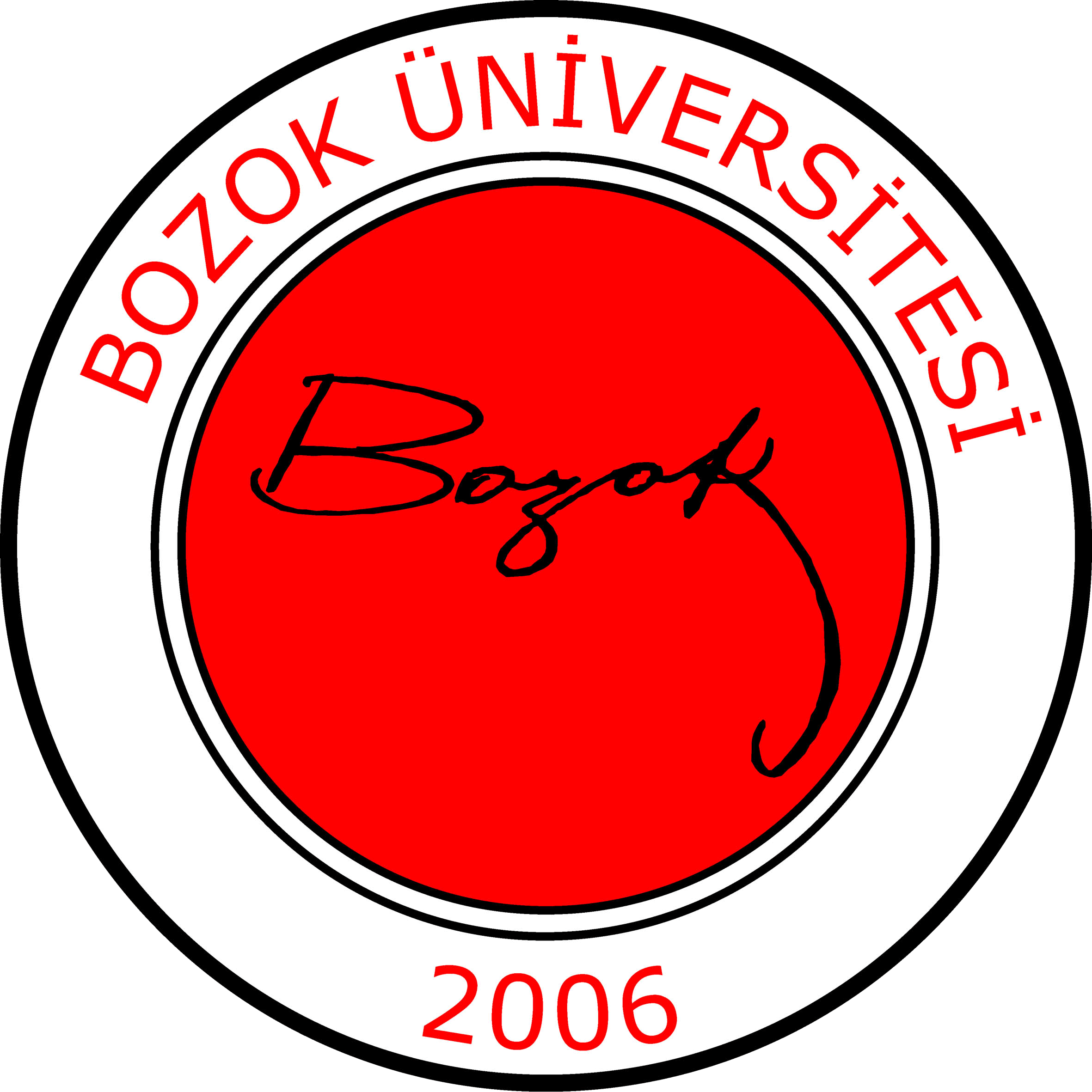
****

**BOZOK ÜNİVERSİTESİ**

**MÜHENDİSLİK MİMARLIK FAKÜLTESİ**

**STAJ DEFTERİ**

****

**YOZGAT BOZOK ÜNİVERSİTESİ MÜHENDİSLİK MİMARLIK FAKÜLTESİ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ÖĞRENCİNİN** | Bölümü | BİLGİSAYAR MÜHENDİSLİĞİ | |  |
| Numarası | 16008116032 | |
| Adı ve Soyadı | Batuhan Özkoç | |
| Fakülteye girdiği yıl | 2016 | |
| Bu çalışmayı kaçıncı stajı olarak yaptığı | 1.STAJ  YAZILIM | 2. STAJ  DONANIM |
| Staj başlama tarihi | 10/07/2019 | |  |
| Staj bitiş tarihi | 07/08/2019 | |  |
| Staj iş günü | 20 iş günü | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STAJ YAPILAN** | FİRMA VEYA KURUMUN | ADI | YORGLASS CAM SANAYİ | |
| ADRESİ |  | |
|  | |
|  | |
| Firma veya kurum adına defteri tasdik eden bilgisayar mühendisinin  Adı ve Soyadı:  ………………….………………….……………  Ünvanı: ……………………………………… | | Yukarıda ismi yazılı ve fotoğrafı bulunan öğrencinin Firmamızda/Kurumumuzda 20 iş günü staj yaptığını ve bu defteri kendisinin tanzim ettiğini beyan ve tasdik ederim. | |
| **KURUM MÜHRÜ / FİRMA KAŞESİ** **VE**  **İMZA:** | **TARİH** :  ……./……/………. |

|  |  |  |  |
| --- | --- | --- | --- |
| **BÖLÜM STAJ KOMİSYONUNUN DEĞERLENDİRMESİ** | | | |
| Yapılan staj, 20 iş günlü ’nci staj çalışması olarak  **BAŞARISIZ**  **BAŞARILI** bulunmuştur. | | | **TARİH**  ……./……/……… |
| **STAJ KOMİSYONU** | | | |
| **BAŞKAN** | **ÜYE** | **ÜYE** | |

**STAJIN GÜNLERE DAĞITIM ÇİZELGESİ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Öğrencinin çalıştığı günler** | | **Öğrencinin çalıştığı konular** | |
| **1** | **10 / 07 / 2019** | **ERP, eBA nedir?** | |
| **2** | **11 / 07 / 2019** | **MES nedir?** | |
| **3** | **12 / 07 / 2019** | **Yapılacak Proje/Projelerin Konusu (MES ve Stok Takip)** | |
| **4** | **16 / 07 / 2019** | **Veri tabanı tablo tasarımı ve ilişkileri** | |
| **5** | **17 / 07 / 2019** | **Veri tabanı tablo tasarımı ve ilişkileri** | |
| **6** | **18 / 07 / 2019** | **Veri tabanı tablo tasarımı ve ilişkileri** | |
| **7** | **19 / 07 / 2019** | **Veri tabanı tablo tasarımı ve ilişkileri** | |
| **8** | **22 / 07 / 2019** | **Veri tabanı tablo tasarımı ve ilişkileri** | |
| **9** | **23 / 07 / 2019** | **Veri tabanı sınıf ve SQL işlemlerinin hazırlanması** | |
| **10** | **24 / 07 / 2019** | **Admin ve Operatör Sınıflarının hazırlanması** | |
| **11** | **25 / 07 / 2019** | **Giriş ekranlarının tasarımı ve kodlarının yazılması** | |
| **12** | **26 / 07 / 2019** | **Ana menü ve içeriklerinin hazırlanması** | |
| **13** | **29 / 07 / 2019** | **Operatör listesi, operasyon listesi kodlarının yazılması** | |
| **14** | **30 / 07 / 2019** | **Makinelerin listesi, duruş listesi kodlarının yazılması** | |
| **15** | **31 / 07 / 2019** | **Alıcı, ürün listesi, sipariş formlarının kodlarının yazılması** | |
| **16** | **01 / 08 / 2019** | **Stok takip projesinin yapısı** | |
| **17** | **02 / 08 / 2019** | **Stok takip projesinin veri tabanı** | |
| **18** | **05 / 08 / 2019** | **Stok takip projesinin kodlarının yazılması** | |
| **19** | **06 / 08 / 2019** | **Stok takip projesinin test aşamaları** | |
| **20** | **07 / 08 / 2019** | **Stok takip projesinin test aşamaları ve Kaynak Kodları** | |
| **Staj Amiri**  (Adı Soyadı, Mühür/Kaşe ve imza) | | | **Öğrenci**  (Adı Soyadı ve imza) |

|  |  |  |
| --- | --- | --- |
| STAJ YERİ TANITIMI VE STAJ PROJESİ İLE İLGİLİ GENEL BİLGİLERStaj Yeri Tanıtımı Yorglass işletmesinin Bilgi Teknolojileri bölümünde toplam 13 kişi çalışmaktadır. Benim bulunduğum departmanda 3 kişi çalışıyor. Burada 1 İş Çözümleri Uzmanı, 1 İş Çözümleri Uzman Yardımcısı ve Altyapı-Operasyonlar Uzmanı bulunmaktadır. Buradaki mühendisler Üretim-Yönetim Sistemlerinin veri tabanlarını denetliyorlar ve ihtiyaca göre rapor çıkarıyorlar. Sürekli üretim-yönetimini kontrol ediyorlar ve buna göre yeni çözümler üreterek, modüller ekliyorlar. 1.1 Firma Hakkında Bilgi Yorglass Cam San. Ve Tic. A.Ş. bir cam fabrikasıdır. Bolu, Gebze, Manisa, İzmir ve Eskişehir’de fabrikaları bulunmaktadır. İstanbul’da ofisleri vardır. Genel merkezi Manisa’dadır. Bu firmanın fabrikalarında plaka halinde cam satışı ve cam işlemesi yapılmaktadır. 1.2 Projede Kullanılan Yazılım/Donanım Araçları Bu staj süresince yapılan iki projede C#, Microsoft SQL ve Ironbarcode eklentisi kullanılmıştır. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**: STAJ YERİ TANITIMI  VE STAJ PROJESİ İLE İLGİLİ  GENEL BİLGİLER | **YAPILDIĞI TARİH**  **10 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| EBA nedir? eBA, İş Akışı Yönetim Sistemidir. İş süreçlerinin standardizasyonu ve otomasyonunu sağlayarak süreçlerin performansını iyileştirir. Süreç maliyetlerini düşürerek kurumsal kaynakların etkin kullanımına katkıda bulunur. Faydaları nedir? İş süreçlerinin hızlı bir şekilde elektronik ortamda otomasyonu yapılır. Kurumsal uygulamalarda hızlı ve kolay entegrasyon yapılır. Süreç maliyetlerinin azalmasını ağlar. Süreç odakları iş yapma anlayışı ortaya çıkarır. İş süreçlerinin standardizasyonu ve takip edilebilirliğini arttırır. Sürükle bırak teknolojisi ile kullanıma kolaylık sağlar. Minimum kod yazım ihtiyacı sağlar ve bu sayede daha az hata ortaya çıkar. Rol bazlı iş akışlarını yönetir. Örneğin bir işletmede çalışan birisi iş toplantısı için, işletmenin başka bir şehirdeki adresine gitmek için uçak bileti veya tren bileti alırken eBA üzerinden başvuru yapıyor ve bu başvuru çalışanın yöneticisine gidiyor. Onay aldığı takdirde, çalışanın isteği kabul edilmiş oluyor ve ücretleri karşılanıyor.    Şekil 1. eBA Örnek Resim | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**:  eBA nedir? | **YAPILDIĞI TARİH**  **10 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 3. ERP nedir? ERP, Türkçe karşılık olarak kurumsal bir şekilde yapılan kaynak planlaması anlamına gelmektedir. İşletmelerin kaynak planlaması yapması ERP şeklinde ifade edilebilmektedir.  Genel anlamda ifade etmek gerekirse ERP, işletmelerin ürünleri veya hizmetleri üretebilmeleri için gerek duydukları iş gücünü, makineleri, malzemeleri verimli bir şekilde kullanabilmesini mümkün kılan bir sistemdir. Sistemin kullanılmasıyla birlikte işletmeler farklı uygulama detaylarını tek bir pencere üzerinden görebilmekte ve aynı şekilde de kontrol edebilmektedir.  ERP üzerinde farklı farklı modüller mevcuttur. Bu modüller sonradan eklenebilir, değiştirilebilir şekilde tasarlanmıştır. Her firma kendi ERP programları üzerinde farklı modüllere sahiptir.  