



CELAL BAYAR ÜNİVERSİTESİ
HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ

STAJ DOSYASI

BÖLÜMÜ : Yazılım Mühendisliği
NUMARASI : 172802032
ADI ve SOYADI : İsmail Sefa AKDENİZ
STAJ TÜRÜ : Zorunlu Yaz Stajı



STAJ TÜRÜ: Zorunlu Yaz Stajı

STAJ YAPILAN FİRMA/KURUM

ADI: Vega Grup Yazılım ve Bilişim Teknolojileri TİC. LTD. ŞTİ

ADRESİ: Üniversiteler Mah. 1596.Cadde Hacettepe Teknokent 5.Arge Binası Kat:5 No:19,
06800 Çankaya/Ankara

TEL: 0850 346 94 94

FAKS: 0312 428 14 60

STAJ İLE İLGİLİ FİRMA/KURUM SORUMLUSUNUN

ADI ve SOYADI: Emre ÇINAR

UNVANI: Yazılım Geliştirme Müdürü

ONAY
(Tarih, Kaşe ve İmza)

STAJA BAŞLAMA TARİHİ: 20 / 07 / 2020

STAJIN BİTİŞ TARİHİ: 02 / 09 / 2020

Bu kısım Bölüm Staj Komisyonu tarafından doldurulacaktır.

STAJ YAPILAN GÜN SAYISI: gün

KABUL EDİLEN GÜN SAYISI: gün

ONAY
(Tarih, Kaşe ve İmza)

☐ Başarılı

☐ Başarısız

20 / 07 / 2020 tarihinden 25 / 07 / 2020 tarihine kadar bir haftalık çalışma programı

| GÜNLER | YAPILAN ÇALIŞMALAR | SAYFA NO | SAAT |
|-----------|-----------------------------------------------------------|-----------------|------|
| Pazartesi | Tanışma ve Kullanılan Teknolojiler Hakkında Bilgilendirme | 1 | 8 |
| Salı | Gerekli Kurulumların Yapılması, Arayüz ve Kod İncelenmesi | 2 | 8 |
| Çarşamba | Veritabanı Şemasının İncelenmesi ve Anlaşılması | 3 | 8 |
| Perşembe | Çiçek Sepeti API Dokümanının İncelenmesi | 4 | 8 |
| Cuma | Proje Gereksinim Analizi Yapılması | 5 | 8 |
| Cumartesi | - | | |
| ONAY | Kaşe / İmza | Toplam Saat: 40 | |

27 / 07 / 2020 tarihinden 01 / 08 / 2020 tarihine kadar bir haftalık çalışma programı

| GÜNLER | YAPILAN ÇALIŞMALAR | SAYFA NO | SAAT |
|-----------|-----------------------------------------------------|-----------------|------|
| Pazartesi | Proje Oluşturulması ve Başlangıcı | 6 | 8 |
| Salı | Siparişlerin Aktarılmasının Gerçekleştirimi | 7 | 8 |
| Çarşamba | Siparişlerin Aktarılmasının Testi ve Geliştirilmesi | 8 | 8 |
| Perşembe | Kurban Bayramı Arifesi (Tatil) | | - |
| Cuma | Kurban Bayramı 1.Gün (Tatil) | | - |
| Cumartesi | - | | |
| ONAY | Kaşe / İmza | Toplam Saat: 24 | |

03 / 08 / 2020 tarihinden 08 / 08 / 2020 tarihine kadar bir haftalık çalışma programı

| GÜNLER | YAPILAN ÇALIŞMALAR | SAYFA NO | SAAT |
|-----------|-------------------------------------------------|-----------------|------|
| Pazartesi | Kurban Bayramı 4.Gün (Tatil) | | - |
| Salı | Entegre Projesine Giriş Yetki Sorgusu Yapılması | 9 | 8 |
| Çarşamba | Şefim Ürünlerinin Çiçek Sepetine Aktarımı | 10 | 8 |
| Perşembe | Çiçek Sepetine Aktarım ve Kategori Servisi | 11 | 8 |
| Cuma | Ürüne ait Özelliklerin Kullanıcıdan Alınması | 12 | 8 |
| Cumartesi | - | | |
| ONAY | Kaşe / İmza | Toplam Saat: 32 | |

10 /08 / 2020 tarihinden 15 /08 / 2020 tarihine kadar bir haftalık çalışma programı

| GÜNLER | YAPILAN ÇALIŞMALAR | SAYFA NO | SAAT |
|-----------|----------------------------------------------------|-----------------|------|
| Pazartesi | Kategoriye ait Özelliklerin Elde Edilmesi | 13 | 8 |
| Salı | Kategoriye ait Özelliklerin Kullanıcıdan Alınması | 14 | 8 |
| Çarşamba | Ürün Yükleme Modellerinin Oluşturulması | 15 | 8 |
| Perşembe | Ürün Yükleme Metodunun Kullanılması | 16 | 8 |
| Cuma | Ürün Yükleme Sonrası Batch-Status Metodu Kullanımı | 17 | 8 |
| Cumartesi | | | |
| ONAY | Kaşe / İmza | Toplam Saat: 40 | |

17 /08 / 2020 tarihinden 22 /08 / 2020 tarihine kadar bir haftalık çalışma programı

| GÜNLER | YAPILAN ÇALIŞMALAR | SAYFA NO | SAAT |
|-----------|-----------------------------------------------------------|-----------------|------|
| Pazartesi | Çiçek Sepeti Ürünlerinin Şefime Aktarımı | 18 | 8 |
| Salı | Çiçek Sepeti Ürünlerinin Şefim Veritabanına Kaydı | 19 | 8 |
| Çarşamba | Çiçek Sepeti Ürünlerinin Varyant Özelliklerin Aktarılması | 20 | 8 |
| Perşembe | Çiçek Sepeti Aktarımı Test ve Sunumu | 21 | 8 |
| Cuma | Çiçek Sepeti Ürün Fiyat Güncelleme Başlangıç | 22 | 8 |
| Cumartesi | | | |
| ONAY | Kaşe / İmza | Toplam Saat: 40 | |

24 /08 / 2020 tarihinden 29 /08 / 2020 tarihine kadar bir haftalık çalışma programı

| GÜNLER | YAPILAN ÇALIŞMALAR | SAYFA NO | SAAT |
|-----------|--------------------------------------------------------|-----------------|------|
| Pazartesi | Ürün Fiyatı Güncelleme Ekranı Oluşturulması | 23 | 8 |
| Salı | Ürün Fiyatı Güncelleme Gerçekleştirilmesi ve Testi | 24 | 8 |
| Çarşamba | Sipariş Kargo Süreçlerinin Planlanması ve Analizi | 25 | 8 |
| Perşembe | Kargo Ekranı ve Veri Giriş Alanlarının Oluşturulması | 26 | 8 |
| Cuma | Kargo Ekran Şefim Bağlantısı ve Bilgilerin Getirilmesi | 27 | 8 |
| Cumartesi | | | |
| ONAY | Kaşe / İmza | Toplam Saat: 40 | |

31 / 08 / 2020 tarihinden 04 / 09 / 2020 tarihine kadar bir haftalık çalışma programı

| GÜNLER | YAPILAN ÇALIŞMALAR | SAYFA NO | SAAT |
|-----------|---------------------------------------------------|-----------------|------|
| Pazartesi | Siparişin Kargoya Verildi Bilgisinin Gönderilmesi | 28 | 8 |
| Salı | Projenin Pilot Mağaza Testinin Yapılması | 29 | 8 |
| Çarşamba | Projenin Teslim ve Devir Yapılması | 30 | 8 |
| Perşembe | | | |
| Cuma | | | |
| ONAY | Kaşe / İmza | Toplam Saat: 24 | |

İÇİNDEKİLER

| | Sayfa No |
|----------------------------------------------------------------------------------|-------------|
| Tanışma ve Kullanılan Teknolojiler Hakkında Bilgilendirme | 1 |
| Gerekli Kurulumların Yapılması, Arayüz ve Kod İncelenmesi | 2 |
| Veritabanı Şemasının İncelenmesi ve Anlaşılması | 3 |
| Çiçek Sepeti API Dokümanının İncelenmesi ve Çiçek Sepeti İletişim Kanalına Giriş | 4 |
| Proje Gereksinim Analizi Yapılması | 5 |
| Proje Oluşturulması ve Başlangıcı | 6 |
| Siparişlerin Aktarılmasının Gerçekleştirimi | 7 |
| Siparişlerin Aktarılmasının Testi ve Geliştirilmesi | 8 |
| Entegre Projesine Giriş Yetki Sorgusu Yapılması | 9 |
| Şefim Ürünlerinin Çiçek Sepetine Aktarımı | 10 |
| Çiçek Sepetine Aktarım ve Kategori Servisi | 11 |
| Ürüne ait Özelliklerin Kullanıcıdan Alınması | 12 |
| Kategoriye ait Özelliklerin Elde Edilmesi | 13 |
| Kategoriye ait Özelliklerin Kullanıcıdan Alınması | 14 |
| Ürün Yükleme Modellerinin Oluşturulması | 15 |
| Ürün Yükleme Metodunun Kullanılması | 16 |
| Ürün Yükleme Sonrası Batch-Status Metodu Kullanımı | 17 |
| Çiçek Sepeti Ürünlerinin Şefime Aktarımı | 18 |
| Çiçek Sepeti Ürünlerinin Şefim Veritabanına Kaydı | 19 |
| Çiçek Sepeti Ürünlerinin Varyant Özelliklerin Aktarılması | 20 |
| Çiçek Sepeti Aktarımı Test ve Sunumu | 21 |
| Çiçek Sepeti Ürün Fiyat Güncelleme Başlangıç | 22 |
| Ürün Fiyatı Güncelleme Ekranı Oluşturulması | 23 |
| Ürün Fiyatı Güncelleme Gerçekleştirilmesi ve Testi | 24 |
| Sipariş Kargo Süreçlerinin Planlanması ve Analizi | 25 |
| Kargo Ekranı ve Veri Giriş Alanlarının Oluşturulması | 26 |
| Kargo Ekran Şefim Bağlantısı ve Bilgilerin Getirilmesi | 27 |
| ONAY | Kaşe / İmza |

| | Sayfa No |
|---------------------------------------------------|----------|
| Siparişin Kargoya Verildi Bilgisinin Gönderilmesi | 28 |
| Projenin Pilot Mağaza Testinin Yapılması | 29 |
| Projenin Teslim ve Devir Yapılması | 30 |

| | |
|------|-------------|
| ONAY | Kaşe / İmza |
|------|-------------|

Yapılan Çalışmanın Konusu: Kullanılan Teknolojiler Hakkında Bilgilendirme

Bugün stajın ilk iş günü şirkete geldiğimde toplantı salonununa geçip beklememi ardından yazılım ekibi lideri ile konuşacağımı söylediler ve bende toplantı odasına girip bekledim. Sonrasında yazılım ekip lideri Emre Bey toplantı odasına geldi ve kısa bir tanışma gerçekleştirdik ardından okulda aldığım dersler, teknik becerimin neler olduğunu ve gerçekleştirdiğim projelerin konusunu ve işleyişini anlatmamı istedi. Bende aldığım dersler ve projelerim hakkında bilgi verdim. Ardından Emre Bey şirketin geliştirmiş olduğu projeler hakkında bilgi verdi. Benden geliştirmemi istedikleri projenin konusunu ve geliştirme aşamasında nasıl bir yol izlemem gerektiğini söyledi. Bana verilen proje bir entegrasyon projesi “Şefim” adında Vega grup yazılım tarafından geliştirilmiş ürünlerin olduğu sipariş ödeme gibi birçok modülü olan sisteme çiçek sepetine gelen siparişlerin yönetimi ile birlikte çiçek sepeti ve şefim veri tabanındaki ürünlerin iki taraflı senkronize edilmesi idi. Projenin kabaca detaylarını yazılım ekip lideri Emre Bey ile konuştuktan sonra şefim ile ilgili hem kod hem de veri tabanı ile bilgili olan şefim üzerinde geliştirme ve bakımları gerçekleştiren Metin Bey ile tanıştım. “Şefim” yazılımı üzerinde herhangi bir teknik konuda Metin Bey’e danışmam gerektiği söylendi. Metin Bey görev olarak aldığım projedeki en büyük sorunun “Şefim” yazılımının legacy bir sistem yani çok uzun zaman önce yazıldığı için üzerine ekleme yapıldıkça kırılgan bir yapıya büründüğünü ve veri tabanının sürekli tablo eklenmesi ile yamalı bir veri tabanı olduğunu söyledi. Günün ileri saatlerinde diğer yazılım ekip üyeleri ile tanışıp sohbet ettik ve günün son saatlerinde staj defterimin ilk sayfasını yazmam için vakit ayırdım. Bu şekilde ilk staj günümü sonlandırmış oldum.

