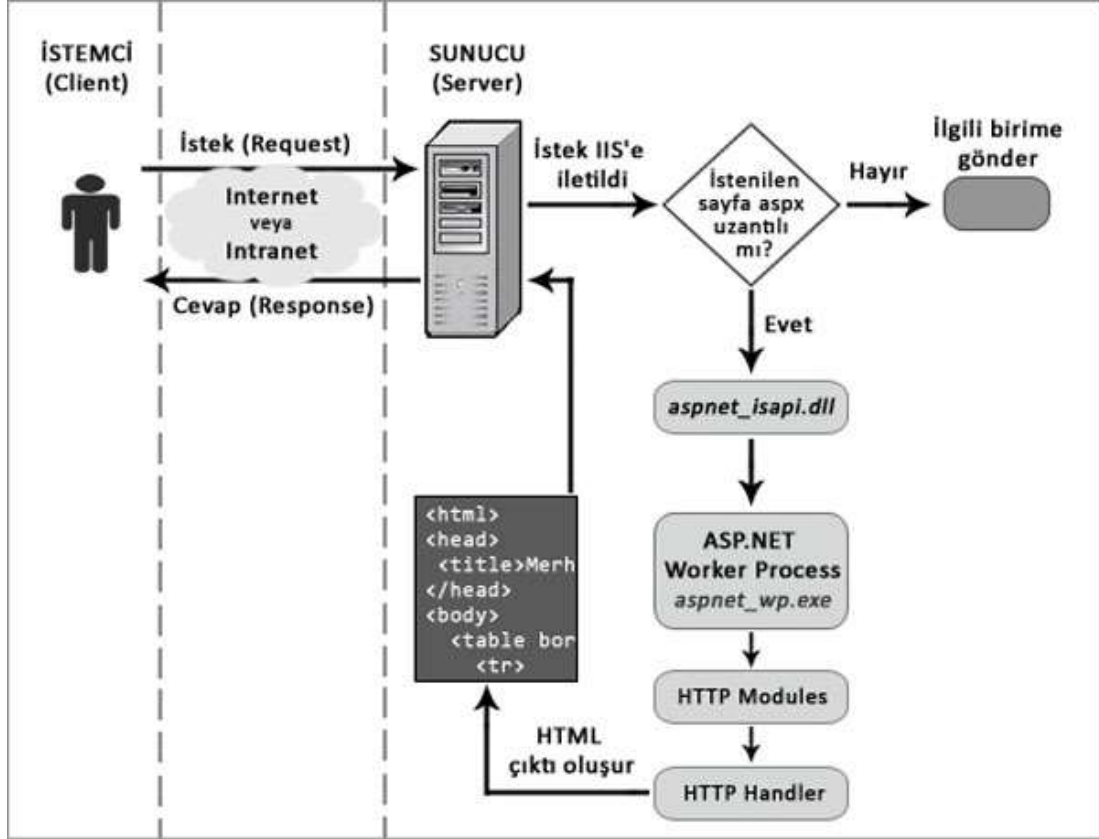
Microsoft'un geliştirdiği, .Net Framework üzerine çalışan web uygulama dilidir. Asp.Net sayesinde web programcılarını kolaylıkla dinamik web siteleri, web uygulamaları veya XML web servislerini geliştirebilir. Asp.net modern bir teknolojidir ve web sayfaları, web uygulamaları ve XML web servisleri hazırlanmasına olanak sağlar. ASP.NET programlama dili değildir. Asp.net ile bir web uygulaması hazırlamak için .net dillerinden (C#, VB.net..) birisi kullanılabilir. .Net platformunun nesnelere Asp.net içerisinde de kullanılabilir. Asp.net ile bir web uygulaması geliştirmek için Visual Studio kullanılır.

Asp.net'in Temel Özellikleri:

- Asp.net code behind denilen tasarım öğeleri ile programlama öğelerini bir birinden ayrı sayfalarda bulunduran bir özelliği destekler. Bu özellik sayesinde bir proje üzerinde tasarımcı ve programcı aynı anda çalışabilir.
- Sayfanın tasarım öğeleri (web forms için) .aspx uzantılı dosyada bulunurken program nesnelere .aspx.cs (Csharp için .cs, VB.net için .vb) uzantılı dosyalarda bulundurulur. Asp.net' te Web Forms veya MVC programlama modellerini kullanarak uygulamalar geliştirebilirsiniz.
- Microsoft .Net platformunun tüm özellikleri Asp.net uygulamaları için geçerlidir.
- Visual Studio içerisindeki web kontrolleri (buton, textbox, dropdown list vs..) sürükle-bırak özelliği kullanarak hızlı ve basitçe uygulamaya dahil edilebilir.
- Modern web uygulama arayüzleri oluşturabilmek için MasterPage ve Theme yapıları mevcuttur. Asp.net' te gelişmiş oturum yönetimi desteği mevcuttur. Bu sayede üyelik işlemleri basit ve güvenli olarak yapılabilir.
- Asp.NET teknolojisi oldukça hızlıdır, eski ASP'ye göre her seferinde sayfaların tekrar tekrar okunarak her istekte bir istek olmasını engelleyerek, daha hızlı ve daha performanslı çalışmasını sağlar.
- ASP.NET , XML, Web Servisleri, Veri tabanı etkileşimi ve email etkileşimi gibi bir çok teknoloji tarafından desteklenir ve bir çok teknolojiyi destekler.
- ASP.NET kullanarak örnek e-ticaret siteleri, haber portalları, çeşitli kurumsal uygulamalar, forum siteleri geliştirilebilir ve yönetilebilir.

İstemci-sunucu mimarisi oldukça başarılı bir şekilde ASP.NET’de uygulanmış, yazılım geliştiricinin arka planda neler olduğunu bilmesine gerek kalmadan uygulamalarda önbellekleme ve performans düzenlenmiştir. ASP.NET tamamen nesneye yönelik programlamayı kullanabilmemizi sağlayan bir teknolojidir.

Asp.net’ in istemci-sunucu mimarisi aşağıdaki görseldeki gibidir:



Asp.Net Web Uygulama Örnekleri:

Asp.net’i daha iyi anlamanız için bir kaç gerçek uygulama örnekleri gösterelim. Tüm diğer Microsoft sitelerinde olduğu gibi eğitim portalı Açık Akademi Asp.net ürünüdür. Türkiye üniversitelerinin çok büyük bir kesiminin öğrenci otomasyonları yine Asp.net ürünü. Hepsi Burada, Ebay gibi alışveriş siteleri ile Vakıf Bank, Teb, Finans Bank, Deniz Bank gibi bankaların siteleri ve internet şubeleri asp.net uygulamalarıdır.

Ebay gibi alışveriş siteleri ile Vakıf Bank, Teb, Finans Bank, Deniz Bank gibi bankaların siteleri ve internet şubeleri asp.net uygulamalarıdır.

C# YAPISI ARAŞTIRMASI

Özellikle .NET uygulamalarını üretmek için Microsoft tarafından geliştirilmiş ve halen geliştirilmeye devam eden bir dildir. C ve C# dilleri C# dillerinin atasıdır. C#, C++ dilinin gücünü ve Visual Basic dilinin kolaylıklarını barındırmaktadır. C# nesne tabanlı, olay güdümlü oldukça güçlü bir dildir. C# dilini kullanarak hem masaüstü, hem web, hem de mobil uygulamalar geliştirebilirsiniz.

Özellikle Türkiye’de C#, gerek Windows uygulamaları gerekse ASP.NET uygulamaları için en çok tercih edilen dildir.

- Yorum satırı // Eğer yapacağımız açıklama tek satırdan oluşuyorsa bunun için iki tane bölü işareti // kullanılır. Birden fazla açıklama satırı kullanılacaksa /* */ karakterleri arasına da yazılabilir.
- Console.WriteLine("..."); Kod satırı ekrana mesaj yazdırır. Mesajı ekrana yazıp bir satır aşağı iner. Bir alt satıra inmesini Line ifadesi gerçekleştirir. Eğer Console.Write("..."); kullanılmış olsaydı alt satıra inmezdi.
- Mesaj yerine değişken yazıldığı zaman tırnak içine yazılmaz. string mesaj = "merhaba dünya"; Console.WriteLine(mesaj);
- Console.ReadKey(); : Herhangi bir tuşa basılmasını bekler. Bu kod kullanılmazsa konsol penceresi açılıp hemen kapanacaktır. Bunun yerine ctrl+f5 tuş kombinasyonu da kullanılabilir.
- void Main C# konsol uygulamasında bir void main metodu bulunmalıdır, programımız bu kontrol ile başlar ve biter.

Değişkenler, bir programlama dilinde verilerin depolanma alanlarını temsil eder. Tanımlanan her değişkene bellek bölgesinden bir alan ayrılır. Bu bellek bölgesine okuma ve yazma işlemleri ise değişken ismi üzerinden sağlanır. “ , “Bir veri tipiyle birden fazla değişken tanımlanabilirken ‘virgül işareti’ ile birbirinden ayrılır. “;”

Her kod satırının sonunda, kod satırının sona erdiğini belirten ‘noktalı virgül işareti’ mutlaka kullanılır.

Örnek : <veri tipi><değişken adı>; Değişken adı içerisinde boşluk kullanılamaz. Eğer değişken adı birden fazla kelimedenden oluşuyorsa kelimeler arasına _ kullanılması tavsiye edilir. Değişkenlerin ilk karakteri harf ya da _ (alt çizgi) ile başlamalıdır. Sayı veya diğer özel karakterlerle başlayamaz. İlk karakter dışında değişken içinde sayı olabilir. Değişken içinde _ (alt çizgi) dışında özel karakterler !,], ?, -, }, >, ” gibi kullanılamaz. Programın kendi kodları değişken adı olarak kullanılamaz.

Bazı yerlerde hata vermese bile, Türkçe karakterler kullanmamaya dikkat etmeliyiz.

- “ +, -, *, / ” **Operatörleri** kullanılarak hesaplama işlemleri yapılabilir. Sağda hesaplanan değerler soldaki değişkene atanmış oluyor.
- == **Operatörü**: Aynı türdeki iki değer birbirine eşitliğini kontrol eden operatördür. Bu iki değer birbirine eşit ise True değilse False değeri döndürülür.
- != **Operatörü**: Aynı türdeki iki değer birbirine eşit olmadığının (eşit değil) kontrolü için kullanılan operatördür.
- <, >, <=, >= **Operatörü**: Bir değer aynı türdeki başka bir değerden küçüklüğünü(<) ve büyüklüğünü(>) kontrolünün yapıldığı operatördür. Bu operatör string ifadelerde kullanılmaz. Char ifadelerinde kullanılabilir.
- *= **Operatörü**: sağdaki ifade ile soldaki ifade çarpılır. Sonuç soldaki ifadenin değeri olur.
- /= **Operatörü**: sağdaki ifade soldaki ifadeye bölünür. Sonuç soldaki ifadenin değeri olur.
- %= **Operatörü**: soldaki ifadenin sağdaki ifadeye bölümünden kalan bulunur. Bu kalan değer soldaki ifadenin değeri olur.