ERP’de örnek olarak Malzeme Yönetimi, Satış ve Dağıtım Yönetimi, Kalite Yönetimi, İnsan Kaynakları Yönetimi, Finans Yönetimi, Üretim Planlama ve Kontrol Yönetimi, Tedarik Zinciri Yönetimi, Satın Alma Yönetimi, Müşteri İlişkileri Yönetimi, Proje Yönetimi gibi modüller bulunabiliyor.    Şekil 2. ERP Örnek Ticaret Modülü | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  ERP nedir? | **YAPILDIĞI TARİH**   1. **/ 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 4. MES (Manufacturing Execution System) nedir? Türkçe karşılığı Üretim Yönetim Sistemidir. Fabrikalarda eş zamanlı bilgileri toplamaya yardımcı olur. Aynı zamanda iş emirlerini elle teslim yerine program üzerinden yapmaya yardımcı olur. Örnek verirsek benim çalıştığım işletmede 300 tane A şekilli cam üretilmesi için İş Emri verilmesi gerekiyor. Bunu evraklar veya insan gücü aracılığıyla söylemek yerine program üzerinden bildiriyorlar. Bu sayede iş emirleri canlı olarak sisteme düşüyor ve yapılması gereken fabrika hattına iletiliyor. Örneğin Hat-3A’da 450 adet BC-4502 kodunda cam üretimi yapılacak diye bir emir geldiğinde, Hat-3A’nın bilgisayarlarına bu iş emri düşüyor ve işleme koyuyorlar.  MES sisteminin en büyük yararlarından birisi üretimi canlı olarak izleyebilmesidir. X adet üretim istenen bir makinede, sistem her üretimi banttan geçirdiği zaman sensörler sayesinde sayım yapıyor ve bu bilgileri veri tabanına kaydediyor. Örnek verecek olursak 450 adet üretimi istenen BC-4502 kodlu cam üretiminde 447 adet üretilmiş, 3 adet hurdaya çıkmıştır.  Performans takibinde ise, program makinelerin saatlik ürettiği malzemeyi, hurdaya çıkan malzemeyi, üretim gecikmesinin takibini yaparak yüzdelik dilimlerini canlı olarak ekrana verir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  MES Nedir? | **YAPILDIĞI TARİH**  **11 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 5. MES Modülleri nedir? Üretim Yönetim Sistemi programında Rapor Modülü, Operatör Kontrol Modülü, Makine Listesi Modülü, Alıcı Listesi Modülü, Ürün Listesi Modülü, Üretim Raporu Modülü, Sipariş Listesi Modülü, Operasyon Listesi Modülü, Duruş Listesi Modülü, Operatör Listesi Modülü ve Planlama Modülü vardır. Bunlar eklenebilir veya değiştirilebilir. 5.1 Rapor Modülü nedir? Rapor modülünde; fabrikanın belirtilen tarih aralığındaki üretim sayısını, performans analizlerini veren bir modüldür. Bu sayede bir ayda, bir haftada, bir günde ne kadar üretim yapıldığı, kaç tane malzemenin hurdaya çıktığı takip edilebilir. 5.2 Makine Listesi, Alıcı Listesi ve Sipariş Listesi modülü nedir? Makine Listesi modülünde; üretim yönetim sisteminde fabrikadaki makinelerin isimlerinin ve kullanıldığı hatların görüntülendiği modüldür. Burada hangi makinenin ne iş yaptığını, makinenin kodunu ve kullanıldığı bantı öğrenebiliriz.  Alıcı Listesi modülünde; üretilen malzemenin kimler tarafından alındığını ve alan kişilerin şirket numaralarının vesaire bulunduğu kısımdır. Burada alıcının ismi veya şirket numarası gibi bilgileri görünür. Sipariş Listesi modülünde, gelen siparişler görüntülenir. Siparişi veren kişilerin numaraları, verdikleri ürün siparişleri ve adet sayıları yazar. Ürün Listesi, Operasyon Listesi, Duruş Listesi modülü nedir? Ürün Listesi modülünde; ürünlerin kodları, adı, açıklaması ve en-boy bilgileri bulunur. Örnek verecek olursak; 450B-27 kodlu ürünün adı Kesik Uzanımlı Cam, ürünün eni 8mm, boyu 6mmdir.  Operasyon Listesi modülünde; fabrikada ürünler üzerine uygulanan operasyonların isimleri ve numaraları yazıyor. Örnek verecek olursak; rodaj, kesim, ısıtma gibi operasyonlardır.  Duruş Listesi modülünde; bir iş emri yapılırken eğer herhangi bir şekilde durursa neden durduğunun yazdığı bir listedir. Duruş listesi kısmında; duruş adı, duruş numarası gibi bilgiler mevcuttur. Örnek verecek olursak; planlı bakım, arıza gibi operasyonlardır. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  MES Nedir? | **YAPILDIĞI TARİH**  **11 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 6. Yapılacak Projelerin Konusu6.1 MAS (MES) Yazılımı Projesi Çalıştığım firmada fabrikadaki üretimleri canlı olarak takip edebilmek için kullandıkları MAS sistemlerinin benzerini yapmamı istediler. Bunun için kullanılacak programlama dilini, veri tabanlarını kendim seçmem gerektiği söylendi. Ben, MAS programı için ilk başta Java düşündüm. O da Nesneye Dayalı bir programlama dili olduğu için, kullanılması daha performanslı olurdu. Ama onun yerine C# kullanmayı tercih ettim. C# kullanmamın sebebi ise ara yüzünün çok rahat olmasından kaynaklanıyordu. 6.2 Barkod Stok Projesi İkinci projemin çıkış fikri ise, buradaki bir Elektrik Mühendisinin depoda yaşadığı takip eksikliğini anlatması üzerine ortaya çıktı. Depodaki malzemeleri aldıkları zaman bunları sisteme kaydedecek bir programlarının olmaması üzerineydi. Buda malzeme eksikliğine, eksik takip gibi sorunlara yol açıyordu. Ben bu eksikliğin bir Barkod Stok Programı ile yapılabileceğini söyledim.  Öncelikle basit yol şu şekildeydi; Malzemelerin isimlerini ilk olarak Code-128 tipinde barkod olarak kaydediyoruz. Bunun sebebi ise barkod isimlerinin çok uzun olması ve daha rahat okunabilirliğini sağlamasından kaynaklıdır. 6.3 Görüntü İşleme ile Camlarda Hata Tespiti Bu sadece teoride bir projeydi. Çalıştığım işletmede camlardaki hataları insan yoluyla buluyorlardı ve hata çıktığı zaman bile bazen gözlerinden kaçırıyorlardı. En son kalite testinde ortaya çıkıyordu. Ben ise kamera ve sensörler kullanarak camlardaki kırık, çizik ve belirgin çatlakları görüntü işleme yöntemiyle bulabileceğimizi söyledim. Bunun işletme için çok ileri düzey bir program olacağını söylediler. Hataları tespit etmek için gelen camın görüntüsü alıyoruz, ardından görüntüyü siyah-beyaz yapıp medyan filtresi uyguluyoruz. Medyan filtresinden sonra ortada sadece kırık, çizik kısımlar kalmış oluyor. Bu kalan hatalı yerlerin rengi, normal camın renginden farklı olduğu için birtakım hesaplamalar yapmaya başlıyoruz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  YAPILACAK PROJENİN KONUSU | **YAPILDIĞI TARİH**  **12 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7. Veri Tabanı Tasarımı İlk olarak veri tabanı tasarımı yaparken kullanıcılarımıza ve olaylarımıza bakmamız gerekiyor. Bunu da çıkarttığımız diyagramların yardımı sayesinde yapacağız. Diyagramlarda gördüğümüz üzere Admin ve Operatör isimli şahısların program üzerindeki etkilerini gördük. Aynı zamanda veri tabanında bulunması gereken tablolar sırasıyla; makineler, ürünler, alıcılar, operatörler, duruşlar, siparişler tablosudur. Bunun haricinde bazı önemli ilişkiler için ekstradan tablolar oluşturacağımız için, en başta tüm tabloları oluşturmaya gerek yok diye düşündüm. 