Yapılan Çalışmanın Konusu: Gerekli Kurulumların Yapılması, Arayüz ve Kod İncelenmesi

Bugün stajın ikinci iş günü. “Şefim” yazılımı ürün müdürü Metin Bey entegre projelerinde işin en önemli kısmının entegre yapacağım sistemin yapısını ve işleyişini iyi kavramış olmam gerektiğini söyledi. Ardından sistemi tanımam ve inceleyebilmem için “Şefim” yazılımı ve modüllerini benim bilgisayarıma kurduk. Yazılımı kullanmak için lisansladık ardından projeye ait kaynak kodları incelemem için kendi kullandığım bilgisayara attık. Proje geliştirmesi yapmak için gereken yazılımları önceden kullanmış olduğum fakat bilgisayarımı formatlamış olmamdan dolayı tekrar yüklemem gerekti. İlk olarak .Net geliştirmesi yapmak için Microsoft Visual Studio 2019’u kurdum ardından veritabanı için SQL server 2019’u ve ardından görsel yönetim paneli olan SQL Server Management Studio’yu kurdum. “Şefim” programı .Net Core windows form application olarak geliştirilmiş, UI tarafını güçlendirmek için DevExpress kullanılmış ve verileri kayıt etmek, saklamak için veritabanı olarak MSSQL teknolojisi kullanılmış. Şefim yazılımı ilk restoran otomasyonu olarak piyasaya girmiş fakat üzerine eklenen yeni özellikler ile birlikte farklı sektörlerde kullanılmaya başlamış. İşin kötü yanı sürekli üzerine ekleme yapıldığından sistem fazlasıyla karmaşık bir durum almış sisteme eklenen yeni özelliklerden sonra refactoring işlemi yapılmamış. Yazılım ekip lideri Emre Bey’e neden refactoring yapılmadığını sordum ve çok legacy bir sistemi olduğu için rebuild (yeniden inşa etmek) yapmanın faydalı olacağını söyledim. Emre Bey ise bana geri yanıt olarak “Şefim” uygulamasını kullanan çok fazla müşteri olduğunu ve alışılmışın dışına çıkmak istemediklerini söyledi. Günün son saatlerinde staj defterimin ikinci sayfasını yazmam için vakit ayırdım. Bu şekilde ikinci staj günümü sonlandırmış oldum.

Yapılan Çalışmanın Konusu: Veritabanı Şemasının İncelenmesi ve Anlaşılması

Bugün stajın üçüncü iş günü. “Şefim” ve modüllerinin kullandığı veri tabanını inceleyip kavramaya çalıştım. Önceki günlerde de bahsettiğim gibi eski bir sistem olduğu için sürekli veri tabanına yeni tablolar yeni field lar açılmış durumda. Şefim veri tabanında 59 tablo bulunmakta ve normalizasyon olmadığı için tüm bilgiler tekrar etmekte bu nedenle tablolar arasındaki ilişkiyi anlamak benim için bir hayli zor oldu ve tüm günümü aldı diyebilirim. Şefim veri tabanındaki tüm tabloları analiz ettikten sonra çiçek sepeti ile entegre ederken kullanmam gereken tablolara yoğunlaşmak için tespit etmem gerekiyordu. Sisteme eklenen ürünlerin hangi tablolarda bilgilerinin tutulacağını görmek için program arayüzü üzerinden ürün eklemeleri yaptım ve ürünlerin nasıl tutulduğunu çözdüm. Sonrasında çiçek sepeti üzerinden gelen sipariş kayıtlarının veri tabanına aktırılması ile ilgili veri tabanı tablolarına yoğunlaşmak için bu tabloların hangileri olduğunu Metin Bey’e danışıp öğrendim. Entegre projesinde en çok kullanacağım tablo olan Product tablosu. Bu tablo üzerinde şefim sistemine kayıtlı ürünlerin bilgileri tutulmakta bu tabloya bağıntısı olan Choice1, Choice2 tablolarında ise ürünlere ait seçeneklerin tutuluyor. Örneğin döner ürününü ele alalım. Döner türü iki farklı veya daha fazla olabilir ve fiyat değişkenliği gösterebilir. Bu farklılıkları Choice1 tablosuna “productId” ile birlikte fiyat farkını da belirterek ekliyoruz. Ardından aynı döner ürününü farklı gramaj ile satış yapılması isteniyor fakat choice1 de belirlediğim farklılığı da göz önüne alınması isteniyor ise choice2 tablosuna choice1 id ile birlikte fiyat farkını da girerek kayıt edebiliriz. Kısacası ürünlere ait seçenekler karar ağaçlarına benzer bir yapısı olması çiçek sepeti ile entegrasyon yaparken fazlasıyla uyumsuzluk yaşanacağını sinyallerini vermeye başlıyor. Ürün ile ilişkili tablolara yapmış olduğum analiz sonrası şefimin modülü olan paket servis uygulamasına çiçek sepetinden gelen siparişlerin gösterilmesi için siparişlerin tutulduğu tablo veya tabloları öğrenmek için Metin Bey’e danıştım. Bilgilendirme sonrası Bill, BillHeader ve PhoneOrderHeader tablolarını arasındaki ilişkiyi anlamak için bu 3 tabloyu görüntülemek için arayüz üzerinden sipariş oluşturdum. Günün sonunda veri tabanında kullanacağım tablolar hakkında kısmen de olsa bilgi sahibi oldum.

Yapılan Çalışmanın Konusu: Çiçek Sepeti API Dokümanının İncelenmesi ve Çiçek Sepeti İletişim Kanalına Giriş

Bugün stajın dördüncü iş günü. İki farklı sistem arasında yapılacak entegrasyon projesinde kısmen de olsa sistem taraflarından bir tanesinin çalışma yapısı ve mimarisi hakkında bilgi sahibi oldum. Sıra geldi diğer sistem olan çiçek sepetine. Çiçek sepeti tarafından geliştirilmiş API ile ilgili dokümanları benimle paylaşmasını yazılım ekip lideri Emre Bey'den rica ettim. Emre Bey bana api dokümanlarının yayınlandığı ".dev" uzantılı çiçek sepeti domain linkini gönderdi. Bu site üzerinden api metodlarını incelemeye başladım. Sonrasında Emre Bey takıldığım yerler ile teknik destek alabileceğim Vega yazılım – Çiçek Sepeti iş birliği adlı mail iletişimine ekledi. Bu mail ağında Çiçek Sepeti tarafından yazılım destek ekibinde çalışan bireyler, ben ve yazılım ekip liderim Emre Bey mevcut. Çiçek Sepeti api metodlarını 4 farklı başlık altında toplamış bunlar;

1. Ürün Yönetimi: Bu başlık altında kullanabileceğim metotlar; Ürün yükleme, ürün listeleme, stok ve fiyat güncelleme, ürün bilgilerini güncelleme, ürün işlemleri kontrolü, çiçek sepeti kategori listesi ve çiçek sepeti kategori özellik listesi
2. Sipariş Yönetimi: Bu başlık altında kullanabileceğim metotlar; Sipariş Listeleme, sipariş kargo süreçleri ve fatura gönderim
3. Finansal Bilgiler
4. İade Yönetimi: Bu başlık altında kullanabileceğim metotlar; İade siparişlerin çekilmesi, iade siparişini teslim aldım ve iade sipariş onaylama veya reddetme

Çiçek sepeti API hizmeti restful şeklinde geliştirilmiş ve json verisi üzerinden işlem yapılmaktadır (request ve response json). API hizmetin kullanmak için Authentication'a ihtiyaç var. Authentication sağlamak için request atarken header'a x-api-key parametresi ile key değeri verilmelidir. Postman uygulaması üzerinden birkaç request denemesi yapmak için çiçek sepeti iş birliği mail ağına geliştirme yapmak için key değerine ihtiyacınız olduğunu ve bize key değerini iletmelerini rica ettim. Birkaç saat sonrasında çiçek sepeti tarafından şu şekilde bir api key içeren mail aldık;

*"Tek bir adet dokümanımız bulunmaktadır.
İlgili doküman üzerinden ilerleyebilirsiniz.*

API için bu test API KEY ile bağlanabilirsiniz.

TEST API KEY: `ya1SmuwxDR1kaoXGYQZ9z774XMjA0hb9anjy14IJ`

Sormak istedikleriniz olursa, bu mail üzerinden bizlere sorularınızı sorabilirsiniz.

Teşekkürler"

Mail sonrası test api key'i kullanarak Postman aracılığıyla tüm metotları denedim. Günün son saatlerinde web api denemelerim sonrasında günlük staj defterimi dolduruyorum.

Yapılan Çalışmanın Konusu: Proje Gereksinim Analizi Yapılması

Bugün stajın beşinci iş günü. Günün başlangıcında Eme Bey tüm hafta boyunca yaptığım inceleme ve araştırmaları tekrardan gözden geçirmemi ve genel bir tekrar yapmamın benim için faydalı olacağını söyledi. Bu tavsiye üzerine bende tüm hafta yapmış olduğum incelemeleri baştan bir göz gezdirmek istedim. Şefim programının ilk geliştirildiği zamanlarda bir doküman oluşturulmadığı için projeye sonradan dahil olan programcı ve mühendislerin projenin mimarisini ve akışını kavraması bir hayli zor olmuş. Sonrasında belirli bir kod geliştirme standarttı olmadığı için her yeni gelen yazılımcı kendi standartlarını kullanarak koda eklemeler yapmış. Bu durumda kendisinden sonraki yazılımcıya bir hayli zorluk katmaktadır. Tüm bu zorluklara rağmen kodları tekrardan inceledim ve geçen gün yapmış olduğum incelememin üstüne mimariyi daha çok kavradığımı düşünüyorum. Ardından veritabanı kısmına geçtim. Veritabanı kısmında bir veritabanı diyagramı oluşturulmadığı için kullanacağım tabloları baz alarak küçük bir diyagram oluşturdum. Ardından gerçekleştireceğim entegre projesi hakkında neler yapacağım hakkında Emre Bey ile konuşmuştuk fakat gereksinimlerin resmi halde belirtilmesi için ve kendime yol haritası çizmem açısından Emre Bey'e gereksinimlerin belirlenmesi hakkında toplantı gerçekleştirmek istediğimi söyledim. Emre Bey ile oturup çiçek sepeti api dokümanını açtık bundan öncesinde ikimizde bireysel olarak hangi servislerin olduğunu bireysel olarak incelemiştik. Yazılım ekip lideri emre bey Şefim programının çok eski bilgisayarlarda (işletim sistemi: Windows XP) çalıştığı için geliştirme yaparken çok kaynak tüketiminin fazla olmamasının gerektiğini söyledi. Projenin .net windows form application formatında geliştirmem gerektiğini söyledi. GUI tarafında üst düzey bir seviye yerine idarelik bir tasarım ilk aşamada prototip seviyesinde yeterli olacağını sonraki aşamada iyileştirme yapılabileceğini söyledi. Emre Bey ile proje başlangıcında ilk olarak çiçek sepeti üzerinden gelen siparişin Şefim programının modülü olan şefim paket servis uygulamasında görüntülenmesini ilk yapılacak olarak belirledik. Bu işlemi gerçekleştirdikten sonra sıradaki işlemin ne olacağını kararlaştıracacağımız hakkında anlaştık. Bu şekilde stajımın ilk haftasını tamamlamış oldum. Özetle stajımın ilk haftası tanışma, kullanacağım teknolojiler hakkında bilgi sahibi olma ve geliştireceğim entegre projesi hakkında nasıl bir yol izleyeceğimi aklımda belirlemiş oldum.