- **++ Operatörü:** bir deęişkenin deęerini bir artırır. Deęişkenin önünde ve arkasında kullanılabilir. Önünde kullanıldığında önce deęişkenin deęerini artırır sonra atama yapar. Arkasında kullanıldığı zaman önce atama yapar sonra artırır.
- **-- Operatörü:** Bir deęişkenin deęerini azaltır. Deęişkenin önünde ve arkasında kullanılabilir. Önünde kullanıldığında önce deęişkenin deęerini azaltır sonra atama yapar. Arkasında kullanıldığı zaman önce atama yapar sonra azaltır.
- **&& Operatörü:** Mantıksal ve operatörü bütün koşulların doğru (true) olması durumunda sonuç doğru (true), koşullardan herhangi biri yanlış (false) olduğu zaman sonuç yanlış (false) olur.
- **|| Operatörü:** Mantıksal ve operatörü bütün koşulların false (yanlış) olması durumunda sonuç false (yanlış), koşullardan herhangi biri true (doęru) olduğu zaman sonuç true (doęru) olur.
- **! Operatörü:** “Deęil” anlamına gelir. Boolean ifadeleri tersine çevirir.
- **?: Koęul Operatörü :** Verilen koęula göre iki ifadeden biri çalışır. Koęun doęru olması ve yanlış olması durumuna göre çalışır.

SQL ARAŞTIRMASI

Veri (data), bir veya birden fazla bilgiden oluşan bir kümedir.

Veri tabanı (Database) temel olarak farklı tiplerdeki verileri düzenli bir şekilde saklamamızı ve kullanmamızı sağlayan depolama ortamıdır. Bu ortam içerisinde verileri saklayabilir, onlara kolay bir şekilde ulaşabilir ve gerektiğinde bu verilerin üzerinde değişiklikler yapabiliriz. Veri tabanı, veriler arasında bütünlük ve düzen sağlarken, veriye hızlı erişim ve bakım kolaylığı da sunar.

Veri Tabanı Yönetim Sistemleri (DataBase Management System - DBMS), veri tabanında tutulacak olan verilerin uyacağı standartları, bu verilere nasıl erişilebileceğini ve verilerin disk üzerinde nasıl tutulacağını belirleyen sistemlerdir.

İlişkisel Veri Tabanı Yönetim Sistemi (Relational Database Management System - RDBMS) Günümüzde en yaygın olarak kullanılan veri tabanı mimarisidir. Veriler tablolar, satırlar ve alanlar halinde tutulmaktadır. Bu sayede veriler arasında ilişkilendirmeler yapılabilmekte, verilerin kullanımı ve yönetimi daha verimli hale getirilmektedir.

Veri tabanı üzerinde gerçekleştirilecek olan işlemleri, programlama dilleri aracılığıyla yapabildiğimiz gibi SQL Server Management Studio uygulamasıyla da gerçekleştirebiliriz. SQL Server Management Studio, Microsoft SQL Server ile birlikte gelmiş olan yeni bir sorgulama ve yönetim aracıdır. Management Studio, SQL Server veritabanlarına erişme, veritabanları üzerinde ayarlama işlemlerini gerçekleştirme, yönetim ve veri tabanı üzerinde sorgular çalıştırma gibi işlemleri yapabilmektedir. Yine Management Studio üzerinde SQL Server projeleri oluşturulup, çalışmaların Visual Studio ortamında olduğu gibi proje şeklinde kaydedilip, daha sonra bu projeler üzerinde geliştirme işlemleri yapılabilmesi sağlanmaktadır. SQL Server Management Studio aynı zamanda, veri tabanı üzerinde T-SQL sorgularını çalıştırmak, sonuçlarını görmek, sorguları analiz etmek ve veri tabanı üzerinde bazı ayarlamaları yapmak için de kullanılabilir.

SQL Server'ı açtığımız zaman ekranda karşımıza sadece Object Explorer ekranı yan ekran olarak çıkar.

Object Explorer: Bağlı bulunan SQL Server sunucusu üzerinde bulunan tüm nesnelere ağaç yapısı biçiminde görüntülememizi ve bu yapıların içerisinde gezinmemizi sağlar. Ayrıca yeni veri tabanı oluşturma, tablo ekleme, varolan nesnelere silme gibi işlemler yapılabilmektedir. Görüntülenen nesnelere üzerinde filtreleme işlemleri yapılabilmektedir. Management Studio ile birden fazla SQL Server sunucusuna bağlanabilir, bağlı bulunduğu tüm sunucuları Object Explorer penceresi içerisinde görüntüleyebilir ve üzerlerinde işlemler yapabiliriz.

Object Explorer üzerinde yapabileceğimiz bazı temel işlemler şunlardır:

- Yeni bir SQL Server sunucusuna bağlanmak için Object Explorer penceresinin sol üst köşesinde yer alan Connect butonu aracılığı ile aynı anda birden fazla SQL Server sunucusuna bağlanılabilir.
- Bağlı olduğun SQL Server’da sorgu çalıştırmak için, sunucu adının üzerine sağ tıklayıp New Query seçeneği seçilir.
- Bir veritabanı üzerinde sorgu çalıştırmak için o veritabanının üzerine sağ tıklayıp New Query seçeneği seçilir.
- Veritabanına yeni bir tablo eklemek için veritabanı içindeki Tables kısmında sağ tıklayarak New Table seçeneği seçilir.
- Bir tablonun yapısını incelemek ve değişiklikler yapmak için tablo üzerine sağ tıklayıp Design seçeneği seçilir.
- Bir tablo içerisindeki kayıtları görmek için tablo üzerine sağ tıklayıp Select Top 1000 Rows seçeneğini kullanmak gerekir.

Object Explorer Details: Bu ekranda ise Object Explorer içerisinde o an için seçilmiş olan nesnenin içeriği görüntülenmektedir. Pencerenin kendi içerisinden bir üstteki veya bir alttaki yapılara ulaşılabilir. Veri tabanı içerisinde gezinme işlemlerini daha kolay hale getirmek için bu pencere kullanılabilir.

Template Explorer: Bazı işlemleri basit bir şekilde yapmamızı sağlayan sorgu şablonları(template) bulunmaktadır. Bu şablonları açıp gerekli değişiklikleri yaparak sorguları çalıştırılabilir ve hızlı bir şekilde işlemler gerçekleştirilebilir. Şablon içerisinde değiştirilmesi gereken parametreler, Query menüsünden Specify Values for Template Parameters seçeneği ile çıkan pencerede gerekli parametreler girilerek düzenlenebilir.

SQL (Yapısal Sorgulama Dili)

SQL, yani Structured Query Language (Yapısal Sorgulama Dili) tüm ilişkisel veritabanlarında standart olarak kullanılan bir dildir. SQL insanların veritabanı ile konuşmasını sağlayan popüler bir dildir. Bu dil sayesinde veritabanından kayıtları alabilir, değiştirebilir ya da yenisini ekleyebiliriz. SQL dilinin standartları ANSI (American National Standart Institute) ve ISO (International Standarts Organization) tarafından sağlanmakla birlikte, günümüzde en yaygın olarak ANSI standartları kullanılmaktadır. SQL bir dildir ancak programlama dili değildir.

10.08.2018

Projeme başlamadan önce SQL le ilgili temel sorgulamaları bilmek konuların anlaşılmasında yardımcı olacağını düşündüğüm için araştırmalarımı bugünde devam ettim.

SQL komutları kullanılarak aşağıdaki işlemler yapılabilir:

- Veritabanı nesnelere oluşturulması ve bu nesnelere ilgili işlemlerin yapılması
- Bilgilerin istenilen koşullara göre görüntülenmesi ve sorgulama işlemleri
- Tablolara veri girişi yapılması
- Bilgilerin güncelleştirilmesi
- Tabloların veya tablolardaki verilerin silinmesi
- Kullanıcı ve rollerin oluşturulması, yetkilerin düzenlenmesi.

SELECT Deyimi

SELECT kolon isimleri **FROM** tablo adı **WHERE** koşul ifadesi;

SELECT NO,ADI,SOYADI **FROM** PERSONEL **WHERE** SOYADI='TULU';

Yardımcı Deyimler

FROM Sorgulamanın yapılacağı tabloyu tanımlar.

WHERE Sorgulamada kullanılacak koşul ifadesini tanımlar.

GROUP BY Bilgilerin belirlenen alanlara göre gruplandırılmasını sağlar.

HAVING Her bir grup için uyulması gereken koşul tanımlanır.

ORDER BY Seçilen kayıtların belirli alan veya alanlara göre sıralanmasını sağlar.

SQL Komutları

SQL komutlarını ikiye ayırabiliriz.

1. DDL (Data Definition Language - veri tanımlama dili)
2. DML (Data ManipulationLanguage - veri işleme dili)

SQL Veri Tipleri

NUMBER: SQL deyimleri içinde tüm sayısal verileri tanımlamak üzere NUMBER veri türü kullanılır. Sayısal veriler en fazla 38 haneye kadar olabilir.

- **MIKTAR NUMBER(8):** 8 hanelik yer ayrılır.
- **MIKTAR NUMBER(7,2):** virgülden önce 7 hane, virgülden sonra 2 hane

CHAR: Sabit uzunluklu karakter dizgilerini tanımlamak için yapılabilir. En fazla 255 karakter olabilir. Uzunluk tanımlanmazsa 1 olarak alınır.

ADI CHAR(10)

VARCHAR2: Değişken uzunluklu karakter dizgilerini tanımlamak için kullanılır. Verinin uzunluğu en büyük değeri verecek şekilde tanımlanır. Uzunluk en fazla 2000 karakter olabilir.

ADRES VARCHAR2(100)

LONG: Değişken uzunluklu karakter dizgilerini tanımlamak için kullanılır. En fazla 2 Gigabyte veri içerebilir. Long veri türünü içeren kolonlar index kolonu olarak kullanılamaz ve bir tabloda en fazla bir kolon LONG olabilir.

RAW/LONG RAW: İkili yada byte dizgiler biçimindeki verileri saklamak için kullanılabilir. RAW ile tanımlanan veriler sadece okunabilir veya saklanabilir. Üzerinde değişiklik yapılamaz.

DATE: Her DATE veri türü ile belirtilen kolon yüzyıl, yıl, ay, gün, saat, dakika, saniye verilerini içerecek biçimde saklanır.

ROWID: Her tablonun satırlarını simgeleyen bir satır tanımlayıcısı bulunmaktadır. Bu tanımlayıcı satırların adreslerini içermektedir. Bu bilgilere dayanarak bazı işlemlerle ilgili satırlara çok hızlı erişme olanağı elde edilebilir.

SQL OPERATÖRLERİ

- Aritmetik operatörler
- Karşılaştırma operatörleri
- Mantıksal operatörler
- Küme operatörleri
- Karakter operatörleri
- Diğer operatörler

Aritmetik Operatörler

- () İşlemleri gruplandırma ve işlem önceliği belirleme
- + Toplama
- Çıkarma
- * Çarpma
- / Bölme

Örnek1: personelin maaşının %10'unu bir kolon halinde görmek için;

```
SELECT NO, ADI, SOYADI, MAAS*0.1 FROM PERSONEL
```

Örnek2: Yukarıdaki örnekte kolonun başlığı 'MAAS*0.1' olur. Kendi başlığımızı görüntülemek için;

```
SELECT NO, ADI, SOYADI, MAAS*1.5 "ZAMLI ÜCRET" FROM PERSONEL
```

Karşılaştırma Operatörleri

(): İşlemleri gruplandırma ve işlem önceliği belirleme.