7.1 Admin Tablosu Öncelikle Admin tablosu hiçbir şeye bağlı olmadığı gibi, sadece programın kullanımına etki edecektir. Herhangi bir özel tabloya bağlı olmayacaktır. O yüzden ilk Admin tablomuzu oluşturduk.    Admin tablomuzda başlıca sütun isimleri mevcuttur. Bunlar Kullanıcı Adı, Şifre, Ad ve Soyad kısımlarıdır. Ek olarak ise veri tabanlarında şuan en çok kullanılan ‘Primary Key’ ekidir. Birincil Anahtar anlamına gelen bu kilit işareti, o tablonun ana dizinin hangi sütuna ait olduğunu gösterir. Aynı zamanda tablodaki verilere ulaşmak için öz nitelikli, kesin ve değişmez bir değişkene ihtiyacımız vardır. O ise genellikle Key\_ID olarak geçen AdminID isimli sütundur. Bu yüzden Primary Key verilmiştir. Ek olarak Primary Key’ler başka tablolara erişim kurmak için Foreign Key olarak da kullanılabilirler. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **16 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.2 Admin Sistem Raporu Tablosu Veri tabanı tasarımının başında anlattığımız gibi, admin tablosu hiçbir önemli tabloya bağlı olmayacaktı. Ama Admin Tablomuzu kontrol edebilmek için ek olarak Admin Sistem Raporu Tablosu ekliyoruz. Bunun sebebi ise Admin Tablosunda bir işlem yapıldığı zaman bunu kaydedecek bir ortam hazırlamak. Buda güvenlik açısından çok önemlidir. Buda Admin isimli kullanıcının ne zaman girdiğini görmemiz için gereklidir.    Admin tablosunun oluşturulmasında bahsettiğimiz Foreign Key burada kullanıldı. Bu ise birincil anahtar değildir. Türkçe karşılığı yabancı anahtardır. Veri tabanlarında iki tabloyu birbirine bağlamak ve bunları uygulamak için kullanılan sütun veya sütunların birleşimidir. SQL ile ilgili yazılı kaynaklarda referans anahtar olarak da isimlendirilir. Bazı veri tabanlarında tüm verileri tek bir tabloya koyarak performansı yavaşlatmak yerine, normalizasyon yaparak tablolara bölmemiz gerekir. İşte bu noktada bölme işlemleri Foreign Key’ler ile yapılır. Burada birincil anahtarımızı, yabancı anahtarımıza bağlayan sütun ise AdminID’dir. Basit olarak şöyle ifade edebiliriz. SistemID’si 1 olan, AdminID’si 2 olan kişi sisteme 29.08.2020 tarihinde giriş yaptı. Buradaki AdminID=2 ise; Admin tablomuzda, o kişiye ait ID numarasıdır. İşte bu yüzden Foreign Key, Primary Key ve ID isimli sütunlar olmadan bu işlemleri yapamayız. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **16 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.3 Duruş Tablosu Veri tabanında, tablolar çoğaldıkça yapılan işlemler karmaşıklaşmaya doğru ilerler. Bu yüzden önceki konuda bahsettiğimiz normalizasyon olmazsa veri tabanının performansı sağlanamaz, veri tabanı şişer ve sistemin çalışmasını sıkıntıya sokar. Bir programda en önemli etken veri tabanının kalitesidir. Tüm işlemlerin hızları veri tabanına bağlıdır. Bu yüzden büyük ölçekli veri tabanımızı oluştururken en az ilişkisi olan tablodan, en çok ilişkisi olana doğru oluşturmalıyız ki tabloda karmaşıklık yaşanmasın. Bu yüzden Duruş İsimli tablomuzu oluşturuyoruz. Bu tabloda ise makineler durdukları zaman, neden durduklarını yazar. Örnek verecek olursak; planlı bakım    Gördüğümüz duruş tablosunda, Duruş ID isimli bir birincil anahtarımız, Duruş Ad ve Duruş Açıklama isminde nvarchar() değişken tipini kullanan toplamda üç adet sütunumuz var. Küçük bir örnek verecek olursak; 3 Numaralı Planlı Bakım şeklinde olabilir.  Önceki kısımlarda bahsetmediğimiz Allow Nulls isimli kutucuklardan bahsedelim. Allow Nulls Türkçe karşılığı ‘Boşa İzin Ver’ demektir. Bu ise veri tabanında bir tabloya değer girildiğinde boş olup, olmayacağına karar verir. Öncelikle en önemli kurallardan birisi, birincil anahtar asla boş olamaz. Çünkü tüm tablolar birbirleri ile Birincil Anahtar sayesinde iletişim kurar. Buradaki veri tiplerimiz bilgisayar mühendisliği derslerinde öğrendiklerimizden biraz farklı olarak karşımıza çıkıyor. Bunun sebebi ise herhangi bir yazı türü programlama dillerinde çoğunlukla String olarak ifade edilse de, veri tabanlarında nvarchar veya varchar olarak gösterilir. NVARCHAR’ı VARCHAR’dan ayıran en önemli nokta Unicode karakterlerini desteklemesidir. Unicode ise Universal Code dediğimiz Evrensel Kod anlamına gelmektedir. Bu ASCII kodlamasından daha gelişmiş bir kodlama türüdür. Aynı zamanda Geleneksel Çince, Kiril Alfabesi ve Arap Alfabesini destekleyebilir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **17 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.4 Duruş Rapor Tablosu Bu tabloda ise, makinelerin neden durduğunun raporunu tutacağız. Duruşların ne zaman başladığını ne zaman bittiğini görecek ve durduran kişinin kim olduğunu öğrenebileceğiz.    Duruş Rapor Tablosunda; RaporID isimli bir sayısal değeri tutan birincil anahtarımız, DuruşID ve OperatorID isimli sütunlarda bulunan yabancı anahtarlarımız, duruşların başlangıç ve bitiş tarihlerini içeren sütunlarımız vardır. Burada iki tablo birbirleri ile yabancı anahtarlar sayesinde haberleşebiliyorlar. Ek olarak henüz geçmediğimiz Operatör isimli tabloya da bağlıdır.  Burada dikkat etmemiz en önemli noktalardan birisi ise, yabancı anahtarlar hiçbir zaman boş olamazlar. Aynı zamanda başka tablolarda veri içeriyorlarsa, bulundukları tabloyu silemezsiniz. Önce o veriyi silmeniz lazım ki ardından silme işlemini yapabilesiniz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **17 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.5 Operatör Tablosu Bu tabloda ise, fabrikada çalışan operatörlerin bilgileri tutulacak. Fabrikanın onlara kendi verdiği okul numarası gibi olan sicil numaraları ile birlikte; adı, soyadı, doğum tarihi, Türkiye Cumhuriyeti kimlik numarası, işe giriş ve çıkış tarihleri, aktif veya pasif çalışan olduklarının bilgileri tutulacaktır.    