Yapılan Çalışmanın Konusu: Proje Oluşturulması ve Başlangıcı

Bugün stajın ikinci haftanın ilk günü. Geçtiğimiz hafta boyunca gerekli araştırma ve hazırlıkları tamamladıktan sonra projenin oluşturulması ve başlanmasına sıra geldi. Geçen hafta kurulumunu yapmış olduğum Microsoft Visual Studio üzerinden yeni bir windows form application projesi oluşturdum. Ardından versiyon kontrolü yapabilmek için kendi github hesabım üzerinden private bir repo açtım ve programın ilk versiyonunu publish (yayınladım) ettim. Yapacağım proje entegre projesi olduğu için şefim veri tabanına erişim sağlayıp dışardan gelen verileri yazmam gerektiği için SQL adında bir sınıf oluşturdum. Ve bu sınıf içerisinde sürekli kullanacağım açıp kapatma metotlarını yazdım.

```
public class SQL
{
    readonly SqlConnection conn = new SqlConnection("Data Source=SEFA; Initial Catalog=sefim; User Id=sa; password=sefim1234;");

    11 başvuru
    public void OpenDatabase()
    {
        conn.Open();
    }

    6 başvuru
    public void CloseDatabase()
    {
        conn.Close();
    }

    17 başvuru
    public SqlConnection GetConn()
    {
        return conn;
    }
}
```

Sonrasında veritabanı tablolarında gelen siparişleri kayıt edeceğim Bill, BillHeader ve PhoneOrderHeader tabloların karşılığı olan model sınıflarımı oluşturdum. Bu sınıflara ait “save” metotlarını yazdım. Bu metotların görevi sınıftan üretilen nesnenin property lerini kullanarak veritabanına kayıt işlemini gerçekleştiriyor. Sonrasında çiçeksepeti dokümanından ilk kullanacağım “getOrder” (sipariş listeleme) metodunun kullanmak için bize verilen api-key i request header’a ekleyerek requesti postman üzerinden yolladım. Fakat aynı requesti geçen hafta denediğimde gelen response üzerinden gelen veriler bu sefer gelmedi. Bu sorunun kaynaklanma nedenini çiçek sepeti aynı api-key i farklı bir test kullanıcısıyla paylaştığını tespit ettik. Bu durumu çözmek için e-posta aracılığıyla çiçek sepeti destek ekibinden test ortamında kullanabileceğim başka bir test mağazasına ait api-key ricasında bulundum. Yaptığım değişiklikleri kaydetmek için github reposuna commit attım.

```
public class Bill
{
    SQL sql = new SQL();

    public DateTime Date;
    public string UserName;
    public string ProductName;
    public int ProductId;
    public int Choice1Id;
    public int Choice2Id;
    public string Options;
    public decimal Price;
    public decimal Quantity;
    public int HeaderId;
    public int OrderId;
    public int CampaignSold;
    public int CampaignFree;
    public int Canceling;
    public decimal OriginalPrice;
    public int Printed;
    public int OrderState;
    public int CampaignDetailId;
    public int isReady;
    public int isSee;
    public DateTime ReadyDate;
    public string UserK;
    public int Aktarildi;
}

public class PhoneOrderHeader
{
    SQL sql = new SQL();

    public string Address;
    public string CustomerName;
    public string OrderNo;
    public string OrderOwnerName;
    public string PaymentNote;
    public string PhoneNumber;
    public string Deliverer;
    public byte Paid;
    public int HeaderId;
    public DateTime CreationTime;
    public string CreatedByUserName;
    public string Directions;
    public string CustomerNote;
    public decimal Discount;
    public string TrackingNumber;
    public DateTime AssignDate;
    public string PaymentDetail;
    public int DailyNumber;
    public byte Aktarildi;
}

public class BillHeader
{
    SQL sql = new SQL();

    public int BillState;
    public int BillType;
    public int SubBillNo;
    public int PaymentId;
    public string TableNumber;
    public int Persons;
    public string Fisno;
    public string FormalBillId;
    public string DailyBillNumber;
    public byte Aktarildi;
}
```

Yapılan Çalışmanın Konusu: Siparişlerin Aktarılmasının Gerçekleştirimi

Bugün stajın ikinci haftanın ikinci günü. Dün atmış olduğum e-postanın yanıtı sabah şirkete geldiğinde maillerimi kontrol ederken gördüm ve yeni api-key ile postman kullanarak requesti attım. Gelen response üzerinde sipariş edilmiş 6 ürün vardı. Bu sayede siparişlerin aktarılmasını implement edebilir hale gelmiştik. Postman'dan attığım requeste karşılık gelen response'u düzenli bir şekilde incelemek için "<http://json.parser.online.fr/>" sitesini kullandım. Ve response verilerini oluşturmuş olduğum modellere aktararak veritabanına kaydetme işlemlerini yaptım. Verileri veritabanına kaydederken siparişlere ait ürünlerin Bill tablosunda productId field'ı var. Bu alanı doldurmak için şefim tabanı Product tablosu üzerinde kayıtlı olan ilgili ürünün id 'sine erişmek için Product adında bir sınıf oluşturdum ardından bir dictionary (sözlük (veri tipi)) tanımladım ve birkaç metod yazdım bu metotlar; Add, SearchId ve ReadDatabase

```
4 başvuru
public void ReadDatabase()
{
    SQL sql = new SQL();
    using (sql.GetConn())
    {
        SqlCommand command = new SqlCommand(
            "select ProductName,Id from Product",
            sql.GetConn());
        sql.OpenDatabase();

        SqlDataReader reader = command.ExecuteReader();

        if (reader.HasRows)
        {
            while (reader.Read())
            {
                Console.WriteLine(reader["Id"].ToString());

                Add(reader["ProductName"].ToString(), Int16.Parse(reader["Id"].ToString()));
            }
        }
        else
        {
            Console.WriteLine("No rows found.");
        }
        reader.Close();
    }
}
```

ReadDatabase metodu sayesinde veri tabanındaki Product tablosundaki tüm kayıtları okuyup tanımladığım dictionary ye eklemek için yazdığım fonksiyon olan "Add" fonksiyonunu kullanıyor.

```
1 başvuru
public void Add(string name, int code)
{
    try
    {
        dictionary.Add(name, code);
    }
    catch
    {
    }
}
```

Add fonksiyonu kendisine parametre olarak gönderilen ürün adını ve ürün id sini alıp tanımladığım dictionary ye ekliyor. Ürün adı key değeri olarak, ürün id si ise value olarak kaydediliyor.

```
5 başvuru
public int SearchId(string name)
{
    try
    {
        return dictionary[name];
    }
    catch
    {
        return -1;
    }
}
```

SearchId fonksiyonu ise kendisine parametre olarak gönderilen ürün adının şefim tablosunda olup olmadığını kontrol ediyor eğer ürün var ise ürün id' sini geri döndürüyor eğer ürün yok ise -1 değerini geri döndürüyor.

Yapılan Çalışmanın Konusu: Siparişlerin Aktarılmasının Testi ve Geliştirilmesi

Bugün stajın ikinci haftanın üçüncü günü. Günün ilk saatlerinde son iki gün boyunca yazmış olduğum kodları tekrar inceledim ve hoşuma gitmeyen kod kısımlarını tekrardan refactoring yaptım. Çiçek sepeti tarafından gelen siparişlerde bulunan ürünlerin senkronizasyonu sağlamak için maalesef ki string matching yapılması zorunda çünkü her iki taraf arasında matching yapılacak bir unique id veya code bulunmamakta. Siparişler şefim programına aktarılırken siparişe ait ürün şefimde bulunamadıysa Bill tablosunu kayıt ederken ilgili ürünün id sini bilmediğimiz için işlem başarısız oluyordu. Bunun önüne geçmek için çiçek sepeti tarafından gelen ürün şefim programı tarafında yani veri tabanında kayıtlı değil ise kayıt etme işlemini yapmak olacağını düşündüm. Bu düşüncemi implement etmek için önceden oluşturduğum Product sınıfı kullanacağım. Product sınıfına veri tabanında olan field leri attrubites olarak sınıfıma ekledim. Sonrasında veritabanına kayıt işlemini gerçekleştirecek olan "Save" metodumu yazdım.

```
2 başvuru
public void Save()
{
    SQL sql = new SQL();

    sql.OpenDatabase();

    string CmdText = "INSERT INTO Product(" +
        "[ProductName],[ProductGroup],[ProductCode]" +
        ",[Price],[VatRate],[FreeItem]" +
        ",[ProductType],[SkipOptionSelection],[Aktarildi]" +
        "VALUES (@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8,@p9)";

    SqlCommand command = new SqlCommand(CmdText, sql.GetConn());

    command.Parameters.AddWithValue("@p1", ProductName);
    command.Parameters.AddWithValue("@p2", ProductGroup);
    command.Parameters.AddWithValue("@p3", ProductCode);
    command.Parameters.AddWithValue("@p4", Price);
    command.Parameters.AddWithValue("@p5", VatRate);
    command.Parameters.AddWithValue("@p6", FreeItem);
    command.Parameters.AddWithValue("@p7", ProductType);
    command.Parameters.AddWithValue("@p8", SkipOptionSelection);
    command.Parameters.AddWithValue("@p9", Aktarildi);
    command.ExecuteNonQuery();

    sql.CloseDatabase();
}
```

Çiçek sepeti üzerinden gelen siparişlerin şefim tarafında kayıtlı olup olmadığını kontrol etmek için basit bir if şartı yazdım ve eğer ürün şefim veritabanına yok ise Save metodunu çağırıp insert işlemini yapacak. Bu if şartı önceden yazmış olduğum SearchId metodunu kullanıyor.

```
if (product.SearchId(details["supplierOrderListWithBranch"][i]["name"].ToString()) == -1)
{
    product.ProductName = details["supplierOrderListWithBranch"][i]["name"].ToString();
    product.ProductGroup = "Cicek Sepeti";
    product.ProductCode = details["supplierOrderListWithBranch"][i]["productCode"].ToString();
    product.Price = Convert.ToDecimal(details["supplierOrderListWithBranch"][i]["totalPrice"]);
    product.VatRate = Convert.ToDecimal(details["supplierOrderListWithBranch"][i]["tax"]);
    product.FreeItem = 0;
    product.ProductType = "Normal";
    product.SkipOptionSelection = 0;
    product.Aktarildi = 0;
    product.Save();
    product.ReadDatabase();
}
```

Yapılan Çalışmanın Konusu: Entegre Projesine Giriş Yetki Sorgusu Yapılması

Bugün stajın dokuzuncu günü. Emre Bey son staj gününden sonra araya bayram tatili girmesi nedeniyle yazmış olduğum kodları tekrar hatırlamak amacıyla tekrar göz gezdirmemenin faydalı olacağını söyledi. Göz gezdirme zamanında kendime ve bir sonraki yazılımcıya yardımcı olması için yazdığım commentleri okudum. Bu hatırlama seansında sadece ben değil tüm yazılım ekip çalışanları en son projelerde nasıl kaldıklarını hatırlamak için kodlarını tekrar incelediler. Ardından entegre projesinde ilk gerçekleştirmiş olduğum task'ın demosunu göstermek için yazdığım kodları tetiklemek için bir button oluşturdum ve butona tıklandığında yazmış olduğum kodları çalıştırıyordu. Siparişlerin aktarılma işlemi arka planda otomatik olarak gerçekleştirileceği için windows servis haline getireceğim fakat şimdilik test için buton ile tetiklemek yeterli olacaktır. Sonrasında yazılım ekip lideri Emre Bey'e demo sunumumu yaptım. Sunum sonrası yazılım ekip lideri Emre Bey Entegre projesinin ara yüzüne giriş yapmak için şefim veri tabanında kayıtlı olan "Admin" adlı kullanıcının şifresi ile kullanıcının girdiği şifreyi karşılaştırmak için "Verification" adında bir sınıf oluşturdum.

```
public class Verification
{
    2 başvuru
    public string TruePassword
    {
        get { return getPassword(); }
        set { TruePassword = value; }
    }

    1 başvuru
    private string getPassword()
    {
        string password="";
        SQL sql = new SQL();
        using (sql.GetConn())
        {
            SqlCommand command = new SqlCommand(
                "select Password from [User] where UserName='Admin'",
                sql.GetConn());
            sql.OpenDatabase();

            SqlDataReader reader = command.ExecuteReader();

            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    password = reader["Password"].ToString();
                }
            }
            else
            {
                Console.WriteLine("No rows found.");
            }
            reader.Close();
        }
        return password;
    }
}
```

Bu sınıf içerisinde "TruePassword" adında bir property var bu property get metodunda kendi yazmış olduğum "getPassword" metodunu çağırarak bu metodun işlevi şefimin veri tabanındaki User adlı veri tabanında Admin kullanıcısının şifresini çekmeye yarıyor.

Yeni sınıfı ekledikten sonra GitHub Desktop uygulaması aracılığıyla proje reposuna yeni commit i push ettim. Ve böylece bugünkü staj günüm sona ermiş oldu.