= : Belirtilen değere eşit olanlar.

> : Belirtilen değerden büyük olanlar.

< : Belirtilen değerden küçük olanlar.

!=, ^=, <> : Belirtilen değere eşit olmayanlar.

>= : Belirtilen değerden büyük veya eşit olanlar.

<= : Belirtilen değerden küçük veya eşit olanlar.

IN: Bir grup değer içinde olanlar.

NOT IN: Bir grup değer içinde olmayanlar.

ANY: Verilen bir koşula uygun olarak, bir grup içindeki değerlerden birini belirler.

ALL: Verilen bir değeri bir liste içindeki tüm değerlerle karşılaştırır.

BETWEEN x AND y: x ve y değeri arasındaki değerleri belirler.

NOT BETWEEN x AND y: x ve y değeri arasında olmayan değerleri belirler. Yani x'den küçük, y'den büyük değerler.

IS NULL: Null (yokluk) değerine sahip olanları tanımlar

IS NOT NULL: Null değerine sahip olmayanları tanımlar.

EXISTS: Kendisine bağlı sorgulama sonucunda en az bir satır bulunabilirse TRUE değerini alır.

LIKE: Belirli bir kalıba uygun olan bilgileri tanımlar.

Örnek1: personel içinde adı “S” ile başlayanlar

```
SELECT NO,ADI,SOYADI FROM PERSONEL WHERE ADI LIKE ‘S%’;
```

Örnek2: maaşı 70 ile 120 milyon arası olanlar

```
SELECT NO,ADI,SOYADI FROM PERSONEL WHERE MAAS BETWEEN 70000000  
AND 120000000;
```

Mantıksal Operatörler

İki yada daha fazla ifade arasında kullanılırlar. Sonucun doğru yada yanlış olmasına göre işlemler yapılır.

(): İşlemleri gruplandırma ve işlem önceliği belirleme

AND: Her iki kıstasta doğru ise işlem yapılır

OR: Kıstaslardan biri doğru ise işlem yapılır

NOT: Kıstasın yanlış olması durumunda işlem yapılır

Örnek : Maaşı 100 milyondan büyük olanlar ve 10 numaralı bölümde çalışanlar

```
SELECT NO,ADI,SOYADI FROM PERSONEL WHERE CALISTIGI_BOLUM=’10’ AND  
MAAS>100000000;
```

Küme Operatörleri

UNION: Verilen koşul ifadesine uygun olarak, çift kayıtları gözardı ederek her iki kümedeki tüm kayıtları seçer.

UNION ALL: Verilen koşul ifadesine uygun olarak, çift kayıtlarda dahil tüm kayıtları seçer.

INTERSECT: Sadece her iki koşulda uyan kayıtları seçer.

MINUS: Birinci kümede olan, ikinci kümede olmayan kayıtları seçer.

Örnek1: Soyadı “D”, adı “E” ile başlayan personel

```
SELECT NO,ADI,SOYADI FROM PERSONEL WHERE ADI LIKE ‘E%’
```

UNION

```
SELECT NO,ADI,SOYADI FROM PERSONEL WHERE SOYADI LIKE ‘D%’;
```

Örnek2: Soyadı “D ile başlayan, ancak adı “E” ile başlamayan personel

```
SELECT NO,ADI,SOYADI FROM PERSONEL WHERE ADI LIKE ‘E%’
```

MINUS

```
SELECT NO,ADI,SOYADI FROM PERSONEL WHERE SOYADI LIKE ‘D%’;
```

Karakter Operatörleri

!! : Tanımlanmış olan iki alanın tek bir alan bilgisi şeklinde birleştirerek işlem yapılmasını sağlar

Örnek: SELECT ADI!!SOYADI “ADI SOYADI” FROM PERSONEL;

SQL Fonksiyonları

AVG: Belirlenen bir alan içerisindeki verilerin aritmetik ortalamasını alır.

SUM: Belirlenen alandaki verilerin toplanmasını sağlar.

MAX: Belirlenen alandaki en büyük değerin bulunmasını sağlar.

MIN: Belirlenen alandaki en küçük değerin bulunmasını sağlar.

ROUND: Belirlenen alandaki sayının virgülden sonraki kısmının yuvarlatılmasını sağlar.

SQRT: Belirlenen alandaki değerin karekökünün alınmasını sağlar.

SUBSTR: Belirlenen alandaki değerin istenilen karakterleri seçilebilir.

Örnek1: En düşük maaş;

SELECT MIN(MAAS) FROM PERSONEL;

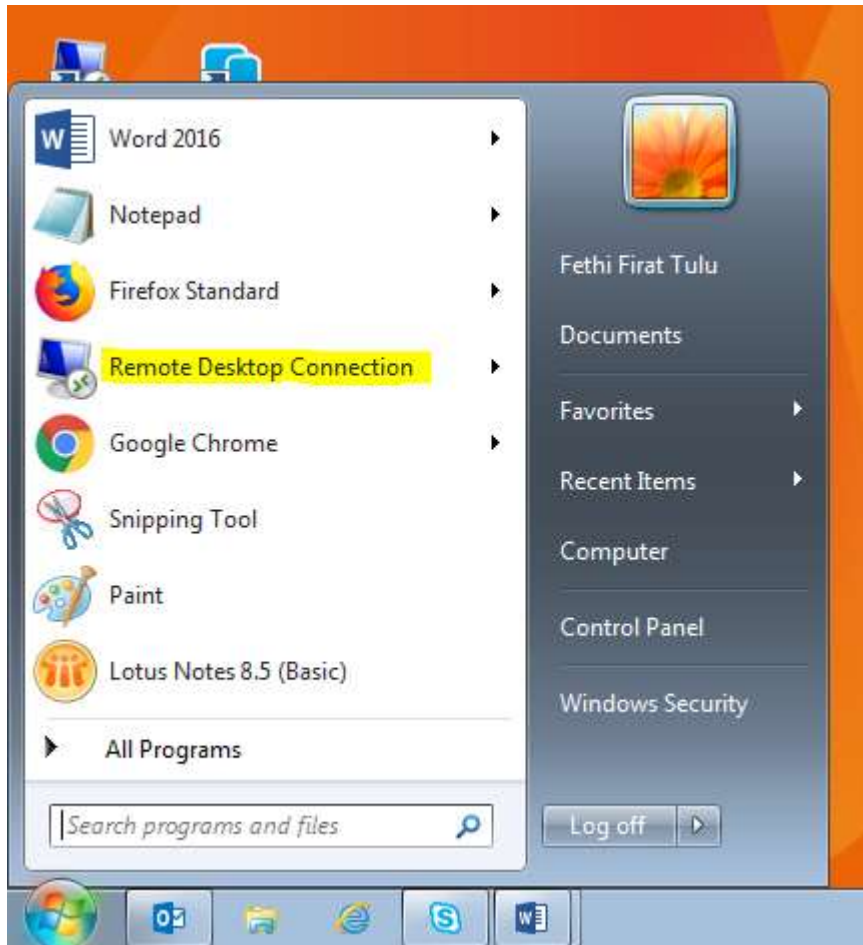
Örnek2: Ortalama maaş;

SELECT AVG(MAAS) FROM PERSONEL;

13.08.2018

Bu haftanın başından itibaren Ayla Hanım bana projeleri görebilme ve developer machine kullanabilme yetkileri verdi. Artık projeleri görebilir ve MS Visual Studio kullanabilirdim. Citrix ile bilgisayara bağlandıktan sonra “Remote Desktop Connection” kullanarak developer machine bağlandım. Ben MS Visual Studio yapısını inceler ve araştırırken Ayla Hanım, Elifnur Hanım ve yazılım ekibi benim ne gibi bir proje yapabileceğimi aralarında kararlaştırdılar.

Remote Desktop Connection Kullanımı



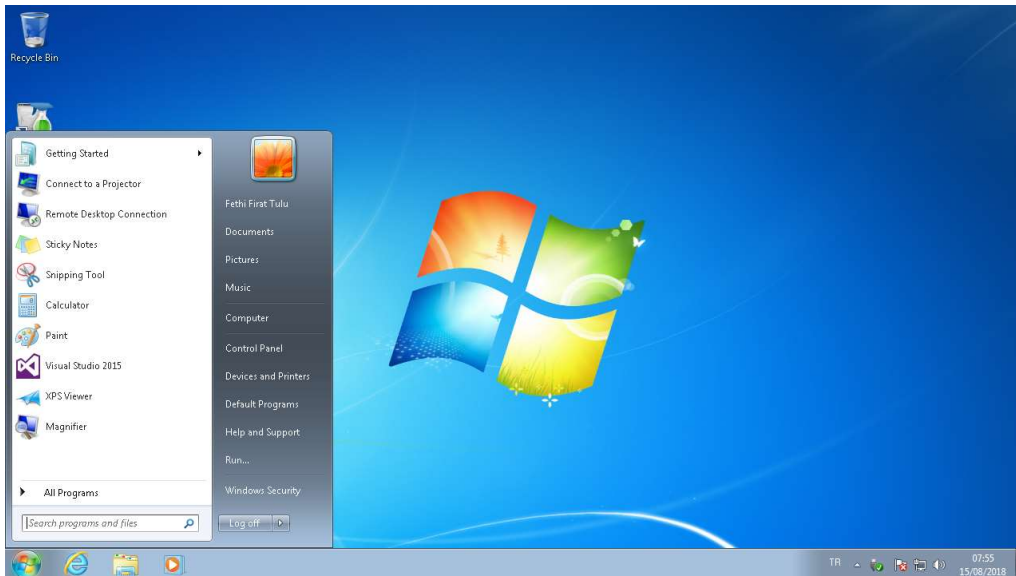
Başlat menüsünden kolayca erişim sağladığım Remote Desktop Connection programı ile daha önceden bana tanımlanmış olan yazılım ekibinin kullandığı işletim sistemine dahil olabilirim.