Buradaki DurumAP isimli sütun bir boolean değişkenidir. Bit denen değişken sadece 0 ve 1’den oluşur. Buda 1 doğru (True), 0 yanlış (False) değişkenini taşır. İşe giriş ve çıkış tarihleri DateTime olarak kayıtlıdır. Yani hangi tarih ve saatte işe girdikleri veya çıktıkları tutulur. Sicil Numarası, firmanın kendi verdiği kişiye ait özel numaradır. Tüm işlemler bunlar ile yapılır, sisteme giriş bu şekilde gerçekleşir. Doğum Tarihide DateTime olarak kaydedilmiştir. Ad ve soyadı maksimum 30 karakterden oluşan NVARCHAR olarak kaydedilmiştir. OperatorID isimli sütun, int değişkenini kullanan birincil anahtardır. Buradaki en önemli kısım ise Türkiye Cumhuriyeti Kimlik Numarası kısmıdır. Bunu bir sayı olarak tutamayız. Çünkü 11 haneli bir sayının kaplayacağı yer çok fazla olacaktır ve performans açısından veri tabanını yavaşlatacaktır. Aynı zamanda bu numarayı int isimli değişkende de tutamayız. Çünkü int değişkeninin maksimum alabileceği 2.147.483.648 değeri 10 haneli bir sayıdır. Biz istersek bu numarayı bigint şeklinde tutabiliriz. (C# dilinde bigint’in karşılığı long’tur. 263-1 değere kadar alabilir.)  NVARCHAR’da uzunluk fark etmese de, VARCHAR kullansaydık, 11 haneli yapmadığımız takdirde ne kadar belirlediysek o kadar yer tutacak ve yine performans sorununa yol açacaktı. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **18 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.6 Operasyonlar Tablosu Bu tabloda operasyon isimleri ve id değerleri tutuluyor. Operasyonlar, fabrikada malzemeler üzerinde yapılan işlemleri gösterir. Örneğin Rodaj, Kesim, Lazer Kesim, Yıkama, Islatma gibi.    OperasyonID isimli sütunumuzun veri tipi sayısal bir tip olan int, OperasyonADI isimli sütun ise nvarchar(MAX) dır. Buradaki (MAX) isimli kısım, nvarchar değişkeninin ne kadar boyutunun olabileceğini söylüyor. NVARCHAR’ın alabileceği maksimum değişken, karakter başına 2 bit kullanımından dolayı toplamda 1 milyar karakter saklayabilir. Buda yaklaşık 2GB boyuta tekabül eder. Eğlenceli bir bilgi verecek olursak Tolstoy’un Savaş ve Barış kitabının 166 kopyasını NVARCHAR(MAX) isimli veri tipinin içinde saklayabiliriz. Savaş ve Barış kitabı 1440 sayfadan, 600.000 kelimeden oluşur.  Ek olarak gördüğümüz gibi “Allow Nulls” isimli kutucuklar yine boştur. Yani buradaki tabloya bir veri girerken bunları boş bırakamayız. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **18 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.7 Makineler Tablosu   Fabrikada kullanılan makineler, buradaki tabloda tutulur.Bu tabloda fabrikadaki makinelerimizin veri tabanı numaraları, isimleri, makine kodları ve yapabileceği operasyonların numaraları tutulmaktadır.  Örnek; 1 numaralı, 492-B koduna sahip, rodaj işlem makinesi, 2 numaralı operasyonu yapabilir.  MakineAdı ve MakineKodu isimli sütunlar NVARCHAR(MAX) olarak belirlenmiştir. MakineKodu isimli sütunun int belirlenmemesinin sebebi yazılan kodların harf içermesinden dolayıdır. Örnek bir makine kodu olan 492-B, sadece sayılardan oluşmaz. Bu yüzden NVARCHAR kullanıyoruz.  Operasyonlar tablosundaki primary key olan OperasyonID isimli sütun, burada Foreign Key olarak kullanılmıştır. Makineler tablosundaki MakineID isimli sütun, plan tablosunda Foreign Key olarak kullanılmıştır. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **18 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.8 Plan Tablosu   Plan tablosu, plan hazırlama tablosudur. Verilen siparişlerin; hangi makinelerde yapılacağı, hangi alıcının alacağı, kaç tane alacağı, hangi ürünü alacağı bu tabloda görüntülenir. Plan tablomuzda 5 adet Foreign Key vardır. Bunlar farklı tablolardaki Primary Key’leri kullanırlar. Burada Foreign Key’ler hariç ek olarak kendi tablomuzun iki tane sütunu vardır. Bunlar primary key olan PlanID ve nvarchar olan IsEmriNo isimli sütunlardır.  Örnek; 3 numaralı sipariş, 4 numaralı ürünü istiyor. 5 numaralı makine tarafından üretilecek. 17 numaralı alıcı tarafından alınacak. 500 adet iş emri vardır. Bu emrin numarası 24’tür. Bu planın hepsi 2 numaralı planda saklanır. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **19 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.9 Alıcı Tablosu   Alıcı tablosunda şirketin müşterileri tutulur. Alıcıların adı AlıcıAD isimli sütunda, nvarchar değişkeni olarak tutulur. Alıcıların şirket numaraları AlıcıŞirketNo isimli sütunda, nvarchar değişkeni olarak tutulur. Aynı zamanda veri tabanında bunları ayırt edebilmemiz için bir primary key olan AlıcıID isimli sütun bulunmaktadır.  Bir sipariş verildiğinde, teslim tarihinden ürün adetine kadar her şey belirlenir. Bu yüzden alıcı ismine ihtiyacımız vardır. Çünkü ürünü kimin alacağını erkenden bilip, sistemden görmek tüm işlemleri hızlandırarak performans sağlar.  MAS Sistemleri; Süreç Performans, Takip sistemleri ile birlikte çalıştığı için canlı olarak işlem yapılabiliyor ve sipariş verilebiliyor. Bazı firmalar kendi aralarındaki siparişleri elektronik yoldan hallediyorlar. Bunun sayesinde fazlalık evraklardan kurtuluyorlar ve hız konusunda baya ileride oluyorlar, gecikme yaşamıyorlar.  Alıcı Tablosunda hiçbir sütun “Boş Bırakılamaz” olarak işaretlenmiştir. Buradaki her bilgiye ihtiyacımız vardır. Ayrıca primary key’ler asla boş bırakılamaz. Boş bırakılabilir olarak işaretlersek hata verecektir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **22 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.10 Ürün Tablosu   Ürün tablosu, fabrikanın veri tabanında bulunan en önemli tablolardan birisidir. Fabrikadaki tabloların birçoğu, bu tablo ile bağlantılıdır. Çünkü fabrikada üretilen tüm ürünler bu tabloda tutulur.  Tablomuzda birincil anahtarımız olarak ürün id isimli bir sütun mevcuttur. Onun haricinde fabrikada üretilen ürünün adı, açıklaması, kodu, eni ve boyu vardır. Ürün kodu sadece sayısal bir değerden oluşmadığı için nvarchar olarak belirlenmiştir. Ürün açıklama ve adı zaten harflerden oluştuğu için nvarchar olarak belirlenmiştir. Ürün id, birincil anahtar olduğu için zorunluluktan dolayı int belirlenmiştir.  Tablodaki en ve boy float belirlenmiştir. Float veri tipi, ondalık sayıları ifade eder. En ve boy isimli uzunlukların milimetre olarak hesaplandığını varsayarsak, burada float almamız gerekiyor. Çünkü santimetreye veya metreye çevirmek istersek elimize farklı tür sayılar geçiyor.  