Yapılan Çalışmanın Konusu: Şefim Ürünlerinin Çiçek Sepetine Aktarımı I

Bugün stajın onuncu günü. Siparişlerin aktarılması işleminin ufak detaylarının kalmasından sonra Emre Bey yeni gerçekleştirmem gereken task'ın Şefim yazılımı tarafından şefimin kendi veritabanına kaydedilen ürünlerin çiçek sepeti tarafına aktarılmasını olduğunu söyledi. Sonrasında bende çiçeksepeti web servis dokümanı üzerinden kullanmam gereken servisi belirleyip Postman ile deneme yaptım. Yeni bir form ekranı oluşturdum ve şefim tarafındaki ürünlerin listeleneceği bir listview oluşturdum. Sonrasında Product sınıfından bir object oluşturup önceden yazdığım metod olan "ReadDatabase" kullanarak Product nesnesinin değişkeni olan dictionary adı gibi dictionary yani sözlük tipindeki değişkeni doldurdum. Bu değişkeni kullanarak listview ı doldurdum. Şefim tarafından çiçek sepetine ürünü aktarırken sadece ürünün isim, kod ve fiyat bilgisini kullanabiliyorum. Diğer request parametrelerini ürün aktarma ekranı yapıp kullanıcıdan almam gerekiyor bunları tespit etmek için örnek requesti ve parametre bilgilerini inceledim.

```
{
  "products": [
    {
      "productName": "Test Urunu",
      "mainProductCode": "testmain2",
      "stockCode": "test2",
      "categoryId": 13349,
      "description": "test açıklaması",
      "mediaLink": "test",
      "deliveryMessageType": 4,
      "deliveryType": 3,
      "stockQuantity": 25,
      "salesPrice": 56.99,
      "listPrice": 99.99,
      "barcode": "test123",
      "images": [
        "test-test-kc6500945-1-392bf4631cf04904a3f0df31a6c10642.jpg"
      ],
      "attributes": [
        {
          "id": 2000353,
          "valueId": 2010697,
          "textLength": 0
        },
        {
          "id": 13,
          "valueId": 143,
          "textLength": 0
        },
        {
          "id": 3,
          "valueId": 191,
          "textLength": 25
        }
      ]
    }
  ]
}
```

| Parametre | Açıklama | Tip | Zorunlu mu? |
|---------------------|-----------------------------------------------------------------------------|---------|-------------|
| productName | Ürün adı | string | Evet |
| mainProductCode | Tedarikçi ürün kodu | string | Evet |
| stockCode | Tedarikçi varyant kodu | string | Evet |
| categoryId | Ürün/varyant kategori kodu, kategori listesinden çekilecek | integer | Evet |
| description | Ürün/varyant açıklaması | string | Evet |
| mediaLink | Ürün/varyant ile ilgili video linki(youtube), imaj linki gönderilmemelidir. | string | Hayır |
| deliveryMessageType | Teslimat aralığı | integer | Evet |
| deliveryType | Teslimat tipi | integer | Evet |
| stockQuantity | Ürün stok adedi | integer | Evet |
| salesPrice | Satış fiyatı | double | Evet |
| listPrice | Üstü çizili fiyat | double | Hayır |
| barcode | Reklamlar için evrensel ürün kodu | string | Hayır |
| attributesid | Kategori özellik listesinde çekilecek attribute id | integer | Hayır |
| attributesvalueid | Kategori özellik listesinde çekilecek attribute value id | integer | Hayır |
| attributetextLength | Kişiyi özel ürünlerde eklenecek bilginin karakter sayısı | integer | Hayır |
| images | Ürün/varyant görseli | string | Evet |

Ürünün adı ve fiyatı haricinde kullanıcıdan almam gereken kısımlar categoryId (Bu id için ayrı bir servis kullanacağım), açıklama, teslimat tipi, teslimat aralığı, resim ve ürüne ait özelliklerin id'si (Bu özellikleri de ayrı bir servis kullanacağım).

Yapılan Çalışmanın Konusu: Çiçek Sepetine Aktarım ve Kategori Servisi

Bugün stajın on birinci günü. Şefim ürünlerin aktarılmasını ile ilgili bir form oluşturup ürünlerin listelenme işlemini yapmıştım. Sonrasında seçilen ürünün isim bilgisini parametre olarak göndereceği yeni bir kayıt formu oluşturulacağı kararını vermiştim. Bu form da olması gereken birkaç input field'ı belirlemiştim. Ürün aktarma tarafında ürünün kategori bilgisini ayrı bir servis kullanarak id aracılığıyla gönderilmesi zorunlu. Bunun için Çiçeksepeti kategori listesi servisini inceledim. Servis için sadece api-key bilgisi yeterli olacak. Response olarak ise ağaç kökünden dallarına gider şekilde bir dallanma söz konusu.

```
"id":255,
"name":"Hediye",
"parentCategoryId":null,
"subCategories":[]
  {
    "id":12453,
    "name":"Evcil Hayvan Ürünleri",
    "parentCategoryId":255,
    "subCategories":[]
      {
        "id":12467,
        "name":"Kuş",
        "parentCategoryId":12453,
        "subCategories":[]
          {
            "id":12612,
            "name":"Kuş Aksesuarları",
            "parentCategoryId":12467,
            "subCategories":[]
          }
        }
      }
    }
```

Sol tarafta kategori response undan bir kısmı alıntı olarak koydum. Görüldüğü gibi her node (düğüm) de id, name, parentCategoryId ve subCategories bilgisi bulunmakta. Resimde görüldüğü gibi derinlik seviyesi 4 olarak gözükmemekte. Bu kategori isimlerini hafızada tutmak için dictionary(sözlük) tanımladım key olarak kategori isimi ve ürün yükleme requesti üzerinde kullanmak için id bilgisini de value olarak hafızada tutuyorum. Son olarak bu oluşturduğum dictionary i parametre olarak ürün aktarma ekranına yolladım.

```
public void getCategory()
{
    var client = new RestClient("https://sandbox-apis.ciceksepeti.com/api/v1/Categories");
    client.Timeout = -1;
    var request = new RestRequest(Method.GET);
    request.AddHeader("x-api-key", "ya1SmuwxDR1kaoXGYQZ9z774XMjA0hb9anjy141J");
    IRestResponse response = client.Execute(request);
    var details = JObject.Parse(response.Content);

    foreach (var firstCategories in details["categories"])
    {
        foreach (var secondCategories in firstCategories["subCategories"])
        {
            dictionary.Add(secondCategories["name"].ToString(), Int16.Parse(secondCategories["id"].ToString()));
            foreach (var thirdCategories in secondCategories["subCategories"])
            {
                try
                {
                    dictionary.Add(thirdCategories["name"].ToString(), Int16.Parse(thirdCategories["id"].ToString()));
                }
                catch{
                }
            }
            foreach (var fourthCategories in thirdCategories["subCategories"])
            {
                try
                {
                    dictionary.Add(fourthCategories["name"].ToString(), Int16.Parse(fourthCategories["id"].ToString()));
                }
                catch{
                }
            }
        }
    }
}
```

Yapılan Çalışmanın Konusu: Ürüne ait Özelliklerin Kullanıcıdan Alınması

Bugün stajın on ikinci iş günü. Ürünün aktarılma işleminde çiçek sepeti tarafından gönderilmesi zorunlu tutulan bilgiler var fakat şefim yazılımı veri tabanında bu bilgiler mevcut değil. Bu yüzden şefim kullanıcıları ürünlerini çiçek sepetine kopyalarken form ekranı üzerinden ekstra bilgiler girmeli bunlardan birincisi açıklama, bu bilgiyi girebileceği bir text box koydum. İkincisi ürüne ait resim, şefim veri tabanında ürüne ait resimlerin local bilgisayarda tutulduğu adresi tutan Product tablosu ile ilişkili bir tablo mevcut fakat çiçek sepeti resmin cloud üzerindeki adresini istediği için kullanıcıya adres girebileceği bir text box ekledim.

```
dicCategory = dictionary;  
textBoxName.Text = name;  
foreach (var item in dictionary.Keys)  
{  
    cmbCategory.Items.Add(item);  
}
```

Üçüncü alan ise parametre olarak forma gönderdiğim dictionary nin key değerlerinde olan kategori isimlerini kullanıcının seçim yapabilmesi için combo box içerisine ekleme işlemini form un load eventi içerisinde gerçekleştirdim.

```
1 Başvuru  
private void btnPreview_Click(object sender, EventArgs e)  
{  
    pictureBox1.ImageLocation = textBoxImageUrl.Text;  
}
```

Yazılım ekip lideri Emre Bey ricası üzerine linki kopyalan ürün resminin ön izlenebilmesini istedi. Bu task 1 gerçekleştirilebilmek için bir buton ve

pictureBox tanımladım. Butonun click eventi kısmında işlemi gerçekleştirdim.

deliveryMessageType Değerleri: 1:Çiçek_Servis 4: Hediye Kargo AynıGün, 5:Hediye Kargo 1-3 Is Gunu, 6:Hediye Kargo 1-5 Is Gunu, 7:Hediye Kargo 1-7 Is Gunu, 13: Hediye Kargo 3-5 Is Gunu, 18: Hediye Kargo 1-2 Is Gunu, 19:Hediye Kargo 1-10 Is Gunu

```
dicDeliveryMessageType.Add("Hediye Kargo Aynı Gün", 4);  
dicDeliveryMessageType.Add("Hediye Kargo 1-3 Is Gunu", 5);  
dicDeliveryMessageType.Add("Hediye Kargo 1-5 Is Gunu", 6);  
dicDeliveryMessageType.Add("Hediye Kargo 1-7 Is Gunu", 7);  
dicDeliveryMessageType.Add("Hediye Kargo 3-5 Is Gunu", 13);  
dicDeliveryMessageType.Add("Hediye Kargo 1-2 Is Gunu", 18);  
dicDeliveryMessageType.Add("Hediye Kargo 1-10 Is Gunu", 19);  
foreach (var item in dicDeliveryMessageType.Keys)  
{  
    cmbMessageType.Items.Add(item);  
}
```

Dördüncü alan ise Teslimat aralığı çiçek sepeti dokümanında değerlerini belirtildiği bir kısım var. O değerleri baz alarak form ekranına eklediğim combo box ı doldurdum.

deliveryType Değerleri: 1: Servis Aracı İle Gönderim, 2:Kargo İle Gönderim, 3: Kargo+Servis Aracı İle Gönderim

```
dicDeliveryType.Add("Servis Aracı İle Gönderim", 1);  
dicDeliveryType.Add("Kargo İle Gönderim", 2);  
dicDeliveryType.Add("Kargo+Servis Aracı İle Gönderim", 3);  
foreach (var item in dicDeliveryType.Keys)  
{  
    cmbType.Items.Add(item);  
}
```

Beşinci alan ise Teslimat tipi çiçek sepeti dokümanında değerlerini belirtildiği bir kısım var. O Değerleri baz alarak form ekranına eklediğim combo box ı doldurdum.

Yapılan Çalışmanın Konusu: Kategoriye ait Özelliklerin Elde Edilmesi

Bugün stajın on üçüncü iş günü. Ürünlerin çiçek sepetine aktarılması ile ilgili şu ana kadar yazdığım kodları inceleyip refactoring yaptım. En son ürün bilgilerinin kullanıcıdan alınması için gerekli input field ları eklemiştim. Bugün ise ürün ekleme kısmında seçilen kategoriye ait gönderilmesi gereken özelliklerin çekilmesi ve kolay, efektif şekilde kullanabilir bir veri yapısı ile bilgisayarın belleğinde tutmayı amaçlıyorum. İşe çiçek sepeti dokümanında bulunan “Çiçeksepeti Kategori Özellik Listesi” adındaki metot dokümanını açtım. Servis Get request olarak çalışıyor parametre olarak kategorinin id'si (kategori bilgilerini tutmak için oluşturduğum dictionary veri tipinde key, value olarak tutuyorum zaten) gönderilmesi yeterli. Postman üzerinden özellikleri çekebileceğim requesti kullandım ve gelen response yapısı resimde görüldüğü gibi.

```
{
  "categoryId":13173,
  "categoryName":"Dizüstü Bilgisayar
(Laptop)",
  "categoryAttributes": [
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    {
      "attributeId":2000541,
      "attributeName":"Kapasite",
      "required":false,
      "varianter":false,
      "type":"Ürün Özelliği",
      "attributeValues": [
        {
          "id":2046100,
          "name":"1 TB"
        },
        {
          "id":2046101,
          "name":"128 GB "
        }
      ]
    }
  ]
}
```

Web servise parametre olarak göndermiş olduğum kategori id'si, kategori adı ve bu kategoriye ait özelliklerin dizisi. Özellikler dizesi altında her bir node da özelliğin id'si, özelliğin adı, gerekliliği, tipi ve özelliğe ait gönderilmesi gereken değerlerin id leri ve adı bulunmakta.