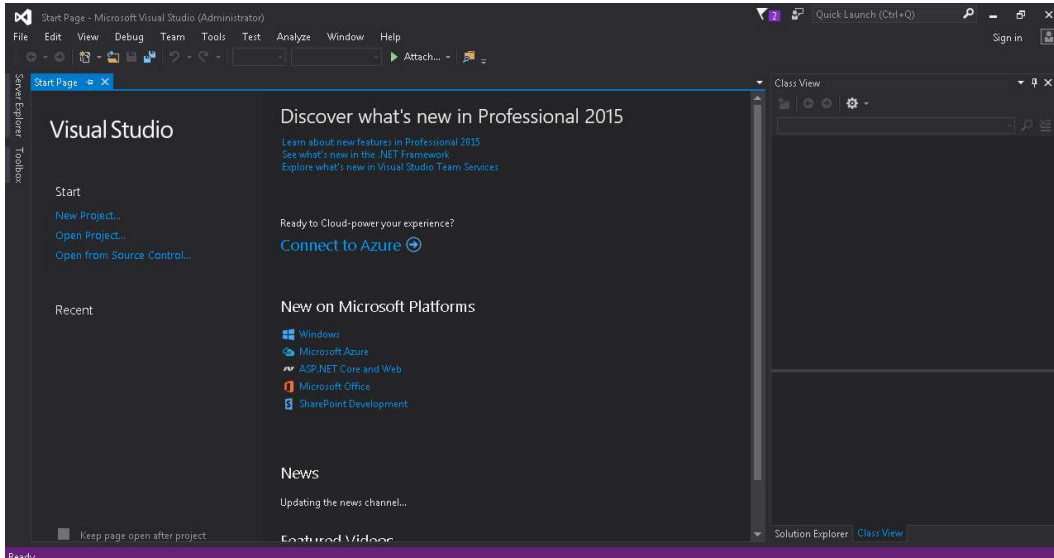
Connect dedikten sonra kendi LeasePlan hesabımın şifresini isteyecektir.



Girip **OK** dedikten sonra erişim sağlayabilirim. Artık tüm yazılım ekibinin erişebildiği program ve projelere erişebilirim.



Microsoft Visual Studio 2015 Yapısı



Resimde Visual Studio açılış ekranını görüyoruz. Start başlığının altından New Project tıkladığımızda yeni bir proje açılacaktır. Open Project oluşturulan projelerden seçim yaparak istenilen proje kolaylıkla açılabilir. Üst menüde bulunan FILE başlığı da yeni proje açma ya da daha önce oluşturulmuş bir projeyi bulma konusunda yardımcı olacaktır. Recent başlığı altında en son açılan projeden başlayarak açılan projeler sıralanıyor. Bu yazılan projelere ulaşmayı kolaylaştırıyor.

Visual Studio'da en üstte komut ve pencerelere erişimi kolaylaştırmak için bir menü bulunmakta. Bu menüde bulunan başlıkları inceleyelim.

File: Üzerinde çalışılan proje kayıt edilebilir ya da kapatılabilir, yeni proje açılabilir.

Edit: Solution Explorer, Server Explorer, Object Browser gibi geliştiricinin ihtiyaç duyacağı pencereler açılabilir.

Project: Projeye yeni kod dosyaları eklenebilir, referans kütüphane yada servis eklenebilir.

Build: Projelerin derlenmesi için kullanılır. Kod yazarken sık sık kodun derlenmesi ve varsa hataların incelenmesi için faydalıdır.

Debug : Kesme işareti (breakpoint) koyulabilir, debug edilebilir. Programların satır satır ilerlenerek incelenmesi için kullanılabilir. Lokal değişkenlerin belli bir anda çalışma zamanındaki durumunu anlamak için faydalıdır.

Data: Bir veri kaynağına bağlanmak için kullanılır. Örneğin bir sql veritabanındaki tablolar ve verileri incelenebilir.

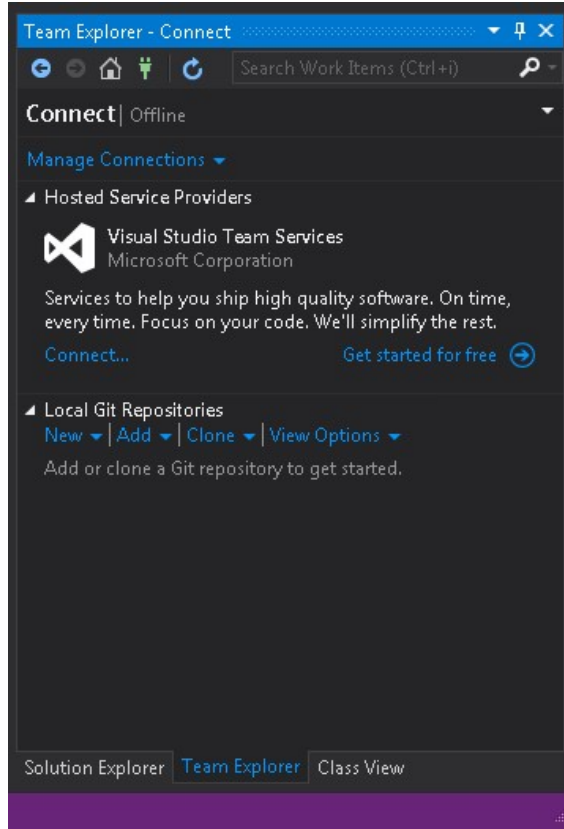
Tools: Visual Studio'nun görsel ayarları yapılabilir.

Window: Visual Studio ekranını bölerek kullanmak, tablo görünümünü değiştirmek için kullanılır.

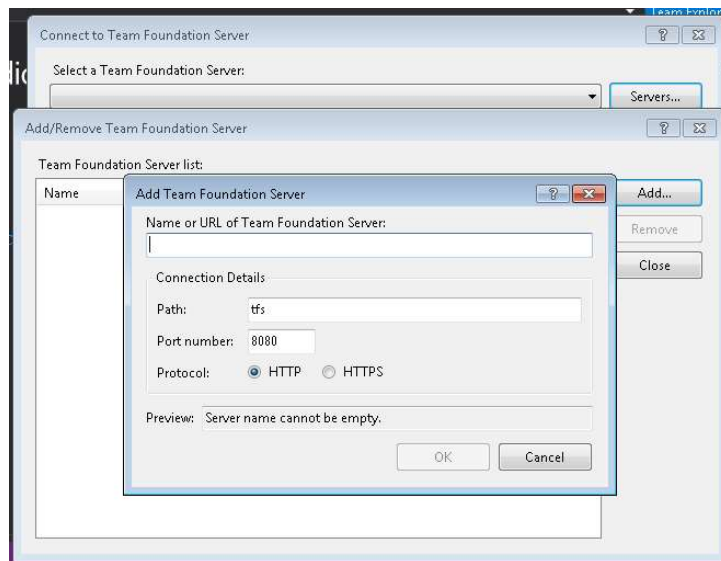
Help : Geniş MSDN (Microsoft Developer Network) kaynakları içerisinde yardım aramak için, Visual Studio'nun versiyonunu öğrenmek için kullanılabilir.

Team Foundation Server (TFS) 'a Bağlanmak

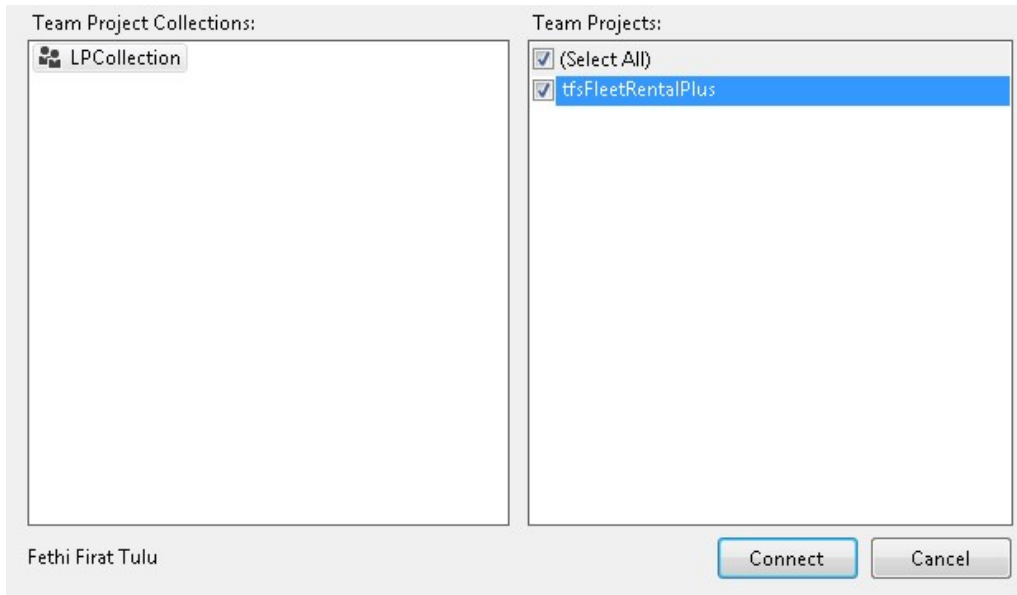
Visual Studio üzerinden yazılım ekibinin projelerine ulaşmak için TFS'e bağlanmamız gerekiyor. Bu yüzden öncelikle açılış ekranında **Start** 'ın altında bulunan **Open From Source Control** seçeneğine tıklıyoruz.



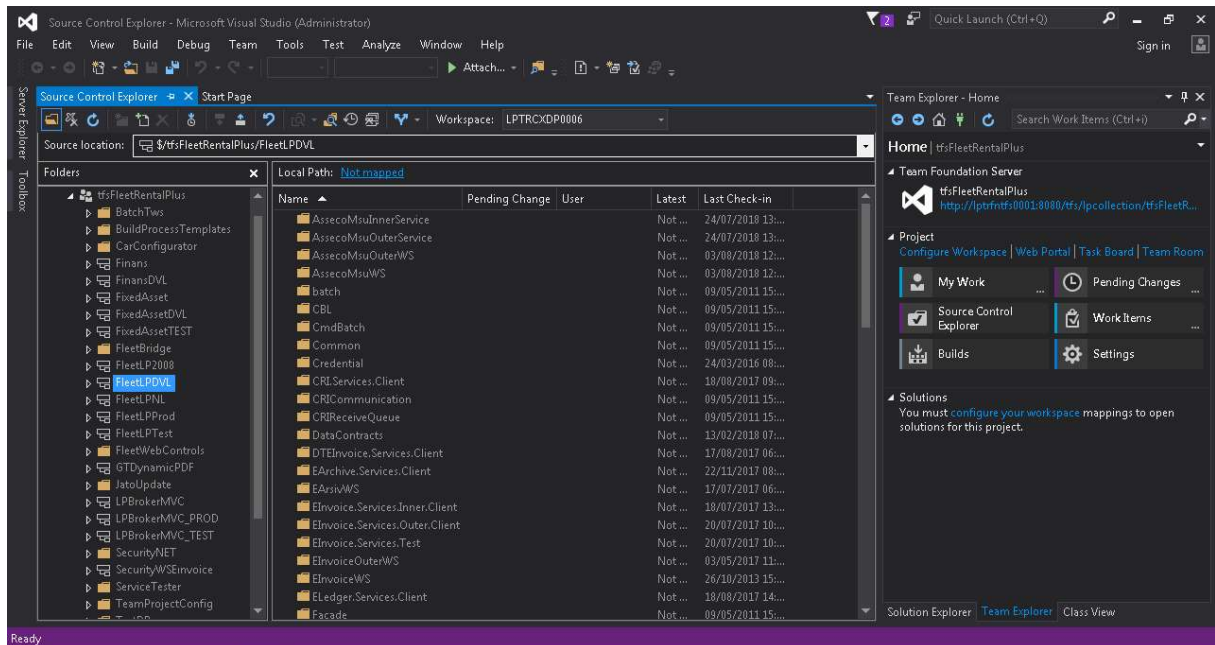
Ekranın sağında Connect bölümü açıldığını görüyoruz yine bu menüden **Visual Studio Team Services** 'in altında **Connect** 'e tıklıyoruz.