Ürün açıklama hariç hepsi boş bırakılamaz olarak işaretlenmiştir. Fabrikalarda bazen ürün isimleri o kadar açıklayıcı oluyor ki, açıklamaya gerek duyulmadan direk boş geçiliyor. Bu yüzden burada “Boş Bırakılamaz” kutucuğu işaretlenmemiştir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **22 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.11 Sipariş Tablosu   Sipariş tablosunda; verilen siparişin numarası, alacak kişinin numarası, ürün numarası, alacağı miktar, siparişi verdiği tarih ve teslim tarihi tutulur. Buradaki tablolarda sadece ID isimli numaralarının tutulmasının sebebi, veri tabanında normalizasyon yapılmasından dolayıdır. Böylece tablolarımız daha anlaşılabilir ve daha hızlı olur. Önceki sayfalarda bahsettiğimiz normalizasyon ile aynı anlama gelir.  İstediği tarih ve teslim tarihi isimli sütunların veri tipleri datetime olarak belirlenmiştir. Datetime kısacası tarih ve saat anlamına gelir. Veri tabanında tarihi ve saati tutmamıza işe yarayan veri tipidir. Farklı formatlarda tutulabilir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **22 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.12 Üretim Rapor Tablosu   Üretim rapor tablosunda; üretilen malzemelerin ne kadar performans ile üretildiği ne kadar erken veya geç teslim edildiği, aynı kişilerin tekrar malzeme alıp almadığını öğrenmek içindir. Kısacası belirli bir tarihin raporunu bu tabloda görebiliriz. Buda süreç performans programları için oldukça işe yarar bir tablodur.  Tablodaki tüm ID sütunları int değerlerini almıştır. KeyID birincil anahtardır. Başlama ve bitiş tarihleri datetime olarak belirlenmiştir. Sipariş, üretim ve hurda adeti sayısal bir değişken olan int olarak belirlenmiştir. Buradaki tüm tablolarda “Boş Bırakılamaz” kutucuğu işaretli değildir. Bunun sebebi ise raporu alırken kullanacağımız filtreden dolayıdır. Buradaki her bilgiyi görmek yerine sadece adet miktarlarını görmek istiyorsak öbürlerini boş bırakıp, bazı kısımları dolduracağız. Bu yüzden tabloda birincil anahtar zorunluluğu hariç hepsi boş bırakılabilir olarak işaretlenmiştir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **22 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.13 Veri Tabanı Kısım 1 | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **22 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.14 Veri Tabanı Kısım 2 | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI TABLO  TASARIMI VE  İLİŞKİLERİ | **YAPILDIĞI TARİH**  **22 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.14 Veri Tabanı Sınıf ve SQL İşlemlerinin Hazırlanması Kısım 1 Yapacağımız projenin yarısını veri tabanı ve ilişkilerini hazırlayarak bitirmiş olduk. Öncelikle neden diyagram kullanmadığımı anlatayım. Diyagram kullanmamamızın sebebi, diyagram kullanılmayacak kadar anlaşılır ve basit olmasından kaynaklanıyor. Eğer çok daha büyük bir otomasyon olsaydı; bu sefer use-case, sınıf, ir diyagramları kullanmak zorunda kalırdık. Aslında diyagram kullanmamızın iki amacı vardır. Birincisi programın taslağını yani sözde kodunu ve algoritmasını hazırlamak. İkincisi ise programı hazırladığımız müşteriye daha basit bir şekilde anlatım yapabilmek. Program için kullanacağımız programlama dili C# olacak. Bu yüzden yardımcı ara yüz olarak Visual Studio 2019 kullandık. Eski sürümlere göre 2019 versiyonunda, yapay zeka destekli IntelliCode Sense bulunuyor. Bu ise yazdığımız kodları basitleştiriyor, hatalarını otomatik düzeltiyor ve daha iyi hale getiriyor. Derin öğrenme algoritması ve yapay zeka kullanan bu canlı sistem sayesinde daha hatasız kod yazabiliyoruz. Şimdi temel olarak programımızda veri tabanı seçimi diye bir kısım olması gerekiyor. Bunun sebebi ise her yıl veri tabanları yedekleniyor ve rafa kaldırıyor. Örnek verecek olursak 2018 yılının MAS veri tabanı, 2019 yılına geçince rafa kaldırılır ve yedeklenir. 2019 yılında yeni bir veri tabanı ile kullanıma başlarız. En önemli şeylerden birisi veri tabanından asla silme işlemi yapılmaz. Başka bir yere atılır veya boolean değeri olarak pasif şekilde belirlenir. Çünkü veri tabanından bir şey silmek tüm tabloların arasındaki ilişkilere zarar verebilecek potansiyele sahiptir. İlk olarak veri tabanı bağlantı kodumuzu oluşturduk. **[7]** Bundan sonra ise veri tabanı bağlantı kodumuzda kullandığımız veri tabanının adı kısmını oluşturduk.**[34]** Burada ise bağlantı kurmadan önce veri tabanının adının ne olduğunu veri tabanları isimli tablodan buluyor ve o veri tabanına bağlantı kuruluyor. FirmaDBList isimli bir veri tabanı oluşturuyoruz. Bu fabrikanın veri tabanını temsil eder. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI SINIF VE  SQL İŞLEMLERİNİN  HAZIRLANMASI | **YAPILDIĞI TARİH**  **23 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.15 Veri Tabanı Sınıf ve SQL İşlemlerinin Hazırlanması Kısım 2 Veri tabanları arasında seçim yapmak için, aktif veri tabanlarını sistemde bulmamız gerekiyor. Bunları ise elle giriş yapıyoruz. Ama SELECT \* FROM sözcüğünü kullanırken, daha profesyonel yazılımlarda listeyi açınca SQL Server üzerinde bulunan tüm veri tabanı isimlerini çekiyorlar ve öyle erişebiliyorlar. Basit bir şekilde kullandığın tablonun hangi veri tabanına ait olduğunu öğrenmek için select db\_name isimli sözcüğü kullanıp bunu bir stored procedure’a atarsın. Ardından bunu “EXEC PROCEDURE” ile çalıştırıp erişebilirsin. Ama bu sadece mevcut tablo üzerindeki veri tabanına erişmeyi sağlar. Bu yüzden çok sağlıklı bir yöntem değildir.  Programımıza iki tane personel giriş yapabilecektir. Bunlar o fabrikada çalışan operatörler ve veri tabanı yöneticileridir. İngilizce karşılığı “Database Admin” olan veri tabanı yöneticileri, veri tabanının güvenliğini, performansını, sağlığını sürekli kontrol eden ve düzelten kişi anlamına gelmektedir. Benim çalıştığım kurumda iki tane veri tabanı yöneticisi olmak üzere toplamda on kişi veri tabanından sorumluydu. Çünkü veri tabanları sürekli doluyor ve hataları düzeltmeleri gerekiyorlar. Bu yüzden bir veri tabanını, programın kalbi olarak alabiliriz. Bu yüzden veri tabanı bağlantı kodunu yazdıktan sonra, programa giriş yapan Operatör ve Yöneticilerin seçim ekranını tasarlamakla devam ediyoruz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI SINIF VE  SQL İŞLEMLERİNİN  HAZIRLANMASI | **YAPILDIĞI TARİH**  **23 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.16 Veri Tabanı Sınıf ve SQL İşlemlerinin Hazırlanması Kısım 3 Seçim ekranımızı hazırladıktan sonra veri tabanı seçimi için aşağıya bir ComboBox koyduk. Böylece veri tabanını seçip ardından bağlantıyı kurabiliyoruz. Burada gördüğümüz gibi seçim butonları kapalıdır. Bunu da güvenlik kontrolü olarak ben eklemek istedim. Eğer giriş yapan kişi veri tabanı ile bağlantı kuramazsa butonlar asla aktif olmayacaktır. Resimde bulunan sağ alt kısımdaki kırmızı renk ise veri tabanı bağlantısının kurulmadığını söylüyor. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  VERİTABANI SINIF VE  SQL İŞLEMLERİNİN  HAZIRLANMASI | **YAPILDIĞI TARİH**  **23 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.17 Admin ve Operatör Sınıflarının Hazırlanması Kısım 1 Giriş ekranımızı hazırladıktan sonra, bu giriş ekranından admin ve operatörün kendi özel ekranına gitmelerini sağlayan kısımları tasarlıyoruz. Ben çalışanlar ve tablolar için bir dinamik kütüphane yöntemi kullandım. Bu bildiğimiz .DLL uzantılı, Dinamic Link Library olarak geçer.  DLL nedir: Microsoft tarafından windows işletim sistemi üzerinde kullanıma açılan ve çalışma sırasında bağlanmaya izin verilen kütüphane yaklaşımıdır. Bu dosyaların amacı birden fazla program tarafından kullanılan kütüphaneleri içermeleri ve her programın gerekli oldukça ilgili kütüphaneden dosyayı okumasıdır.    DLL’de oluşturmak istediğimiz sınıflar bulunur. Bu sınıflar tek bir kullanıcı isimli sınıftan miras alır. Miras aldırma bazı ortak özellikleri tek bir sınıf üzerinde kullanarak, farklı özelliklere sadece kendilerinin erişebilmesi anlamına da gelebilir.  Aynı zamanda Genel Sınıf olarak oluşturduğumuz sınıf, MAS üzerinde kullandığımız bazı makinelerin isimleri, ürün listeleri, sipariş numaraları gibi değerleri alıp kısa süreliğine elinde tutacak olan sınıftır.    Bu DLL dosyalarımızı programımıza referans olarak ekleyip kullanabiliyoruz. DLL dosyalarının kodlarına, sayfa sonundaki **[1],[2],[3],[4],[5],[6]** numaralı kısımdan bakabilirsiniz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  ADMİN VE OPERATÖR  SINIFLARININ  HAZIRLANMASI | **YAPILDIĞI TARİH**  **24 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.18 Admin ve Operatör Sınıflarının Hazırlanması Kısım 2 Admin ve Operatör sınıflarının ilk önce Seçim Ekranı isimli formda giriş buton olaylarını yapıyoruz. Ondan sonra ise bunlar adına iki tane sınıf oluşturuyoruz. Bunlar ise Admin ve Operatör kaydı yapılırken, giriş yapılırken kullanacağımız sınıflardır. Kısacası kontrol ediyorlar. Eğer böyle bir operatör kayıtlı ise, kaydı iptal et gibi özelliklere erişebiliyoruz. **[14]**  Giriş algoritması şu şekilde anlatılabilir:   1. İsim ve şifreni girdikten sonra butona tıkla 2. Böyle bir kayıt veri tabanında var mı? Yok ise güvenlik açısından “İsim ve Şifreniz yanlış” uyarısı çıksın. Giriş Ekranı açılmaz. 3. Böyle bir kayıt veri tabanında varsa ve girilen bilgiler ile eşleşiyorsa giriş yap. Giriş Ekranı açılır. 4. Böyle bir kayıt veri tabanında varsa ve girilen bilgiler ile eşleşmiyorsa “İsim ve Şifreniz yanlış” uyarısı çıkar.   Kayıt algoritması şu şekilde anlatılabilir:   1. Operatör Kaydı yapma butonuna tıkla 2. Böyle bir kimlik numarası ve sicil numarasına kayıtlı bir operatör sistemde var mı? Yok ise kaydı başarıyla yap. 3. Böyle bir kimlik numarası ve sicil numarasına kayıtlı bir operatör sistemde varsa “Zaten böyle bir kullanıcı mevcut” hatasını ver ve kaydı iptal et.   Admin silme algoritması şu şekilde anlatılabilir:  Öncelikle operatörler silinmiyor ve işten çıkış tarihleri girilerek durum isimli veri tabanındaki sütun değeri 0 olarak değiştiriliyor. Adminleri silebiliyoruz. Admin silme işlemini DB üzerinden yapıyoruz. Program üzerinden güvenlik amaçlı silmiyoruz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  ADMİN VE OPERATÖR  SINIFLARININ  HAZIRLANMASI | **YAPILDIĞI TARİH**  **24 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.19 Giriş Ekranlarının Tasarımı ve Kodlarının Yazılması Kısım 1 Öncelikle iki giriş ekranımıza admin ve operatör isimli kişilerin temsili resimlerini, kullanıcı adı ve şifre isimli kutucukları ekliyoruz.    Giriş yap butonuna tıkladığımızda çalışacak kodlara **[7.1] ve [7.2]** numaralı kısımdan ulaşabilirsiniz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  GİRİŞ EKRANLARININ  TASARIMI VE  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **25 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.20 Giriş Ekranlarının Tasarımı ve Kodlarının Yazılması Kısım 2 Giriş yap butonuna tıkladığımız esnada AdminSınıf isimli sınıfı çalıştırıyor ve kullanıcı adı, şifre kutucuklarına girilen değerleri AdminSınıf isimli sınıfa gönderiyor. Bu sayede girişi kontrol edebiliyoruz. Aynı zamanda giriş yap butonuna tıkladığımızda “Giriş Kısmını Çalıştır” isimli fonksiyonu çalıştıracağımızı söylüyoruz. Bu sayede bilgiler AdminSınıf isimli sınıfa gidince nerede çalışacaklarını biliyorlar. Bu fonksiyonun adı AdminKontrol’dür. Buranın kaynak kodları **[8.1], [8.2]** ve **[9]** numaralı kısımlardadır.    Giriş yapmak için SQL’de en çok kullanılan sözcüklerden birisi olan SELECT \* FROM cümleciğini kullanıyoruz. Bunlarla işlem oluşturmaya SQL Query’leri denir. SELECT \* FROM anlam babında veri tabanından her şeyi getir demektir. Oradaki yıldız(\*) işareti her şeyi getireceğini gösterir. Eğer sadece ad ve soyad getirmek isteseydik. SELECT ad,soyad \* FROM yazmamız gerekirdi. FROM’dan sonra gelen kısım ise hangi tablodan çekileceğini gösterir. Örnek verecek olursak; SELECT ad,soyad FROM operatörler yazarsak, operatörler isimli tablodan ad ve soyad isimli sütunları getir anlamına gelmiş olacaktı. Kaynak kodlarının hepsi en son sayfada olacağı için burada sadece bir kısmını gösterebilirim. Bu aşağıda gösterilen sql cümleciği admin isimli tablodan tüm bilgileri, kullanıcı adı ve şifreye göre çekiyor. Yani kullanıcı adı veya şifre eşleşmezse bu cümle boş bir sonuç döndürür. WHERE deyimi koşul anlamına gelmektedir. SELECT \* FROM operatör WHERE operatorid=’5’ cümlesi operatör id’si 5 olan kişinin tüm bilgilerini getir anlamına gelir.    