Kullanıcı her combo box üzerinden her kategori seçme işleminde gelen kategori özellikleri yenilenmeli. Bu yüzden kategori isimlerinin listelendiği combo box ın “Selected Index Changed” Eventi her gerçekleştirdiğinde “Çiçeksepeti Kategori Özellik Listesi” metoduna tekrardan ilgili kategori id si ile birlikte request atılmalı. Bununla ilgili bir method yazdım ve her event gerçekleştiğinde yazmış olduğum methodu çağırarakta.

```
var client = new RestClient("https://sandbox-apis.ciceksepeti.com/api/v1/Categories/" + dicCategory[cmbCategory.SelectedItem.ToString()] + "/attributes");
client.Timeout = -1;
var request = new RestRequest(Method.GET);
request.AddHeader("x-api-key", "bV0k6tFB3x9tWYUx2HmRq56sOSxRHLm08vAHnNEY");
IRestResponse response = client.Execute(request);
attributesResponse = JObject.Parse(response.Content);
```

Yarın gelen response a göre dinamik bir kullanıcı arayüzü yapacağım. Bugünlük bu kadar yaptığım değişiklikleri kaydetmek için github reposuna commit attım.

Yapılan Çalışmanın Konusu: Kategoriye ait Özelliklerin Kullanıcıdan Alınması

Bugün stajın on dördüncü iş günü. Dün kategoriye ait özelliklerin elde edilmesini çiçek sepetinin ilgili web servisini kullanarak gerçekleştirmiştim. Bugün bu verileri kullanarak çiçek sepetine eklemek istediği ürünün kategori özelliklerini kullanıcıdan almak için dinamik bir form oluşturdum. Özellik bilgileri açık uçlu olmadığı için combo box üzerinden özelliğe ait value değerlerinden birini seçmelidir. Bu yüzden her özelliğe ait bir combo box ve hangi özellik olduğunu belirten label oluşturdum. Bu oluşturma işlemlerini kategorilerin listelendiği combo box ın “Selected Index Changed” eventi ile tetiklenerek çağrılacak olan kendi yazdığım metod içerisinde yaptım.

```
startPoint = 6;
comboBoxCount = 0;
foreach (var combo in ComboBoxArray)
{
    this.Controls.Remove(combo);
}
foreach (var label in LabelArray)
{
    this.Controls.Remove(label);
}
for (int i = 0; i < 35; i++)
{
    ComboBoxArray[i] = null;
    LabelArray[i] = null;
}
```

Bu kod bloğu metodun başlangıcı olan kısım. Kategori değiştiğinde geçmişte ki kategoriye ait özelliklerin combo box ve label'ların silinme işleminin yapıldığı kısımdır. startPoint değişkeni ise komponentlerin form üzerindeki y koordinatını temsil etmekte her bir komponent yerleştirildikten sonra komponentlerin alt alta gelmesi için x koordinatı sabit tutup y koordinatı artırdım.

```
foreach (var attributes in attributesResponse["categoryAttributes"])
{
    if (Convert.ToString(attributes["type"]) != "Dinamik Özellik")
    {
        ComboBox myCombo = new ComboBox();
        ComboBoxArray[comboBoxCount] = myCombo;

        myCombo.Name = attributes["attributeId"].ToString();
        myCombo.Location = new System.Drawing.Point(575, startPoint);
        myCombo.Width = 171;
        myCombo.Height = 21;
        myCombo.Font = new Font("Times New Roman", 10);

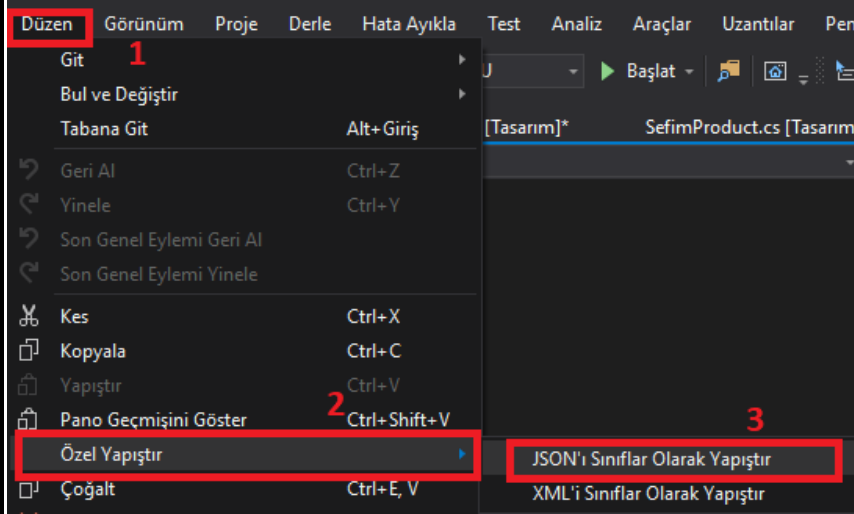
        Label mylab = new Label();
        LabelArray[comboBoxCount] = mylab;
        comboBoxCount++;
        mylab.Text = attributes["attributeName"].ToString();
        mylab.Font = new Font("Times New Roman", 13, FontStyle.Bold);
        mylab.Location = new Point(320, startPoint);
        mylab.AutoSize = true;
        mylab.Padding = new Padding(1);

        foreach (var value in attributes["attributeValues"])
        {
            myCombo.Items.Add(value["name"]);
        }
        this.Controls.Add(myCombo);
        this.Controls.Add(mylab);
        startPoint++;
    }
}
```

Dün kategoriye ait özelliklerin web requesti atacak kod bloğunu oluşturmuştum. Bugün response form ekranına döktüm. Özelliklerin id bilgileri ekstra bir değişkende tutmak yerine oluşturmuş olduğum combo box ların name bilgisi yerinde tuttum. Her oluşturan combo box lara ait özelliklerin value değerlerini ekledim.

Yapılan Çalışmanın Konusu: Ürün Yükleme Modellerinin Oluşturulması

Bugün stajın on beşinci iş günü. Dün ve dünenden önceki gün kullanıcıdan request parametresi olarak gönderilmesi gereken bilgileri alabileceğimiz form ekranını oluşturma işlemlerini tamamlamıştım. Bugün itibari ile input field lardan gelen bilgilerin bir modelle oturtturulması için model oluşturdum. Oluştururken internetten araştırmalarım sonucunda çok pratik bir yöntem buldum. Bu yöntem sayesinde elinizdeki json formatındaki veriye göre ilişkili olarak model sınıflarını otomatik oluşturuyor.



Bu gösterdiğim özelliği kullanarak çiçek sepeti dokümanındaki ürün yükleme sayfasındaki örnek requesti kullanarak modellerimi oluşturdum.

```
2 başvuru
public class Rootobject
{
    1 başvuru
    public SendProduct[] products { get; set; }
}
```

```
3 başvuru
public class Attribute
{
    public int id;
    public int valueId;
}
```

```
5 başvuru
public class SendProduct
{
    1 başvuru
    public string productName { get; set; }
    1 başvuru
    public string mainProductCode { get; set; }
    1 başvuru
    public string stockCode { get; set; }
    1 başvuru
    public int categoryId { get; set; }
    1 başvuru
    public string description { get; set; }
    2 başvuru
    public int deliveryMessageType { get; set; }
    2 başvuru
    public int deliveryType { get; set; }
    1 başvuru
    public int stockQuantity { get; set; }
    1 başvuru
    public decimal salesPrice { get; set; }
    1 başvuru
    public string[] images { get; set; }
    1 başvuru
    public Attribute[] attributes { get; set; }
}
```

```
{
  "products": [
    {
      "productName": "Test Urunu",
      "mainProductCode": "testmain2",
      "stockCode": "test2",
      "categoryId": 13349,
      "description": "test açıklaması",
      "mediaLink": "test",
      "deliveryMessageType": 4,
      "deliveryType": 3,
      "stockQuantity": 25,
      "salesPrice": 56.99,
      "listPrice": 99.99,
      "barcode": "test123",
      "images": [
        "test-test-kc6500945-1-392bf4631cf04904a3f0df31a6c10642.jpg"
      ],
      "attributes": [
        {
          "id": 2000353,
          "valueId": 2010697,
          "textLength": 0
        },
        {
          "id": 13,
          "valueId": 143,
          "textLength": 0
        },
        {
          "id": 3,
          "valueId": 191,
          "textLength": 25
        }
      ]
    }
  ]
}
```


Yapılan Çalışmanın Konusu: Ürün Yükleme Metodunun Kullanılması

Bugün stajın on altıncı iş günü. Dün kullanıcıdan alınan verilerin tek bir yapıda tutulması için model oluşturmuştum. Bugün input field lardaki bilgilerin modele geçirilmesi ve requesti gönderme işlemini yaptım.

```
Attribute[] attribute = new Attribute[comboBoxCount];
int valueID = 0;

for (int i = 0; i < comboBoxCount; i++)
{
    foreach (var item in attributesResponse["categoryAttributes"])
    {
        if (item["attributeId"].ToString() == ComboBoxArray[i].Name)
        {
            foreach (var item2 in item["attributeValues"])
            {
                if (item2["name"] == ComboBoxArray[i].SelectedItem)
                {
                    valueID = Convert.ToInt32(item2["id"]);
                }
            }
        }
    }
    Attribute attr = new Attribute();
    attr.id = Convert.ToInt32(ComboBoxArray[i].Name);
    attr.valueId = valueID;
    attribute[i] = attr;
}
```

```
string[] images = new string[1];
images[0] = textBoxImageUrl.Text;
SendProduct[] sendProduct = new SendProduct[1];
SendProduct sndProduct = new SendProduct();
sndProduct.attributes = attribute;
sndProduct.images = images;
sndProduct.productName = textBoxName.Text;
sndProduct.mainProductCode = getProductCode();
sndProduct.stockCode = getProductCode();
sndProduct.categoryId = dicCategory[cmbCategory.SelectedItem.ToString()];
sndProduct.description = textBoxDescription.Text;
sndProduct.deliveryMessageType = dicDeliveryMessageType[cmbMessageType.SelectedItem.ToString()];
sndProduct.deliveryType = dicDeliveryType[cmbType.SelectedItem.ToString()];
sndProduct.stockQuantity = 999;
sndProduct.salesPrice = Convert.ToDecimal(textBoxPrice.Text);
if (sndProduct.deliveryType != 2)
{
    sndProduct.deliveryMessageType = 1;
}
sendProduct[0] = sndProduct;
Rootobject rootobject = new Rootobject();
rootobject.products = sendProduct;
var json = new JavaScriptSerializer().Serialize(rootobject);

var client = new RestClient("https://sandbox-apis.ciceksepeti.com/api/v1/Products");
client.Timeout = -1;
var request = new RestRequest(Method.POST);
request.AddHeader("x-api-key", "bV0k6tFB3x9twyUx2HmRq56sOSxRHLm08vAHnNEY");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("application/json", json, ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
var details = JObject.Parse(response.Content);
```

İlk olarak en küçük düğüm olan attrubite den başladım. Özellik sayısı boyutunda attrubite tipinde array oluşturdum. Sonrasında kullanıcın seçmiş olduğu özelliğe ait value name ile ram de tuttuğum kategori özellik response ile karşılaştırarak id bilgisini elde ettim. Elde ettiğim id bilgisi ile combo box in adı olarak sakladığım özellik id si ve value id lerini dizime ekledim.

Formdan aldığım diğer bilgileri modele ekledim. Sonrasında JavaScriptSerializer sınıfını kullanarak modelimi serialize ettim. Serialize ettiğim json verisini requestimin body kısmına ekledik ten sonra requesti gönderdim.

Yapılan Çalışmanın Konusu: Ürün Yükleme Sonrası Batch-Status Metodu Kullanımı

Bugün stajın on yedinci iş günü. Dün itibari ile ürün yükleme web servisini kullanma işlemini gerçekleştirmiştim. Response olarak batch id dönüyor. Fakat çiçek sepeti sayfasında batch-status sorgu servisinin olduğu sayfaya erişim kapalıydı. Ürün yükleme işleminin gerçekleşmesi ile ilgili bilginiz olmuyordu. Bu sorun ile ilgili çiçek sepeti yetkililerinden yardım almak için çiçeksepeti – Vega grup yazılım iş birliği e-posta ağına aşağıdaki maili attım.

“Merhaba,

Sipariş listeleme ve ürün yükleme işlemlerini gerçekleştirdik. Response olarak "batchId" alıyoruz fakat CRUD işlemlerin doğru çalıştığı teyit etmek için henüz herhangi bir arayüz olmadığını söylemiştiniz bununla birlikte anladığım kadarıyla batch-status metodunuzda kullanım dışı. Teyit işlemi nasıl yapabileceğimiz hakkında yardımcı olabilirsiniz seviniriz.

İyi çalışmalar dilerim.”

Çiçek sepeti tarafından mailime yanıt olarak aşağıdaki maili aldım.

“Merhaba,

Batch-status metodu kullanılabilir durumda, aldığınız bir hata varsa gönderirseniz ürün tarafı ile ilgilenen arkadaşlarımız kontrol etsinler. Ürün tarafından Neşe Hanımı ekliyorum maile.

Teşekkürler.”