Açılan pencerede önce **Servers** sonra da **Add** diyoruz. **Add Team Foundation Server** penceresi açıldıktan sonra Ayla Hanım'ın benim için yetki verdiği server ismini yazıp **OK** tuşuna basarak ekliyoruz.



Artık yazılım ekibinin projesi olan **tfsFleetRentalPlus** 'a erişmeye hazırız. **Connect** diyoruz.



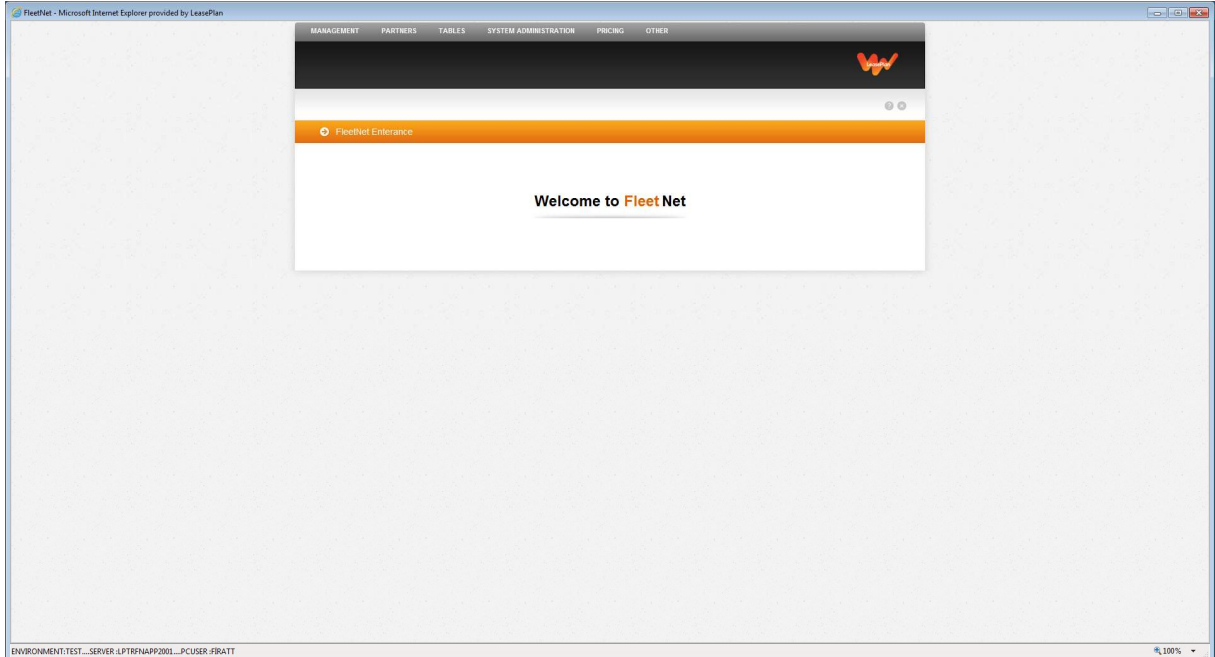
FleetNet

FleetNet tüm süreçleri ve fonksiyonları şirket genelinde entegre eden LeasePlan Türkiye'nin temel uygulamasıdır. Tüm süreçlerde etkili olan bir dizi yerleşik modüle sahiptir.

Tüm şirket çalışanları işlerini bu uygulama üzerinden hallederler. Yazılım ekibi ise bu uygulamanın hem geliştirilmesi hem de bakımından sorumludur.

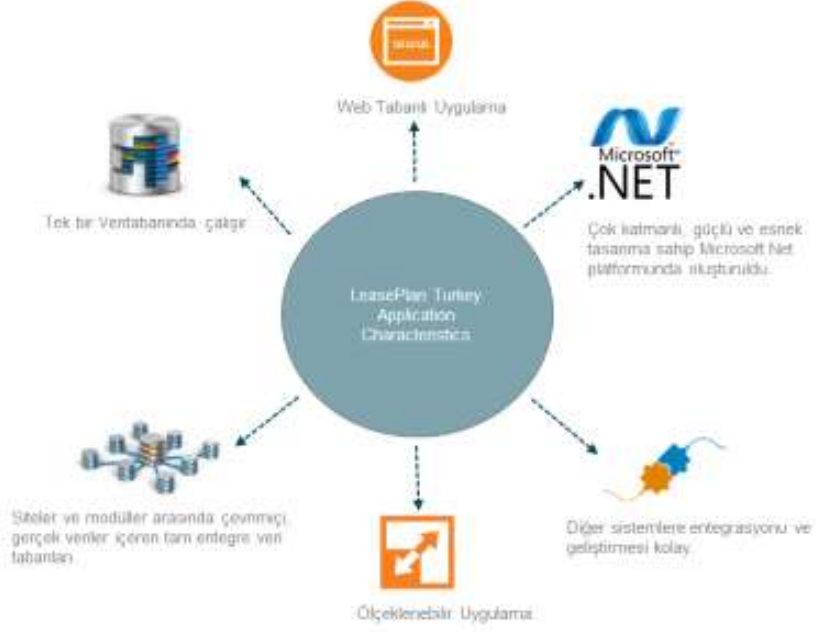


Kullanıcı adı ve şifre oluşturulduktan sonra FleetNet sisteminin test ortamında girişime ve erişimime izin verildi.



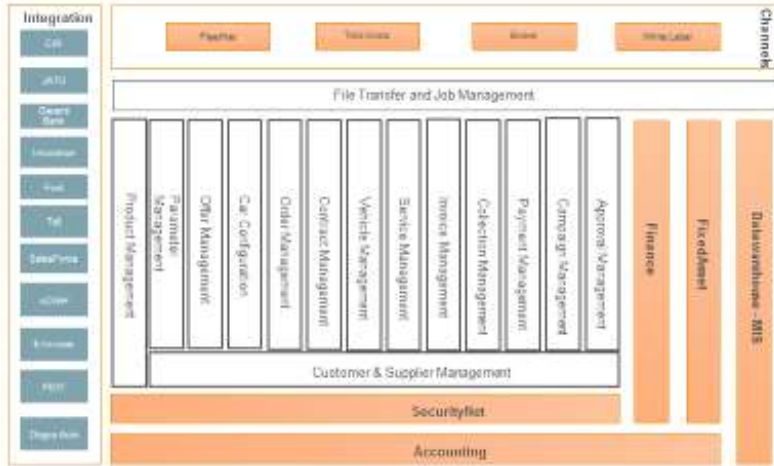
FleetNet Temel Özellikleri:

FleetNet Temel Özellikleri

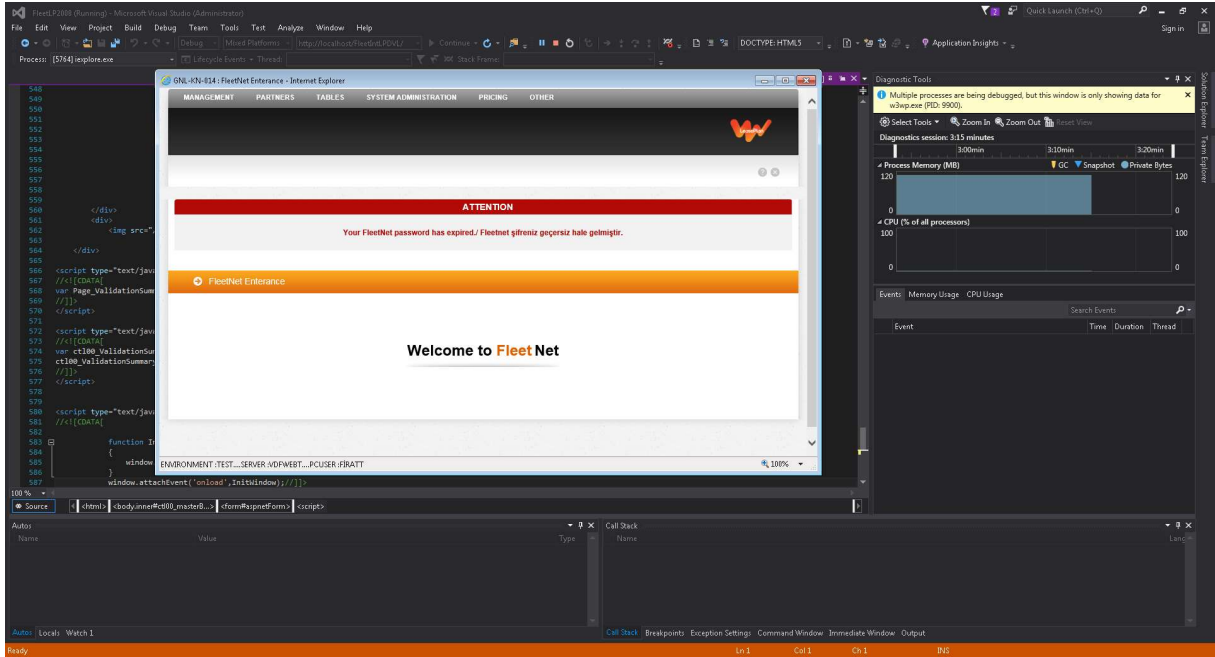


FleetNet Mimarisi:

FleetNet Application Architecture



Start düğmesine bastıktan sonra FleetNet'e test ortamında erişimimiz tamam olacaktır.



TFS kurulumu ve FleetNet incelemesinden sonra Visual Studio ile FleetNet 'i birbine bağladım. Yani Visual Studio da FleetNet'i test ortamında çalıştırabilirim artık. Bundan sonra Ayla Hanım, Elif Nur Hanım ve geliştiricilerin bana proje önermesini bekleyeceğim.

15.08.2018

Ayla Hanım ve diğerleri proje konusunda karar aldılar ve Jira da açılan taleplerden “Incoming Invoice Approval List Olarak Alma” projesini bana uygun gördüler.

The screenshot shows a Jira issue page for 'Incoming Invoice Approval List Rapor Olarak Alma' (LPTRSP-993). The issue is a Story with a Minor priority, Unresolved status, and no resolution or fix version. The reporter is Naz Günsür. The issue was created on 06/Apr/18 6:17 AM and updated on 06/Apr/18 6:59 AM. The description asks if the data from the FleetNet Incoming Invoice Approval List can be exported as an Excel file. A small screenshot of the FleetNet interface is included in the description. The 'People' section shows the issue is unassigned, and the 'Dates' section shows the creation and update times. The 'Development' and 'Agile' sections are also visible.