Not: Buradaki kullanım parametresiz kullanımdır. Parametreli kullanımdan ileride bahsedilecektir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  GİRİŞ EKRANLARININ  TASARIMI VE  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **25 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.21 Ana Menü ve İçeriklerinin Hazırlanması Kısım 1   Ana menümüz tabControl ile yapıldı. Tüm özellikleri bir arada içeriyor, bu sayede performans sayesinde kar etmiş olduk. Eğer tek tek başka formlara geçseydik, öbür formlar arkada açık kalabilir ve fazladan boş yere bellek israfı yapabilirdi. Ana menü resminde gördüğümüz üzere sağ altta kutucukları temizleyen, programı yenileyen, programı kapatan ve giriş ekranına dönmemizi sağlayan butonlar bulunuyor. Bu sayede programımızı daha stabil ve kullanışlı hale getirebiliyoruz. Resimde görmüş olduğumuz örnek Operasyonlar listesi isimli tabControl’dür. Burada Operasyon ekleyebilir, silebilir ve güncelleyebiliriz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  ANA MENÜ VE  İÇERİKLERİNİN  HAZIRLANMASI | **YAPILDIĞI TARİH**  **26 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.22 Ana Menü ve İçeriklerinin Hazırlanması Kısım 2   Programımızda içerik olarak eklenecek şeyleri aslında veri tabanını hazırlarken belirlemiştik. Bu sayede rahat bir şekilde ekleme yapabiliyoruz. Gördüğümüz gibi ana menüde rapor tablosu, operatör kontrol ekranı, makine listesi, alıcı listesi, ürün listesi, üretim raporu, sipariş listesi, operasyon listesi, duruş listesi, admin raporu, operatör listesi, planlama ve veri tabanları isimli kısımlar vardır. Buradaki tabControl kısımlarının bazılarını hazırlayamadım. Çünkü staj süresi yeterince kısa, çalışma süresi de yetersiz olduğu için, günde 10 saat çalışma ile bu kadarını yapabildim.  Ana menü resminde gördüğümüz temizle butonu foreach(Erişene Kadar) döngüsü ile kutucuklardaki (TextBox) yazıları temizlemek için bir fonksiyonu çalıştırır. **[62]**  Yenile butonu tablolarımızı yeniler ve en güncel halini ekrana getirir. **[61]**  Çıkış butonu ana menü ekranına geri dönmemizi sağlar. Bunun için ise mevcut formu kapatarak, bir önceki formu geri açar. **[61]**  Kapat butonu tüm formları ve işlemleri bitirir. Ardından uygulamayı kapatır. Arka planda boş yere çalışmasını da engeller. **[61]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  ANA MENÜ VE  İÇERİKLERİNİN  HAZIRLANMASI | **YAPILDIĞI TARİH**  **26 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.23 Operatör Listesi ve Operasyon Listesi Kodlarının Yazılması Kısım 1   Öncelikle operatörler ekleme **[16]**, silme [**15]** ve güncelleme **[17]** olarak üç işlemde ayrılıyorlar. Aslında silme olmaması gerekiyordu. Bunun yerine Durum isimli kutucuk değiştirilecek. Resimde gördüğümüz OperatorID isimli kutunun kapalı olmasının sebebi, veri tabanında ID isimli sütun Auto-Increment olarak belirlenmiştir. Bu ise ID numaraları, birer birer artar olduğu içindir. Bu sayede veri tabanında id ler karışmamış olur. O yüzden OperatörID isimli kısım sadece silme veya güncelleme işlemi yapılırken kullanılacağı için oraya eklenmiştir. Ek olarak İşten Ayrıldı isimli kutucuğu işaretlersek işe giriş tarihi isimli tarih seçim kutusu ve durum isimli kutucuk kapanıyor. Durum isimli kutucuğa 0 değerini atıyor. Bunun sebebi ise işten ayrılması kişinin tamamen çıktığı anlamına gelir. 0 işten ayrıldı, 1 aktif çalışıyor anlamına gelir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  OPERATÖR LİSTESİ VE  OPERASYON LİSTESİ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **29 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.24 Operatör Listesi ve Operasyon Listesi Kodlarının Yazılması Kısım 2   Resimde gördüğümüz en üstte Operatör Silme **[15]** isimli TabControl’a bastığımızda OperatorID hariç tüm kutucuklar kilitleniyor ve siliniyor. Bunun sebebi ise silme işlemini ID’ye göre yapmamızdır.  Tekrardan bahsedildiği gibi veri tabanlarında silme işlemi yapmak çok efektik bir yöntem olmadığı için buradaki silme işleminin amacı, bir operatör kayıt edilirken bilgileri yanlış girilirse geri dönülebilsin amaçlıdır. Bilgileri güncelleyebilir veya silip baştan hazırlayabilirsiniz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  OPERATÖR LİSTESİ VE  OPERASYON LİSTESİ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **29 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.25 Operatör Listesi ve Operasyon Listesi Kodlarının Yazılması Kısım 3   Operator güncelleme işlemi **[17]**, operatörün bilgilerini güncellemek için yapılmıştır. Primary Key olan ID hariç her şeyi güncelleyebiliriz. Eğer tabloda varsa Foreign Key sütunlarıda güncellenebilir. Örneğin operatör işten çıktığı zaman bu güncellemeyi yapmamız gerekiyor. “İşten Ayrıldı” kısmına tıklamamız gerekiyor. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  OPERATÖR LİSTESİ VE  OPERASYON LİSTESİ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **29 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.26 Operatör Listesi ve Operasyon Listesi Kodlarının Yazılması Kısım 4   Öncelikle ana menüde, operatörler bölümünde bulunan tablodan bir operatöre tıklayıp değerlerini buraya aktarıyoruz. Bu işlem otomatik yapılıyor. Seçtiğimiz DataGridView’daki veriyi alıyor ve eşleştiği TextBox’lara aktarma işlemini yapıyor. **[73]** Böylece tek tek elle bilgileri girmemiş oluyoruz. Bilgileri güncelledikten sonra tabloyu otomatik yeniler ve yeni değerleri aktarır. **[46],[47]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  OPERATÖR LİSTESİ VE  OPERASYON LİSTESİ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **29 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.27 Operatör Listesi ve Operasyon Listesi Kodlarının Yazılması Kısım 5   Önceki sayfalarda bahsettiğimiz operatör listesinin görünümü bu şekildedir. Tasarlamasını buradaki bir değere tıkladığımızda, operatör işlemlerindeki kutucuklara otomatik aktarması şeklinde yaptım. Bu sayede tıkladığımız kaydı otomatik alıp üzerinde hızlıca işlem yapabiliyoruz.  Burada klasik operatör bilgileri bulunmaktadır. Sıralama ID değerlerine göre yapılır. Bu tabloya bilgi eklenmesi kodları **[39]** numaralı kısımda mevcuttur. Çok basit mantıkla çalışan tablo doldurma işlemi, bir SELECT \* FROM sözcüğü kullanarak gelen bilgileri id’ye göre sıralayarak aktarır. Burada bulunan tablodaki sütunların sıraları, genişlikleri ve başlıkları ayarlanabilir şekilde tasarlanmıştır. **[39]** Kaynak kodlarının sayfa sonunda bulunması gerektiğinden dolayı, SELECT cümleciğini gösterebilirim.    Şekil: DataGridView tablosunu doldurma kodunun bir kısmı  Kaynak Kodlar:   1. Operatör Silme Butonu **[45]** 2. Operatör Güncelleme Butonu **[46], [47]** 3. DataGridView Tıklanınca Kutuları Doldurma Olayı **[39]** 4. Operator Ekleme Butonu **[44]** 5. İşten Ayrıldı Kutucuğuna Tıklama Olayı **[61]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  OPERATÖR LİSTESİ VE  OPERASYON LİSTESİ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **29 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.27 Operatör Listesi ve Operasyon Listesi Kodlarının Yazılması Kısım 6 Önceki sayfalarda operatör kısmını anlattık. Şimdi ise operasyon listesini anlatacağız.    