Maili aldıktan sonra çiçeksepeti dokümanına girdiğimde batch-status servisine erişim vardı.

```
Thread.Sleep(2000);

var client2 = new RestClient("https://sandbox-apis.ciceksepeti.com/api/v1/Products/batch-status/" + details["batchId"].ToString());
client2.Timeout = -1;
var request2 = new RestRequest(Method.GET);
request2.AddHeader("x-api-key", "bv0k6tFB3x9twyUx2HmRq56s0SxRHLM08vAHnNEY");
IRestResponse response2 = client2.Execute(request2);
var details2 = JObject.Parse(response2.Content);

if (details2["items"][0]["status"].ToString() == "Failed")
{
    foreach (var item in details2["items"][0]["failureReasons"])
    {
        MessageBox.Show(item["message"].ToString(), "Geri Bildirim");
    }
}
else
{
    MessageBox.Show("Ürün Başarıyla Eklendi.", "Geri Bildirim");
}
```

Ürünü yükleme servisinde gelen response içerisindeki batchId kodunu batch-status servisine request attım. İşlemin başarılı veya başarısız olması ile ilgili gelen response u kullanıcıya message box ile gösterdim. Requesti göndermeden önce 2sn bekletme koyma sebebim ise requesti delay koymadan gönderdiğimde processing mesajını alıyordum.

Yapılan Çalışmanın Konusu: Çiçek Sepeti Ürünlerinin Şefime Aktarımı

Bugün stajın on sekizinci iş günü. Şefim veri tabanındaki ürünü çiçek sepetine aktarımını gerçekleştirdikten sonra yazılım ekip lideri Emre Bey'e task'ın (görevin) gerçekleştirildiğine dair küçük bir sunum yapım. Emre Bey sıradaki task'ın bu işlemin tersi olan çiçek sepetindeki ürününün şefim yazılımına aktarılmasının olduğunu söyledi. Bunun için kullanabileceğim api bulmak için çiçek sepeti dokümanını açtım. "Ürün Listeleme v2" metodu işimi göreceğini düşündüm. Bu metot ile ürün aktarma işlemini gerçekleştirdim. İlk işlem olarak kullanıcının çiçek sepetinden gelen ürünlerden şefime aktarmak istediği ürünü seçmesi için önceden oluşturup şefim ürünleri listelediğim formu büyüttükten sonra yan kısmına da çiçek sepeti ürünlerini listelemek için bir listview oluşturdum.

```
Product product = new Product();
product.ReadDatabase();
listCicekSepeti.View = View.Details;
listCicekSepeti.Columns.Add("ÜRÜN ADI", 500);

var client = new RestClient("https://sandbox-apis.ciceksepeti.com/api/v1/Products");
client.Timeout = -1;
var request = new RestRequest(Method.GET);
request.AddHeader("x-api-key", "bV0k6tF83x9twyUx2HmRq56s0SxRHLm08vAHnNEY");
request.AddParameter("text/plain", "", ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
var details = JObject.Parse(response.Content);

foreach (var item in details["products"])
{
    try
    {
        int productId = product.dictionary[item["productName"].ToString()]; // Ürün şefimde yok ise catch kısmına düşecek.
        tempdict.Add(item["productName"].ToString(), 1);
    }
    catch
    {
        listCicekSepeti.Items.Add(item["productName"].ToString()).ForeColor = Color.Green;
        try
        {
            sendProduct.Add(item["productName"].ToString(), item); // Listeye eklenen ürünlerin aktarma ekranına göndermek üzere JToken
            // veri tipinde tutmak için ürün ismi ile birlikte tutuyorum.
        }
        catch
        {
            // Aynı isimli ürün yüklenmiş !
        }
    }
}
```

Çiçek sepeti ürünlerini listelerken eğer aynı isimli ürün şefimde varsa listeye eklenmemesini gerçekleştirmek için ilk adımda şefimdeki ürünlerin isimlerini almak için önceden yazmış olduğum ReadDatabase metodunu kullandım. Sonrasında çiçek sepeti ürün bilgilerini alacağım requesti attım. Listeye eklenecek ürün şefim veri tabanında mevcut ise listeye eklenmemesini sağlayacak algoritmamı yazdım.

```
private void listCicekSepeti_DoubleClick(object sender, EventArgs e)
{
    var newForm = new SefimProduct(sendProduct[listCicekSepeti.SelectedItems[0].SubItems[0].Text]);
    newForm.Show();
}
```

Listview komponent 'in double Click eventinde seçilen ürünün JToken veri tipinde nesnesini ürün aktarma formuna iletiyorum.

Yapılan Çalışmanın Konusu: Çiçek Sepeti Ürünlerinin Şefim Veritabanına Kaydı

Bugün stajın on dokuzuncu iş günü. Çiçek sepeti tarafından şefime aktara bilinir olan ürünlerin listelenmesini ve kullanıcının aktarmak istediği ürünü seçtiğinde ürün bilgilerinin aktarma form ekranına aktarılmasını sağlamıştım. Bugün ise aktarılan ürünün bilgilerini şefim veritabanına yazılma işlemini gerçekleştirdim. Aktarılabak ürünün bilgileri JToken veri tipinde parametre olarak geliyor fakat gelen response üzerinde ürüne ait KDV bilgisi gelmemekte. Bu bilgiyi ürünü şefime kaydederken kullanıcı form ekranından KDV bilgisini gireceği bir combo box ekledim.

```
1 başvuru
private void SefimProduct_Load(object sender, EventArgs e)
{
    txtBoxName.Text = prod["productName"].ToString();
    if (prod["salesPrice"].ToString().IndexOf(",")!=-1)
    {
        txtBoxPrice.Text = prod["salesPrice"].ToString();
    }
    else
    {
        txtBoxPrice.Text = prod["salesPrice"].ToString() + ",00";
    }
    cmbTax.Items.Add(8.00M);
    cmbTax.Items.Add(18.00M);
}
```

Parametre olarak gelen ürün bilgilerini kullanıcının görmesi için formdaki input field lara aktardım.

```
private void btnAdd_Click(object sender, EventArgs e)
{
    product.ReadDatabase();
    if (product.SearchId(txtBoxName.Text) == -1)
    {
        product.ProductName = txtBoxName.Text;
        product.ProductGroup = "Çiçek Sepeti";
        product.ProductCode = prod["productCode"].ToString();
        product.Price = Convert.ToDecimal(txtBoxPrice.Text);
        product.VatRate = Convert.ToDecimal(cmbTax.SelectedItem);
        product.FreeItem = 0;
        product.ProductType = "Normal";
        product.SkipOptionSelection = 0;
        product.Aktarildi = 0;
        product.Save();
    }
    product.ReadDatabase();
    SQL sql = new SQL();
    sql.OpenDatabase();
}
```

Önceden oluşturduğum Product sınıfının nesnesini tanımladım ve property bilgilerini doldurdum. Sonrasında oluşturduğum modeli save methodu ile veritabanına kaydettim. Ürüne ait olan özelliklerin tutulduğu Choice1 ve Choice2 tablolarını doldurmak için ürünün varyant özelliklerini alacağım methodu (getVariant) tanımladım.

Yapılan Çalışmanın Konusu: Çiçek Sepeti Ürünlerinin Varyant Özelliklerin Aktarılması

Bugün stajın yirminci iş günü. Ürünün şefim veritabanına aktarım işlemini gerçekleştirmiştim. Fakat ürüne ait özelliklerin şefim veri tabanında farklı tablolarda tutulduğu için özellik aktarımını bugüne bırakmışım.

```
if (attrList.Length > 1)
{
    label4.Visible = true;
    label5.Visible = true;
    label5.Text = attrList[1];
    btnAdd.Location = new Point(103, 133);
    this.Size = new Size(319, 212);
}
if (attrList.Length > 2)
{
    label6.Visible = true;
    label7.Visible = true;
    label6.Text = attrList[2];
    btnAdd.Location = new Point(103, 169);
    this.Size = new Size(319, 246);
}
```

Formun load koduna eklediğim kod bloğu sayesinde ürüne ait varyant sayısına göre form ekranında ürüne ait varyant özelliklerinin gösterimini sağlıyorum. Aynı şekilde ürün sayısına göre form boyutu ve ürün aktarma butonunun yerini set ediyorum.

Aşağıdaki kod bloğunda ise ürüne ait varyant özelliği veya özellikleri var ise Choice1 ve Choice2 tablolarına kayıt işlemini gerçekleştiriyorum.

```
if (attrList.Length>1)
{
    getChoice1Id(attrList[1]);
    if (choice1Id.Count == 0)
    {
        string CmdText = "INSERT INTO Choice1(" +
            "[ProductId],[Name],[Price],[Aktarildi]" +
            ")VALUES (@p1,@p2,@p3,@p4)";

        SqlCommand command = new SqlCommand(CmdText, sql.GetConn());
        command.Parameters.AddWithValue("@p1", product.SearchId(attrList[0]));
        command.Parameters.AddWithValue("@p2", attrList[1]);
        command.Parameters.AddWithValue("@p3", 0.0m);
        command.Parameters.AddWithValue("@p4", 0);
        command.ExecuteNonQuery();
    }
}
if (attrList.Length > 2)
{
    getChoice1Id(attrList[1]);

    foreach (var item in choice1Id)
    {
        string CmdText2 = "INSERT INTO Choice2(" +
            "[ProductId],[Choice1Id],[Name],[Price],[Aktarildi]" +
            ")VALUES (@p1,@p2,@p3,@p4,@p5)";

        SqlCommand command2 = new SqlCommand(CmdText2, sql.GetConn());
        command2.Parameters.AddWithValue("@p1", product.SearchId(attrList[0]));
        command2.Parameters.AddWithValue("@p2", item);
        command2.Parameters.AddWithValue("@p3", attrList[2]);
        command2.Parameters.AddWithValue("@p4", 0.0m);
        command2.Parameters.AddWithValue("@p5", 0);
        command2.ExecuteNonQuery();
    }
}
sql.CloseDatabase();
```

Eğer bir varyant özelliği var ise bu özelliği ürün id si ve ürün adı ile birlikte Choice1 tablosuna kaydediyorum.

Eğer ikinci bir özelliği daha var ise bu ikinci varyant özelliğini Choice1 tablosunda ilgili ürüne ait tüm kayıtların id si ile birlikte Choice2 tablosuna kaydediyorum.

Yapılan Çalışmanın Konusu: Çiçek Sepeti Aktarımı Test ve Sunumu

Bugün stajın yirmi birinci iş günü. Çiçek sepetindeki ürünü şefim veritabanına varyant özellikleri ile birlikte aktarma işlemini gerçekleştirmiştim. Bugün ise ara yüzün dinamik oluşumu ile ilgili testi ve verilerin doğru bir şekilde geçirildiğini onaylamak için veri testi yaptım. Arayüz testinde ilk olarak çiçek sepetine kayıtlı fakat şefimde olmayan ürünlerin doğru listelendiğini teyit ettim. Sonrasında ürüne ait hiç varyant özelliğinin olmadığı form ekranı, ürüne ait tek bir varyant özelliğinin olduğu form ekranı ve son olarak ürüne ait iki varyant özelliğinin olduğu form ekranı oluşumunu denedim.

The image shows three screenshots of a web form titled 'ÇİÇEK SEPETİ ÜRÜNÜ'. Each screenshot displays a different product configuration with fields for 'Ürün Adı', 'Ürün Fiyatı', 'KDV Oranı', and 'Varyant 1'. The 'Ürün Adı' field is a text input, 'Ürün Fiyatı' is a text input, 'KDV Oranı' is a dropdown menu, and 'Varyant 1' is a text input. A 'Ürünü Şefime Ekle' button is present in each form.

| Ürün Adı | Ürün Fiyatı | KDV Oranı | Varyant 1 |
|----------|-------------|-----------|-----------|
| Deneme | 3,00 | 8,00 | |
| Cotton | 8450,00 | 8,00 | Orta Boy |

This screenshot shows the 'ÇİÇEK SEPETİ ÜRÜNÜ' form with two variant fields. The 'Ürün Adı' is 'Test Urunu', 'Ürün Fiyatı' is '56,99', and 'KDV Oranı' is '18,00'. The 'Varyant 1' field is 'XS' and the 'Varyant 2' field is 'Siyah'. A 'Ürünü Şefime Ekle' button is at the bottom.

| Ürün Adı | Ürün Fiyatı | KDV Oranı | Varyant 1 | Varyant 2 |
|------------|-------------|-----------|-----------|-----------|
| Test Urunu | 56,99 | 18,00 | XS | Siyah |

Form ekranları varyant özelliğine göre dinamik bir şekilde oluşmakta. “Ürünü Şefime Ekle” butonuna tıklandığında ürün özellikleri ile birlikte veri tabanına doğru bir şekilde kaydedildiğini teyit ettim.