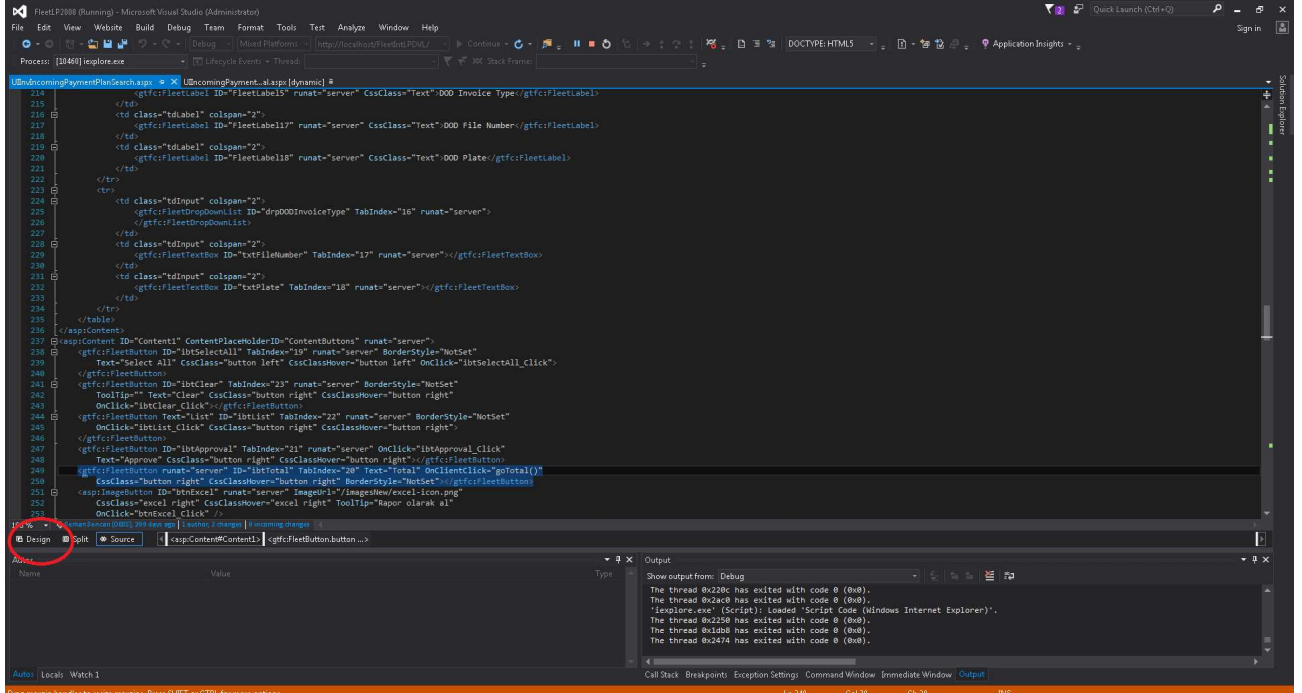
Yukarıda ki ekran görüntüsünde görüldüğü üzere kullanıcılardan talep oluşturma yetkisi bulunan Naz Hanım FleetNet'teki Incoming Invoice Approval List ekranındaki kayıtları Excel dosyası olarak almak istiyor. Bende bu talep doğrultusundaki geliştirmeyi üstlenmiş bulunuyorum. Önümüzdeki günlerin tamamında bu projeye odaklanmayı planlıyorum.

Management > Approval > Incoming Invoice Approval														
Incoming Invoice Approval														
Invoice Owner ID	Invoice Owner Name		<input type="radio"/> E-Invoice <input type="radio"/> Printed Invoice											
Invoice Date From	Invoice Date To	LP.Date From	LP.Date To	Accounting Date From	Accounting Date To									
06.01.2018	06.04.2018													
Approval Status	Invoice Status	InvoiceType	Invoice No											
Please Select	Please Select	Please Select												
Approval User	E-Invoice Profile Type													
burcuse	Please Select													
			Clear			Reject			Approve			List		
Incoming Invoice Approval List											Total 44 Records / 5 Page			
Inv.Owner	Inv.No	Inv.Date	Accounting Date	L.P.Date	Last Amount	Event Cost	Inv.Status	E-Inv. Profile Type	App. Users	App. Status				
ERMAT MOTORLU ARAÇLAR VE	GID201800000108	09.03.2018	09.03.2018	09.04.2018	22.420,00	22.420,00	Assessment	TICARIFATURA	PDF burcuse	Approved				
ERMAT MOTORLU ARAÇLAR VE	GID201800000109	09.03.2018	09.03.2018	09.04.2018	12.390,00	12.390,00	Assessment	TICARIFATURA	PDF burcuse	Approved				
NAZAR OTOMOBİL CLİK YDK	IN02018000000504	09.03.2018	09.03.2018	09.04.2018	9.440,00	9.440,00	Approved	TICARIFATURA	PDF burcuse	Approved				
KIYI OTOMOTİV SAN. VE Tİ	KY02018000001115	06.03.2018	06.03.2018	05.04.2018	11.210,00	11.210,00	Approved	TICARIFATURA	PDF burcuse	Approved				
OKAR OTOMOTİV P AZARLAMA	OKR201800000148	05.03.2018	05.03.2018	19.04.2018	118,00	118,00	Approved	TICARIFATURA	PDF burcuse	Approved				
KEMAL TEPRETOĞULLARI OTO	KTS2018000002627	02.03.2018	02.03.2018	02.04.2018	3.293,83	3.293,83	Approved	TEMELFATURA	PDF burcuse	Approved				
Kuzey Otomotiv ve Diş Ti	KZ12018000000907	01.03.2018	01.03.2018	16.04.2018	5.248,05	5.248,05	Approved	TICARIFATURA	PDF burcuse	Approved				
K.TEPRETOĞULLARI OTOMOTİ	NTO2018000000683	28.02.2018	01.03.2018	16.04.2018	236,00	236,00	Approved	TICARIFATURA	PDF burcuse	Approved				
ARTKIY OTOMOTİV İNŞAAT A	ARP2018000000251	28.02.2018	01.03.2018	30.03.2018	9.812,71	9.812,71	Approved	TEMELFATURA	PDF burcuse	Approved				
SERTEPE OTOMOTİV İNŞAAT	SRT2018000001344	28.02.2018	01.03.2018	30.03.2018	118,00	118,00	Approved	TEMELFATURA	PDF burcuse	Approved				

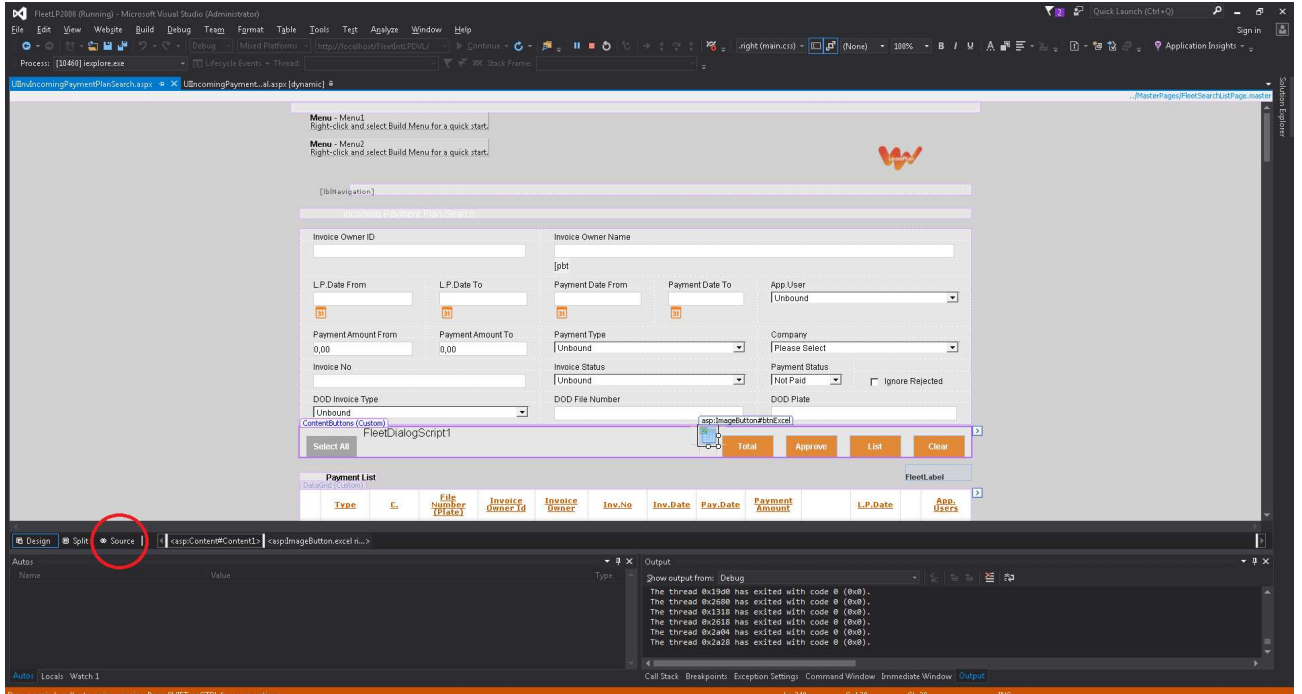
Projenin taslağı olarak yukarıda ki ekran görüntüsünde işaretlenmiş boşluğa, istenilen verilerin Excel dosyası olarak çekilmesi için bir simge koymayı planlıyorum.

Kullanıcı simgeye tıkladığında istenilen veri Excel dosyası olarak bilgisayarına indirecek. Geliştirmemi bu yönde yapmayı planlıyorum.

Daha sonra MS Visual Studio da” UIInvIncomingPaymentPlanSearch” sayfasının kodlarına erişiyorum.



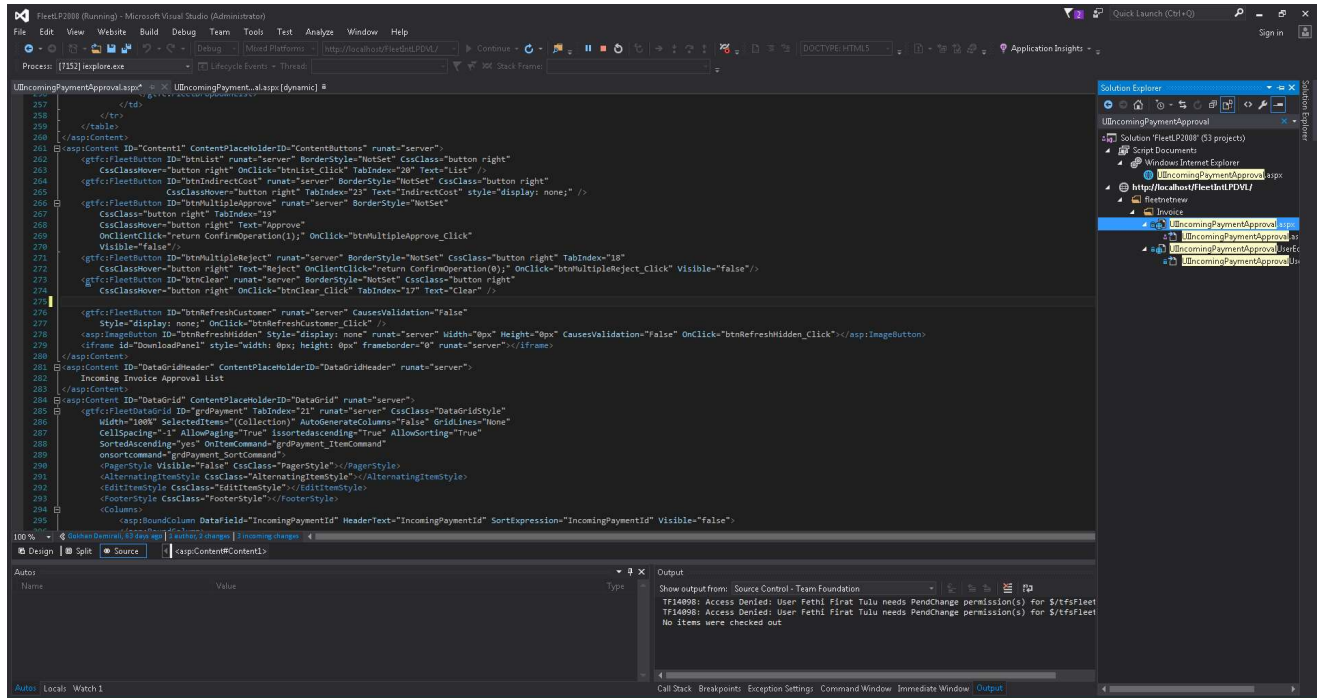
Üstte ki fotoğrafta belirttiğim gibi “Design” kısmına basarak, sayfanın tasarımının bulunduğu ekrana geçiyorum.



Excel butonunu seçerek “Source” sekmesine tıklıyorum bu sayede. Son katman da bulunan Excel butonunun kodlarını görmüş olacağım.

Kodları yeterince inceledikten sonra, benim çalışacağım sayfa olan Incoming Invoice Approval sayfasına sağ tıklayarak, “UIIncomingPaymentApproval” kaynağına ulaşıyorum. Aşağıda gördüğümüz üzere Microsoft Visual Studio da arama bölümüne yine bu adı yazarak sayfanın kaynak kodlarına erişmiş bulunuyorum.

Açılan bu kısım sayfanın ilk katmanı olan arayüz kısmına ait. Görüldüğü üzere sayfa da bulunan butonların yer aldığı kodlar burada. Şimdi buraya buton olarak Excel fonksiyonu ekleyelim.



```
257 </td>
258 </tr>
259 </table>
260 </asp:Content>
261 <asp:Content ID="Content1" ContentPlaceHolderID="ContentButtons" runat="server">
262 <tf:FleetButton ID="btnList" runat="server" BorderStyle="None" CssClass="button right"
263 CssClassHover="button right" OnClick="btnList_Click" TabIndex="20" Text="List" />
264 <tf:FleetButton ID="btnIndirectCost" runat="server" BorderStyle="None" CssClass="button right"
265 CssClassHover="button right" TabIndex="23" Text="IndirectCost" style="display: none;" />
266 <tf:FleetButton ID="btnMultipleApprove" runat="server" BorderStyle="None"
267 CssClass="button right" TabIndex="19"
268 CssClassHover="button right" Text="Approve"
269 OnClientClick="return ConfirmOperation(1); OnClick='btnMultipleApprove_Click'
270 Visible='false' />
271 <tf:FleetButton ID="btnMultipleReject" runat="server" BorderStyle="None" CssClass="button right" TabIndex="18"
272 CssClassHover="button right" Text="Reject" OnClientClick="return ConfirmOperation(0); OnClick='btnMultipleReject_Click' Visible='false' />
273 <tf:FleetButton ID="btnClear" runat="server" BorderStyle="None" CssClass="button right"
274 CssClassHover="button right" OnClick="btnClear_Click" TabIndex="17" Text="Clear" />
275
276 <tf:FleetButton ID="btnRefreshCustomer" runat="server" CausesValidation="False"
277 Style="display: none;" OnClick="btnRefreshCustomer_Click" />
278 <asp:ImageButton ID="btnRefreshHidden" Style="display: none" runat="server" Width="30px" Height="30px" CausesValidation="False" OnClick="btnRefreshHidden_Click" />
279 <iframe id="DownloadExcel" style="width: 0px; height: 0px; frameborder="0" runat="server" />
280 </asp:Content>
281 <asp:Content ID="DataGridHeader" ContentPlaceHolderID="DataGridHeader" runat="server">
282 Incoming Invoice Approval list
283 </asp:Content>
284 <asp:Content ID="DataGrid" ContentPlaceHolderID="DataGrid" runat="server">
285 <tf:FleetDataGrid ID="gridPayment" TabIndex="21" runat="server" CssClass="DataGridStyle"
286 Width="100%" SelectedItems="(collection)" AutoGenerateColumns="False" GridLines="None"
287 CellSpacing="1" AllowWagging="True" Isorteddescending="True" AllowSorting="True"
288 Sortdescending="yes" OnItemClick="gridPayment_ItemCommand"
289 onsortcommand="gridPayment_SortCommand" />
290 <PagerStyle Visible="False" CssClass="PagerStyle" />
291 <AlternatingItemStyle CssClass="AlternatingItemStyle" />
292 <EditItemStyle CssClass="EditItemStyle" />
293 <FooterStyle CssClass="FooterStyle" />
294 <Columns>
295 <asp:BoundColumn DataField="IncomingPaymentId" HeaderText="IncomingPaymentId" SortExpression="IncomingPaymentId" Visible="False">
```

Araştırıp öğrendiğim gibi buraya;

```
<asp:ImageButton ID="btnExcel" runat="server" ImageUrl="/imagesNew/excel-icon.png"
```

```
CssClass="excel right" CssClassHover="excel right" ToolTip="Rapor olarak al"
```

```
OnClick="btnExcel_Click" />
```

kodunu ekliyorum.

Bu kod daha önce “btnExcel” adında oluşturulmuş olan bir fonksiyonu çağırarak, gerekli bilgileri nereden alması gerektiğini ve neleri kullanması gerektiğini gösteriyor.

Örneğin “ToolTip” fonksiyonu Mouse ile butonun üzerine geldiğinde bilgilendirme olarak yazılan yazıyı gösteriyor onun dışında butonun simgesinin nereden gelmesi gerektiğini ImageUrl olarak gösterebiliyoruz.

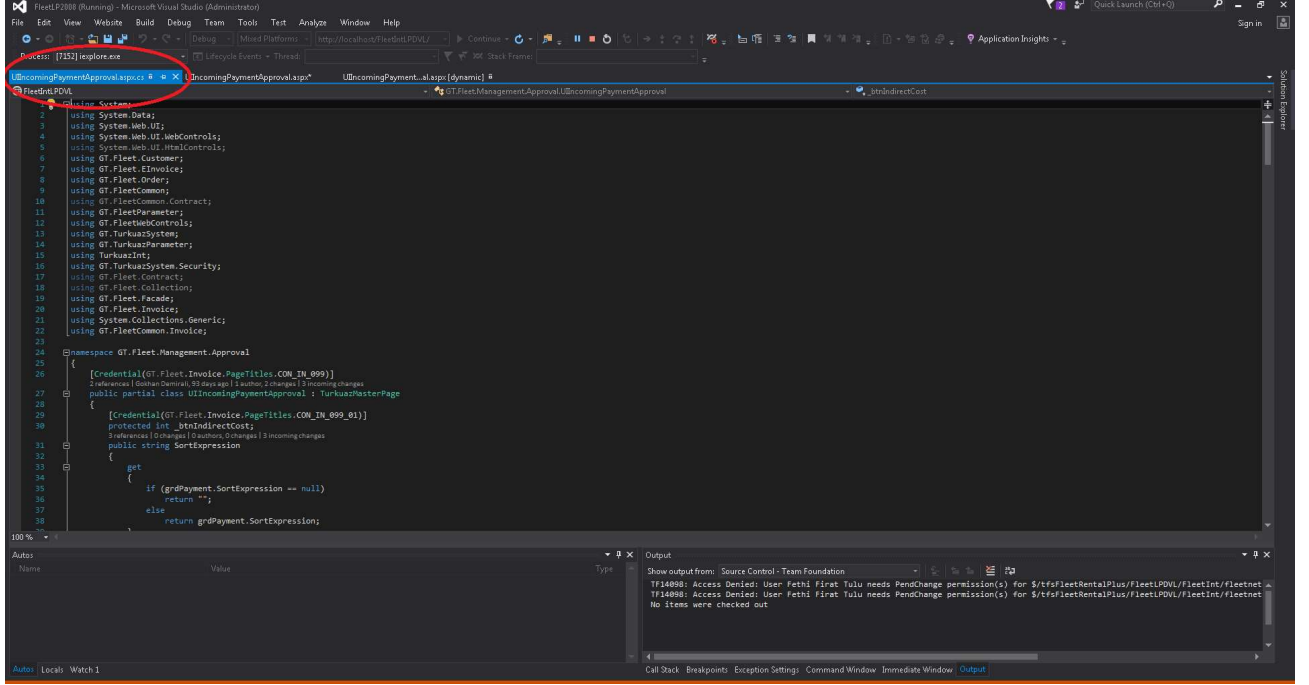
```
214 <td class="tdInput" colspan="2">
215 <gtfci:FleetDropDownList ID="grpInvoiceProfileType" TabIndex="16" runat="server" CssClass="Text">
216 </gtfci:FleetDropDownList>
217 </td>
218 </tr>
219 </table>
220 </asp:Content>
221 <asp:Content ID="Content1" ContentPlaceHolderID="ContentButtons" runat="server">
222 <gtfci:FleetButton ID="btnList" runat="server" BorderStyle="None" CssClass="button right"
223 CssClassHover="button right" OnClick="btnList_Click" TabIndex="20" Text="List" />
224 <gtfci:FleetButton ID="btnIndirectCost" runat="server" BorderStyle="None" CssClass="button right"
225 CssClassHover="button right" TabIndex="23" Text="IndirectCost" style="display: none;" />
226 <gtfci:FleetButton ID="btnMultipleApprove" runat="server" BorderStyle="None"
227 CssClass="button right" TabIndex="21" Text="Approve"
228 OnClientClick="return ConfirmOperation(1);" OnClick="btnMultipleApprove_Click"
229 Visible="false" />
230 <gtfci:FleetButton ID="btnMultipleReject" runat="server" BorderStyle="None" CssClass="button right" TabIndex="18"
231 CssClassHover="button right" Text="Reject" OnClientClick="return ConfirmOperation(0);" OnClick="btnMultipleReject_Click" Visible="false" />
232 <gtfci:FleetButton ID="btnClear" runat="server" BorderStyle="None" CssClass="button right"
233 CssClassHover="button right" OnClick="btnClear_Click" TabIndex="17" Text="Clear" />
234 <asp:ImageButton ID="btnExcel" runat="server" ImageUrl="/images/new/excel-icon.png"
235 CssClass="excel right" CssClassHover="excel right" ToolTip="Rapor olarak al"
236 OnClick="btnExcel_Click" />
237 <gtfci:FleetButton ID="btnRefreshCustomer" runat="server" CausesValidation="false"
238 Style="display: none;" OnClick="btnRefreshCustomer_Click" />
239 <asp:Image ID="btnRefreshHidden" Style="display: none;" runat="server" Width="0px" Height="0px" CausesValidation="false" OnClick="btnRefreshHidden_Click" />
240 <iframe id="DownloadPanel" style="width: 0px; height: 0px; frameborder="0" runat="server" />
241 </asp:Content>
242 <asp:Content ID="DataGridHeader" ContentPlaceHolderID="DataGridHeader" runat="server">
243 Incoming Invoice Approval List
244 </asp:Content>
245 <asp:Content ID="DataGrid" ContentPlaceHolderID="DataGrid" runat="server">
246 <gtfci:FleetDataGrid ID="grdPayment" TabIndex="21" runat="server" CssClass="DataGridStyle"
247 Width="100%" SelectedItems="(collection)" AutoGenerateColumns="false" GridLines="None"
248 CellSpacing="1" AllowPaging="true" IsSortedAscending="true" AllowSorting="true"
249 SortedAscending="yes" OnItemCommand="grdPayment_ItemCommand"
250 onsortcommand="grdPayment_SortCommand">
251 <PagerStyle Visible="false" CssClass="PagerStyle" />
252 <AlternatingItemStyle CssClass="AlternatingItemStyle" />
253 </gtfci:FleetDataGrid>
254 </asp:Content>
255 </div>
256 </div>
```

Yukarı da gösterildiği gibi kodu yazdıktan sonra “Desing” e tıklıyoruz.