Test işlemlerim sonrasında yazılım ekip lideri Emre Bey’e gerçekleştirdiğim task ile ilgili bir sunum yaptım. Sonrasında Emre Bey ile birlikte hangi işlevi gerçekleştireceğim ile ilgili araştırma yapmak için çiçek sepeti web servis dokümanını inceledik. Sonrasında Emre Bey yeni görevimin ürün fiyat güncelleme işlemi olduğunu söyledi.

Yapılan Çalışmanın Konusu: Çiçek Sepeti Ürün Fiyat Güncelleme Başlangıç

Bugün stajın yirmi ikinci iş günü. Dün verilen görevim olan Çiçek sepetindeki ürünün fiyat güncelleme işlemini gerçekleştirmek için. Çiçek sepeti web servis dokümanındaki ilgili servis metodunun sayfasını açtım. “Stok ve Fiyat Güncelleme V2” metodu PUT request tipinde. Verilmesi gereken parametrelerin açıklaması ve veri tipleri şu şekilde;

| Parametre | Açıklama | Tip | Zorunlu mu? |
|---------------|-----------------------------|---------|-------------|
| stockCode | Tedarikçi ürün varyant kodu | string | Evet |
| StockQuantity | Ürün stok adedi | integer | Hayır |
| listPrice | Üstü çizili fiyat | double | Hayır |
| salesPrice | Satış fiyatı | double | Hayır |

Response olarak aynı ürün yükleme response undaki gibi batch-id dönmekte. Bu batch-id’yi kullanarak batch-status metodu ile işlemin gerçekleşip gerçekleşmediğini kontrol edilecek.



Şefim veri tabanındaki ürünlerin çiçek sepetine aktarılması için listelediğim listview’i biraz değiştirdim. Eğer ürün ilk defa çiçek sepetine aktarılacaksa ürün ismi kırmızı bir font ile yazılacak eğer ürün çiçek sepetine aktarılmış sadece fiyat güncellemesi yapılacaksa mavi bir font yazı ile yazılacak şekilde ayarladım.

```
foreach (var item in product.dictionary)
{
    if (tempdict.Remove(item.Key) == false)
    {
        listSefim.Items.Add(item.Key).ForeColor = Color.Red;
    }
    else
    {
        listSefim.Items.Add(item.Key).ForeColor = Color.Blue;
    }
}
```

“tempdict” çiçek sepeti ürünlerini listelerken doldurduğum içinde hem çiçek sepetindeki hem de şefimdeki ürünlerin isimlerini tutmaktadır.

Yapılan Çalışmanın Konusu: Ürün Fiyatı Güncelleme Ekranı Oluşturulması

Bugün stajın yirmi üçüncü iş günü. Dün şefim ürünlerinden hangisinin yeni eklenebilir, hangisinin eklenmiş sadece fiyat güncellemesi yapılabilir olduğunu ayırt etme işlemini gerçekleştirip buna göre listview içindeki ürün ismini renklendirmiştım. Bugün kullanıcının listview üzerinden seçtiği ürünün rengine göre yeni ürün ekleme veya fiyat güncelleme ara yüzüne aktarımını gerçekleştirdim. Fiyat güncelleme ekranı için yeni bir form oluşturmak yerine ürün yükleme ekranını dinamik hale getirip aynı anda ürün fiyat güncelleme yapabilmesine olanak sağladım. Bunu gerçekleştirmek için ürün yükleme form ekranına bir parametre daha ekledim. Bu parametre işlemin ürün yükleme mi yoksa ürün güncelleme mi olduğunu tutan bir integer değişken. Listview in double click eventi kod bloğunu değiştirip şöyle bir yapı kurdum;

```
private void listProduct_DoubleClick(object sender, EventArgs e)
{
    if (listSefim.SelectedItems[0].ForeColor== Color.Blue)
    {
        var newForm = new CicekSepetProduct(listSefim.SelectedItems[0].SubItems[0].Text, dictionary,1);
        newForm.Show();
    }
    else
    {
        var newForm = new CicekSepetProduct(listSefim.SelectedItems[0].SubItems[0].Text, dictionary,0);
        newForm.Show();
    }
}
```

```
public CicekSepetProduct(string name, Dictionary<string, int> dictionary,int isUpdate)
{
    InitializeComponent();
    if (isUpdate == 1)
    {
        settingsUpdateScreen();
    }
    else
    {
        settingsInsertScreen();
    }
}
```

Form sınıfının constructor metodunda parametre olarak gelen “isUpdate” değişkenine göre form ekranının dinamik ayarlarını yaptığım metotları çağırıyor. Bu metotlar;

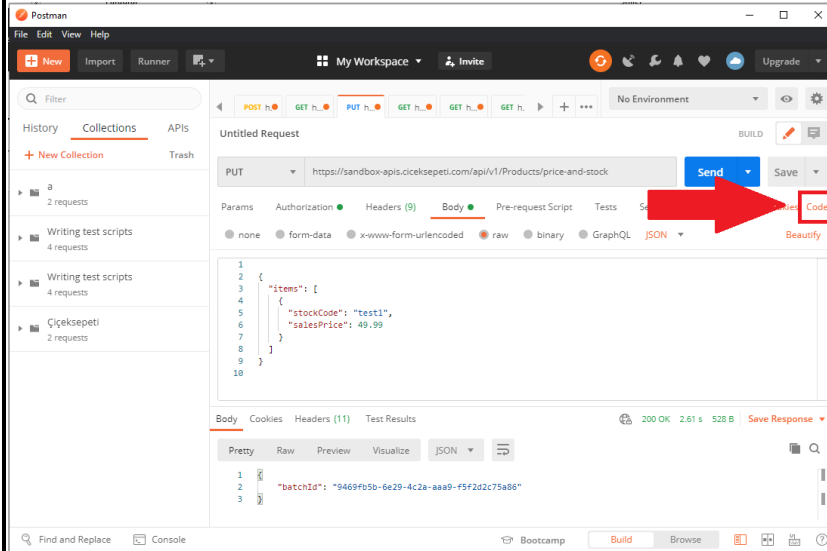
```
1 başvuru
private void settingsInsertScreen()
{
    btnUpdate.Enabled = false;
}

1 başvuru
private void settingsUpdateScreen()
{
    this.Size = new Size(345, 600);
    this.Text = "ÇİÇEK SEPETİ FİYAT GÜNCELLEME";
    textBoxDescription.Enabled = false;
    textBoxImageUrl.Enabled = false;
    cmbCategory.Enabled = false;
    cmbMessageType.Enabled = false;
    cmbType.Enabled = false;
    btnPreview.Enabled = false;
    btnSendData.Enabled = false;
}
```

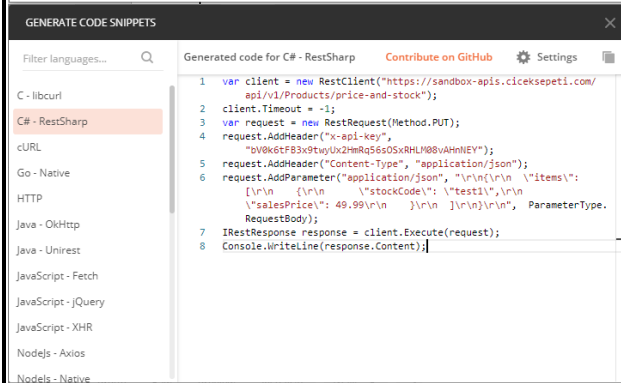
Parametre olarak “1” yani bir update işlemi olduğunu belirten parametre geliyor ise formun sağ tarafında ürüne ait özelliklerin seçilmeyeceği için formun küçültmesini, başlık değiştirilmesi, kullanılmayacak input field ların enabled değerinin false yapılması ve ürün yükleme metodunun devre dışı bırakılmasını sağlıyorum. Aynı şekilde parametre 0 olarak gelirse ürün güncelleme butonunu devre dışı bırakılmasını sağlıyorum.

Yapılan Çalışmanın Konusu: Ürün Fiyatı Güncelleme Gerçekleştirilmesi ve Testi

Bugün stajın yirmi dördüncü iş günü. Dün kullanıcının listview üzerinden seçtiği ürüne göre formun dinamik düzenlenmesini sağlamıştım. Bugün kullanıcının seçtiği ürünün fiyat güncellemesini web servisi kullanarak işlemini gerçekleştirdim.



Postman üzerinden bir request denedim ve başarı bir şekilde response aldıktan sonra postman üzerinden requesti farklı dillerde koda çeviren aracını kullandım ve otomatik bir şekilde request kod bloğunu elde ettim.



Postman'dan oluşturduğum request kod bloğunu kendi programıma uyarlayıp güncelleme butonunun click eventi ile kod bloğumu çalıştırdım.

```
1 başvuru
private void UpdateProduct()
{
    var client = new RestClient("https://sandbox-apis.ciceksepeti.com/api/v1/Products/price-and-stock");
    client.Timeout = -1;
    var request = new RestRequest(Method.PUT);
    request.AddHeader("x-api-key", "bV0k6tFB3x9twyUx2HmRq56sOSxRHLm08vAHnNEY");
    request.AddHeader("Content-Type", "application/json");
    request.AddParameter("application/json", "\r\n{\r\n  \"items\": [\r\n    {\r\n      \"stockCode\": \"\" + getProductCode() +  
      \"\", \r\n      \"salesPrice\": \" + txtBoxPrice.Text.Replace(",", ".") + \"\r\n    } \r\n  ] \r\n}\r\n", ParameterType.RequestBody);
    IRestResponse response = client.Execute(request);
    MessageBox.Show("İşlem Başarılı!");
}
1 başvuru
private void btnUpdate_Click(object sender, EventArgs e)
{
    UpdateProduct();
}
}
```

İşlemi gerçekleştirdikten sonra modülün düzgün olarak gerçekleştiğini teyit etmek için black box test yaptım. Modül doğru bir şekilde ürünün fiyat bilgisini güncellemekte. Sonrasında yazılım ekip lideri Emre Bey'e modülün stabil çalıştığını gösteren küçük demo gerçekleştirdim. Şu ana kadar implement ettiğim çiçek sepeti web servislerinin bilgisini verdim. Sonrasında gün sonuna kadar yazdığım kodlarda iyileştirmeler yaptım.

Yapılan Çalışmanın Konusu: Sipariş Kargo Süreçlerinin Planlanması ve Analizi

Bugün stajın yirmi beşinci iş günü. Geçtiğimiz staj günlerinde entegre projesinin birçok modülünü oluşturdum. Yazılım ekip lideri Emre Bey ile geliştirmiş olduğum modüller üzerinden ufak bir göz gezdirdikten sonra modüller üzerindeki düzeltmeleri konuştuk. Sonrasında staj bitiş gününün yaklaşması ile son geliştireceğim modüle karar verdik. Bu modül kargo süreçlerinin ile ilgili gelen siparişin çiçek sepeti tarafına siparişin kargoya verildi bilgisini gönderecek. Çiçek sepeti kargo süreç yönetimi üç farklı başlık altına incelenmekte bunlar;

1. Tedarikçinin Çiçeksepeti Kargo Entegrasyonunu Kullanması
2. Tedarikçinin Kendi Kargo Entegrasyonunu Kullanması
3. Tedarikçinin Kendi Kargo Entegrasyonunu Kullanması Gönderi Takibini Çiçek sepetinin Yapması

Ben bu başlıklardan sadece ikinci başlığı implement edeceğim. Bunun için ikinci başlığın dokümanını açtım. Request parametrelerin açıklama ve veri tipleri şu şekilde;

| Parametre | Açıklama | Tip | Zorunlu mu? |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------|
| orderId | İşlem yapmak istediğiniz alt sipariş numarası | integer | Evet |
| orderIdStatus | Siparişin güncel statü id'si. Id karşılıkları aşağıda iletilmiştir. | integer | Evet |
| businessId | Çalıştığınız kargo firması aşağıda verilen listede yoksa çiçeksepeti bayi sorumluları ile iletişime geçip tanımlama yaptırabilirsiniz. Siparişi hangi kargo firmasıyla çıkaracaksınız o firmanın id'sini iletmeniz | integer | Sipariş kargoya verildi durumuna çekilecekse Evet, diğer durumlarda Hayır. |
| shipmentNumber | Siparişin kargo tarafından size verilen takip numarası | integer | Sipariş kargoya verildi durumuna çekilecekse Evet, diğer durumlarda Hayır. |
| shipmentTrackingUrl | Siparişin kargo tarafından size verilen takip linki | integer | Sipariş kargoya verildi durumuna çekilecekse Evet, diğer durumlarda Hayır. |
| receiverName | Sipariş statüsü teslim edildi olarak gönderilecekse siparişi teslim alan kişi adı gönderilmelidir. | integer | Zorunlu değil. Sipariş teslim edildi durumuna çekilirken gönderilmezse default değer olan siparişin alıcısının adı basılır. |
| deliveryTime | Sipariş statüsü teslim edildi olarak gönderilecekse saat kaçta teslim edildiği gönderilmelidir. | string, date | Zorunlu değil. Sipariş teslim edildi durumuna çekilirken gönderilmezse istediğin atıldığı tarih ve saat basılır. |

Atılan request karşılığında gelen response parametrelerin açıklama ve veri tipleri şu şekilde;

| Parametre | Açıklama | Tip |
|-----------|-----------------------------------------------|---------|
| orderId | İşlem yapmak istediğiniz alt sipariş numarası | integer |
| isSuccess | İşlem başarılı / başarısız | string |
| message | İşlem başarısız ise hata açıklaması | string |

Doküman üzerinden request ve response yapısını inceledikten sonra. Şefim yazılımı üzerinde bakım ve geliştirme yapan Metin Bey siparişin kargo bilgisinin gönderilme ekranına benim yapmış olduğum client üzerinden değil de şefim sipariş ekranı üzerinden geçiş işleminin yapılmasının iyi olacağını söyledi. Çözüm olarak benim oluşturduğum client komut satırı parametresi ile açıldığında parametre olarak gelecek sipariş id'si ile birlikte kargo ekranı açılacak. Özetle benim oluşturduğum client açıldığı anda herhangi bir komut satırı parametresi var ise kargo ekranı gelecek eğer bir parametre yok ise şu an çalıştığı gibi login ekranı gelecek.