Inv. Owner	Inv.No	Inv.Date	Accounting Date	L.P.Date	Last Amount	Event Cost	Inv.Status	E-Inv. Profile Type	App. Users	App. Status	Indirect Cost			
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	PDF	Databound	Databound	✓	✗	🕒
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	PDF	Databound	Databound	✓	✗	🕒
Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	PDF	Databound	Databound	✓	✗	🕒

İşte Excel butonu şuanda görünürde.

Arayüze butonu ekledikten sonra diğer katmanlarda çalışması gereken kodları yazmam gerekiyor. Kodlara geri gelip “F7” tuşuna basarak alt tarafta çalışan kaynak kodların olduğu .cs uzantılı dosyaya gidiyorum.



Şimdi burada arayüzde eklemiş olduğumuz butonun üstüne tıkladığımız da çalışması gereken kodları yazmalıyım. Bunu “OnClick=btnExcel_Click” kodunda yazdığım gibi; “btnExcel_Click” fonksiyonu ile yapmalıyım. Butona tıkladığımızda listelediğimiz bilgiler bir Excel dosyası olarak bilgisayarımıza indirilmeli.

Diğer click fonksiyonlarının arasına yeni bir “btnExcel_Click” fonksiyonu ekliyorum. Bunun içinde bu kodu kullanıyorum;

```
protected void btnExcel_Click(object sender, ImageClickEventArgs e)
```

```
{  
  
    try  
  
    {  
  
        DataTable dt = new DataTable();  
        if (dt != null && dt.Rows.Count > 0)  
  
        {
```

```

        Session.Add("CSVDataTable", dt);

        Session.Add("CSVDataGridHeader", grdPayment.Columns);

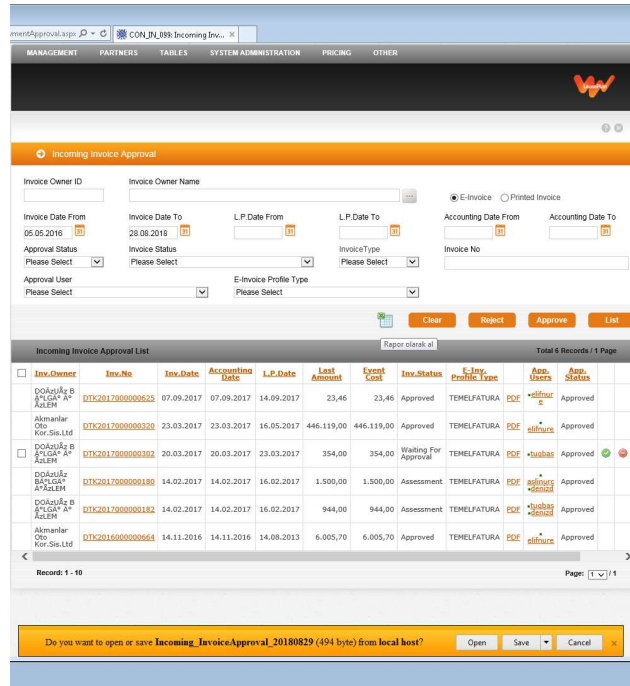
        string url = string.Format(Request.ApplicationPath +
"/Admin/UICSVDownload.aspx?P1=Incoming_InvoiceApproval_{0:yyyyMMdd}",
DateTime.Today);

        Response.Redirect(url,false);
    }
    else
    {
        UserMessage = GetUserMessage(ConstUserMessages_GNL.RecordNotFound);
    }
}
catch (Exception ex)
{
    UserMessage = GetUserMessage(ConstUserMessages_ERR.Generic);
    ExceptionManager.Publish(ex);
}
}

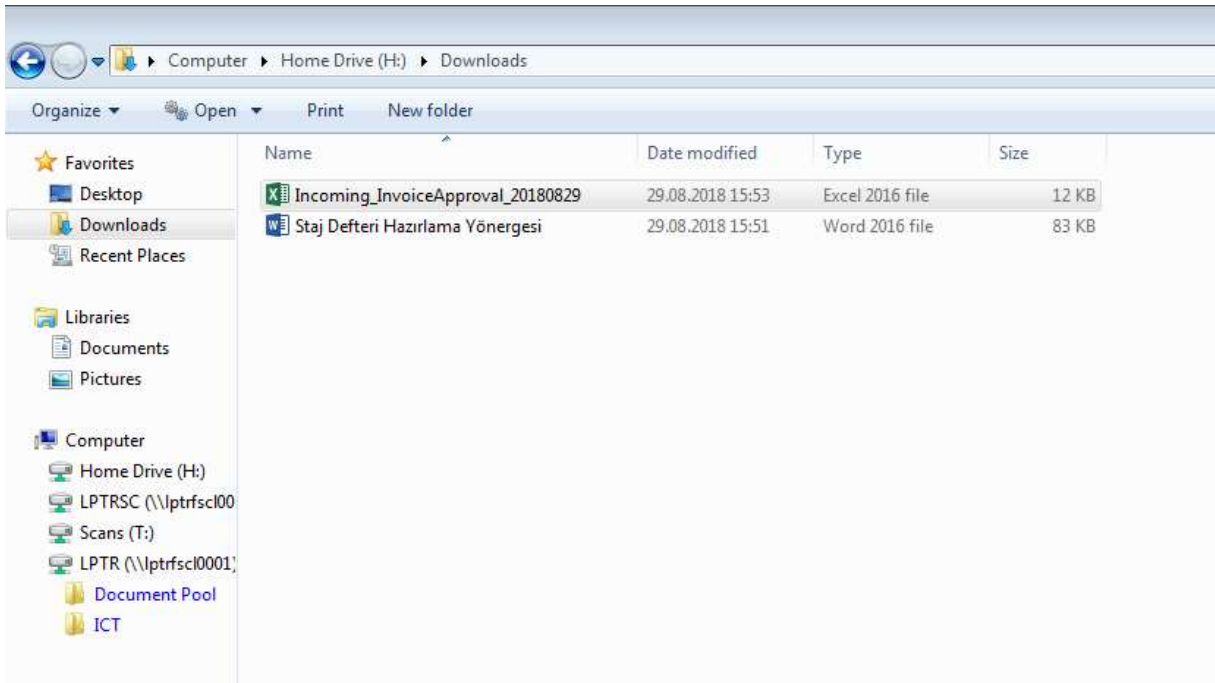
```

Bu arada session'lara ekliyorum ki, kod farklı bir şeyle karşılaşmadığı sürece kaydetsin tekrar tekrar çalıştırıp okumaya gerek kalmasin. Ayrıca string url olarak verdiğim indirme linkinin yanında, dosyaya özgün bir isim vermek için DateTime.Today ekliyorum. Çünkü her indirmede o günün tarihini versin, yenilenen listelerde o günün tarihi dosya ismine geçsin istediğim için.

Bir önce ki ekran görüntüsünde gördüğümüz gibi, kodlarımız çalışmış ve buton eklenmiş. Şimdi son olarak tıkladığımızda her hangi bir sorunla karşılaşmamak adına butona tıklıyoruz.



Internet Explorer da indirme uyarısı açıldı. Save dedikten sonra dosyamız indirilmiş vaziyette ve kullanıma hazır.



6.1.5. PROJENİN DEĞERLENDİRİLMESİ

Bana uygun görülen bu proje; kullanıcıların taleplerinin ne kadar önemli olduğu, geliştiricilerin bu durumu en kolay ve en kısa şekilde nasıl halledebileceğini gösterdi bana. Sürekli kullanılan bir sistem ya da bir web sitesi bile anlık oluşan sorunları ya da talepleri, geliştirilmesi gereken çokça özelliği barındırdığını fark etmiş oldum. Microsoft Visual Studio'nun yapısını nasıl hızlı ve kolay kullanabileceğimi öğrendim.

Çok katmanlı yazılım mimarisi nedir? Nasıl kullanılmalıdır? Bu sorunun cevaplarını buldum bu projede. Yazdığım kodların hangi katmanda ne işe yaradığını, neler yaptığını öğrendim. Ayrıca yazdığım kodlarda ki fonksiyonlar, tanımlamalar, değişkenler vs tam olarak ne ifade ediyor bunları öğrendiğimi düşünüyorum.

Sadece kendi yazdığım kodların değil, diğer geliştiricilerin sürekli ve yenilenen kodlarını, geçmişte yazdıklarını tüm bunların ne işe yaradıklarını da araştırma fırsatı buldum.

Ayrıca projeden önce yaptığım araştırmalar ve izlediğim videolarda kendime çok şey kattığımı düşünüyorum. Microsoft.NET, ASP.NET, HTML, C#, SQL konularında bilgi sahibi oldum. Bu anlamda yaptığım stajın ileri ki iş hayatımda ve yazacağım kodlarda bana çok yardımcı olacağına inanıyorum.

7. STAJ DEĞERLENDİRİLMESİ

8. KAYNAKÇA

<https://lportal.sharepoint.com/sites/lptr-Intranet/>

<https://www.citrix.com.tr/products/receiver/>

<https://barisinceisci.wordpress.com/2014/05/17/pgp-universal-server-symantec-encryption-management-server-kurulumu/#more-100>

<http://www.abdullahboztas.net/symantec-endpoint-encryption-nedir-bolum-1/>

<http://www.hakanuzuner.com/index.php/category/active-directory>

<https://bbozkurt.wordpress.com/2012/06/27/symantec-endpoint-protection-genel-bakis/>

<http://elifkkurt.blogspot.com/2015/11/agile-cevik-ve-waterfall-selale-yazlm.html>

<http://muratyucedag.blogspot.com.tr/>

<http://www.kodlamamerkezi.com/>

<https://www.teknologweb.com/>

<http://akademi.teknokroki.com/>