Yapılan Çalışmanın Konusu: Kargo Ekranı ve Veri Giriş Alanlarının Oluşturulması

Bugün stajın yirmi altıncı iş günü. Dün kargo ekranına geçiş işleminin ne şekilde olacağını Metin Bey ile konuşmuştuk. Bugün ilk işim Kargo adında bir form oluşturup aldığı parametrelere siparişin id bilgisini ekledim.

Form oluşturulurken sipariş no bilgisi şefim yazılımı ara yüzünden Metin Bey gönderecek fakat kullanıcı görüntüleyebilmesi için metin kutusu üzerinde sipariş no bilgisini yazdırdım. Kullanıcı kargo firması seçebileceği bir combo box ekledim. Requeste gönderilecek olan takip no ile takip linki bilgisini elde etmek için text boxlar oluşturdum. Diğer komponent leri sadece bilgi gösterme amaçlı oluşturdum.

cargoBusinessId Karşılıkları

1: Mng Kargo 2: Yurtiçi Kargo 25: Sürat Kargo 43: Aras Kargo 44: PTT Kargo 45: UPS Kargo 46: Horoz Lojistik

```
1 başvuru
private void fillCmb()
{
    dicCargoBusiness.Add("Mng Kargo", 1);
    dicCargoBusiness.Add("Yurtiçi Kargo", 2);
    dicCargoBusiness.Add("Sürat Kargo", 25);
    dicCargoBusiness.Add("Aras Kargo", 43);
    dicCargoBusiness.Add("PTT Kargo", 44);
    dicCargoBusiness.Add("UPS Kargo", 45);
    dicCargoBusiness.Add("Horoz Lojistik", 46);
    foreach (var item in dicCargoBusiness.Keys)
    {
        cmbBusiness.Items.Add(item);
    }
}
```

“fillCmb” adlı metod ile web servis doküman içindeki yukarda görüldüğü gibi kargo firmalarının id bilgilerini kullanarak request parametrelerine eklemek üzere kullanıcıdan kargo firmalarının id bilgisini alabileceğim combo box’ı dolduruyorum. Bu metodu formun load eventi içinde çağırıyorum.

Yapılan Çalışmanın Konusu: Kargo Ekran Şefim Bağlantısı ve Bilgilerin Ekrana Getirilmesi

Bugün stajın yirmi yedinci iş günü. Dün kargo ekranını oluşturup gerekli input field ları ve siparişe ait ürünlerin listeleneceği listview eklemiştim. Fakat bu oluşturduğum formu kullanmaya kapalı. Şefim yazılımı tarafından bu forma herhangi bir erişim yok. Benim oluşturduğum client'ı çalıştırma isteği geldiğinde kargo formunu ekrana getirmek için program ilk başlangıcında çalışan Program.cs altındaki main fonksiyonunu aşağıdaki gibi güncelledim.

```
[STAThread]
0 başvuru
static void Main(string[] args)
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    if (args.Length == 0)
    {
        Application.Run(new Form1());
    }
    else if (args.Length == 1)
    {
        Application.Run(new Kargo(args[0]));
    }
}
```

Eğer komut satırı parametresi var ise o parametreye birlikte kargo form ekranım açılacak fakat herhangi bir parametre yok ise standart olan login ekranım açılacak.

```
PhoneOrderHeader.getAddress_HeaderId(orderId);
txtAddress.Text = PhoneOrderHeader.Address;
```

```
public void getAddress_HeaderId(string orderId)
{
    SQL sql = new SQL();
    using (sql.GetConn())
    {
        SqlCommand command = new SqlCommand(
            "select Address,HeaderId from PhoneOrderHeader WHERE OrderNo=" + orderId.ToString(),
            sql.GetConn());
        sql.OpenDatabase();

        SqlDataReader reader = command.ExecuteReader();

        if (reader.HasRows)
        {
            while (reader.Read())
            {
                HeaderId = Int16.Parse(reader["HeaderId"].ToString());
                Address = reader["Address"].ToString();
            }
        }
        else
        {
            Console.WriteLine("No rows found.");
        }
        reader.Close();
    }
}
```

Bu kod bloğu formun load eventinde çalışmakta siparişe ait adres bilgisini ve siparişe ait ürünleri çekmek için HeaderId bilgisini elde etmeyi sağlıyor. PhoneOrderHeader sınıfında yazdığım metodu çağırıyor.

```
var products = Bill.getProductList(PhoneOrderHeader.HeaderId.ToString());
listOrderItem.View = View.Details;
listOrderItem.Columns.Add("", 500);
foreach (string p in products)
{
    listOrderItem.Items.Add(p);
}
```

Az önce elde ettiğim HeaderId verisini kullanarak siparişe ait ürün adlarının listview'e aktarıyorum. Ürün adlarını Bill sınıfında tanımladığım "getProductList" adlı metot ile elde ediyorum.

Yapılan Çalışmanın Konusu: Siparişin Kargoya Verildi Bilgisinin Gönderilmesi

Bugün stajın yirmi sekizinci iş günü. Kargo verildi bilgisinin gönderilmesi ile ilgili hazırlıkları ve şefim bağlantısını tamamlamıştım. Bugün request gönderim işlemi tamamladım. Şefim tarafından gelen orderId bilgisini kullanarak orderItemId bilgilerini elde ettim. Bunun sebebi ise kargo bilgisi gönderilirken sipariş id' si yerine siparişe ait ürünlerin id bilgisi gönderilmektedir.

```
private void getOrderItemsId()
{
    var client = new RestClient("https://sandbox-apis.ciceksepeti.com/api/v1/Order/GetOrders");
    client.Timeout = -1;
    var request = new RestRequest(Method.POST);
    request.AddHeader("x-api-key", "bV0k6tFB3x9twyUx2HmRq56sOSxRHLm08vAHnNEY");
    request.AddHeader("Content-Type", "application/json");
    request.AddParameter("application/json", "{\r\n  \"startDate\": \"2020-07-19\", \r\n  \"endDate\": \"2020-08-18\", \r\n  \"pageSize\": 10, \" +
    \"\r\n  \"page\": 0, \r\n  \"statusId\": 1\r\n}", ParameterType.RequestBody);
    IRestResponse response = client.Execute(request);
    var details = JObject.Parse(response.Content);

    foreach (var item in details["supplierOrderListWithBranch"])
    {
        if ((item["orderId"].ToString()) == orderId)
        {
            itemIdList.Add(Int32.Parse(item["orderItemId"].ToString()));
        }
    }
}
```

```
2 başvuru
public class Rootobject2
{
    1 başvuru
    public Orderitem[] orderItems { get; set; }
}

5 başvuru
public class Orderitem
{
    1 başvuru
    public int orderItemId { get; set; }
    1 başvuru
    public int orderItemStatusId { get; set; }
    1 başvuru
    public int cargoBusinessId { get; set; }
    1 başvuru
    public string shipmentNumber { get; set; }
    1 başvuru
    public string shipmentTrackingUrl { get; set; }
}
```

Bu tanımladığım metot ile orderId 'ye ait orderItemId leri tanımlamış olduğum listeye aktarıyorum.

Requeste ekleyeceğim modelleri otomatik olarak web servis dokümandaki örnek response json çıktısını kullanarak oluşturdum.

```
Orderitem[] orderitems = new Orderitem[itemIdList.Count];

for (int i = 0; i < itemIdList.Count; i++)
{
    Orderitem orderitem = new Orderitem();
    orderitem.orderItemId = itemIdList[i];
    orderitem.orderItemStatusId = 5;
    orderitem.cargoBusinessId = dicCargoBusiness[cmbBusiness.SelectedItem.ToString()];
    orderitem.shipmentNumber = txtShipmentNo.Text;
    orderitem.shipmentTrackingUrl = txtTrackingUrl.Text;
    orderitems[i] = orderitem;
}

Rootobject2 rootobject2 = new Rootobject2();
rootobject2.orderItems = orderitems;
var json = JavaScriptSerializer().Serialize(rootobject2);

var client = new RestClient("https://sandbox-apis.ciceksepeti.com/api/v1/Order/statusupdatewithsupplierintegration");
client.Timeout = -1;
var request = new RestRequest(Method.PUT);
request.AddHeader("x-api-key", "bV0k6tFB3x9twyUx2HmRq56sOSxRHLm08vAHnNEY");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("application/json", json, ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Oluşturmuş olduğum modellere kullanıcıdan aldığım bilgileri ekledikten sonra Serialize edip requestimin yolladım.

Yapılan Çalışmanın Konusu: Projenin Pilot Mağaza Testinin Yapılması

Bugün stajın yirmi dokuzuncu iş günü. Geçtiğimiz staj günlerinde yazdığım kodları tekrardan gözden geçirip iyileştirmeler yaptım. Sonrasında modüller üzerinde testlerimi gerçekleştirdikten sonra Çiçek sepeti destek ekibine mail attım. Mail içerisinde geliştirmelerimizi tamamladığımızı ve süreç içerisinde geliştirme yapılması için test ortamı sunucusunu kullandık. Programımızı gerçek verilerin olduğu ortamda denemek için pilot mağaza üzerinde çalışmak istediğimizi belirttim. Çiçek sepeti yetkilileri pilot mağazaya ait api-key i bize gönderdi. Bizde request' lerde kullandığımız api-key bilgilerini değiştirerek pilot mağaza üzerinde testlerimizi gerçekleştirdik. Geliştirme ortamımızdaki verilerin tutarlı olmaması dolayından geliştirdiğim kodların bir kısmında pilot mağaza testinde hatalar ile karşı karşıya kalındı. Bu belirlediğim hataları düzeltecek zamanım kalmadığı için hataların neyden kaynaklandığını belirttiğim liste hazırladım ve bu listeyi yazılım ekip lideri Emre Bey' e verdim.

Yapılan Çalışmanın Konusu: Projenin Teslim ve Devir Yapılması

Bugün stajın son iş günü. Proje zamanımdaki son gün olması nedeniyle benden sonra projenin geliştirilmesi ve hataların çözülmesi için yazdığım tüm kod bloklarını sırasıyla Metin Bey'e aktardım. Mimarının ne şekilde olduğunu, kod bloğundaki amacın ne olduğunu, nasıl çalıştığının tüm detaylarını aktardıktan sonra private oluşturduğum github reposunu aktardım. Sonrasında yazılım ekip lideri Emre Bey Vega yazılım şirketinin gelişmekte olduğu diğer yazılımlar hakkında bilgi alıp kendimde fikir sahibi olmam için diğer yazılım geliştirici bireyler tarafından geliştirmiş olduğu sistem veya yazımların işleyişi hakkında bilgi aldım. Ekstra bir bilgi olarak Google Play Store ve App Store adındaki mobil uygulama marketlerinde yayınlanan yazılımın nasıl güncelleme yapıldığı hakkında yazılım geliştiricisi Mert Bey tarafından bilgi aldım. Günün son saatleri yaklaşırken staj sorumlum Emre Bey ile birlikte staj dosyası ve staj sicil fişini ne şekilde doldurulacağını konuştuk. Sonrasında son staj günümün yapılan çalışma ile ilgili sayfasını doldurmaya başladım. Böylelikle stajım son buldu.