Operatör ekleme, silme, butonları ile aynı işlemleri yapan operasyon ekleme **[44]**, silme **[45]** ve güncelleme **[46],[47]** işlemleri vardır. Üstte ise yine tekrardan operatör bölümünde gördüğümüz tablo mevcuttur. Buradan seçerek aşağıya bilgileri çekebiliriz. **[39]**  Veri tabanlarında silme işlemlerini id’ye göre yaparken, eğer foreign key belirlemişsek ve silmeye çalıştığımız id başka bir tabloda kayıtlıysa silme işlemi yapılamayacaktır. O yüzden ilk olarak foreign key id’lerini silmeli ve sonra silmek istediğimiz kısmı silmeliyiz. Örneğin 3 numaralı makine 4 numaralı Operasyonu kullanıyorsa, 4 numaralı operasyonu buradaki Operasyon Listesi isimli bölümde silemeyiz. İlk önce 3 numaralı makinedeki 4 numaralı operasyon numarasını silmeli veya değiştirmeliyiz. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  OPERATÖR LİSTESİ VE  OPERASYON LİSTESİ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **29 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.28 Makinelerin Listesi ve Duruş Listesi Kodlarının Yazılması Kısım 1 Makineler sadece kendi başına kullanılamıyorlar. Makinelerin çalışması için en önemli şey hangi operasyonları yapabildiğidir. Bu yüzden Makineler Listesi isimli bölümde makineler ile birlikte operasyonlarda bulunmaktadır. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  MAKİNELERİN LİSTESİ VE  DURUŞ LİSTESİ KODLARININ  YAZILMASI | **YAPILDIĞI TARİH**  **30 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.29 Makinelerin Listesi ve Duruş Listesi Kodlarının Yazılması Kısım 2   Makine ekleme ekranında **[51],** OperasyonID isimli kutucuğun kapalı olduğu görülüyor. Bunun sebebi ise makinenin kullanacağı operasyonu sağdan seçmemiz gerekiyor. Çünkü yaptığı operasyon bir foreign keydir ve öbür tablolar ile bağlantı kurarken bunu kullanacaktır. Bu yüzden OperasyonID girmeden makine kaydetmek mümkün değildir.  Makinelerdeki güncelleme, silme, ekleme işlemleri önceki sayfalarda gördüğümüz operatör ekleme,silme,güncelleme işlemleri ile aynı şeyi yapmaktadır. Bu yüzden anlatımı yapılmamıştır.  Hiçbir farkı yoktur. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  MAKİNELERİN LİSTESİ VE  DURUŞ LİSTESİ KODLARININ  YAZILMASI | **YAPILDIĞI TARİH**  **30 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.30 Makinelerin Listesi ve Duruş Listesi Kodlarının Yazılması Kısım 3   Bu şekilde bir kayıt yapmaya çalıştığımızda ekleme işlemini yapmayacaktır. Operasyon ID’yide soldaki kutucuktan seçip buradaki kutuya aktarırsak ekleme işlemi başarıyla tamamlanacaktır.    Kaynak Kodlar:   1. Makineyi Ekleme Butonu **[51]** 2. Makineyi Silme Butonu **[52]** 3. Makineyi Güncelleme Butonu **[53]** 4. Operasyonlar Tablosuna Tıklanınca Kutucuğu Doldurma **[69]** 5. Makineler Tablosuna Tıklanınca Kutucuğu Doldurma **[68]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  MAKİNELERİN LİSTESİ VE  DURUŞ LİSTESİ KODLARININ  YAZILMASI | **YAPILDIĞI TARİH**  **30 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.31 Makinelerin Listesi ve Duruş Listesi Kodlarının Yazılması Kısım 4   Duruş ekranında yine bir primary key olan DuruşID isimli kısım kapalıdır. Bunun sebebi birer birer arttırmasından dolayıdır. O yüzden oraya bir değer girmiyoruz. Kendisi atama işlemini yapıyor. Duruş’un adı ve Duruş Açıklaması vardır. Bunlar TextBox ve RichTextBox kutucukları olarak belirlenmiştir. RichTextBox temel yapı bakımından TextBox ile aynı özelliklere sahiptir. Sadece fazladan alanı görmemizi sağlar.  Duruş ekleme **[57]**, silme **[58]**, güncelleme **[59]** butonları ve tabControl bölümleri öbür bölümlerle aynı kodlara sahiptir. Sadece bazı değerleri farklıdır. Sayfa sonundaki kaynak kodları kısmında bunları görebilirsiniz.    Kaynak Kodları:   1. Duruş Ekleme Butonu **[57]** 2. Duruş Silme Butonu **[58]** 3. Duruş Güncelleme Butonu **[59]** 4. Duruş Tablosuna Tıklanınca Kutucuğu Doldurma **[69]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  MAKİNELERİN LİSTESİ VE  DURUŞ LİSTESİ KODLARININ  YAZILMASI | **YAPILDIĞI TARİH**  **30 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.32 Alıcı, Ürün Listesi ve Sipariş Formlarının Kodlarının Yazılması Kısım 1   Ürünler ekranında temel işlemleri yapan ürün ekleme **[48]**, silme **[49]**, güncelleme **[50]** butonları vardır. Bunun tasarımının öbürlerinden farklı olmasının sebebi bunun prototip olarak geliştirilen bir kısım olduğudur. Çünkü ekstra özellik olarak Arama Butonu eklemeyi düşünüyordum. Resimde sağ altta gördüğünüz arama butonuna bir ürünün herhangi bir bilgisini yazınca sağda sadece o ürün aramada sonuç olarak gözükecekti. Ama göründüğünden de zor olan bu işlem yetersiz zamandan dolayı yapılmamış, sadece prototipi kalmıştır. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  ALICI, ÜRÜN LİSTESİ,  SİPARİŞ FORMLARININ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **31 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.33 Alıcı, Ürün Listesi ve Sipariş Formlarının Kodlarının Yazılması Kısım 2   Tabloda gördüğümüz en ve boy değerleri milimetre cinsindendir. Örnek gösterirsek ÜrünID’si 1 olan sütunun boyutu 26mm,13mm olarak görülebilir. Ürünün adı, açıklaması, kodu, en ve boy değerleri buradaki tabloda gösteriliyor.  Kaynak Kodları:   1. Ürün Ekleme Butonu **[48]** 2. Ürün Güncelleme Butonu **[50]** 3. Ürün Silme Butonu **[49]** 4. Ürünler Tablosuna Tıklanınca Kutucuğu Doldurma **[68]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  ALICI, ÜRÜN LİSTESİ,  SİPARİŞ FORMLARININ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **31 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 7.34 Alıcı, Ürün Listesi ve Sipariş Formlarının Kodlarının Yazılması Kısım 3   Sipariş ekleme ve alıcı listesinin sadece tasarımı yapılmıştır. Bu yüzden kaynak kodları yoktur. Çünkü kaynak kodları operatör listesinin kaynak kodları ile neredeyse aynıdır. MAS sistemini böylelikle bitirmiş olduk. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  ALICI, ÜRÜN LİSTESİ,  SİPARİŞ FORMLARININ  KODLARININ YAZILMASI | **YAPILDIĞI TARİH**  **31 / 07 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 8 Stok Takip Projesinin Yapısı İlk sayfalarda anlattığım gibi, fabrikadaki bir parça bölümünde parçaların sayıları ve ne zaman alındığı kontrol edilemiyor. Bu yüzden bir barkod okuyucu ile stok takip projesi yapmaya karar verdim. Bu çok kısa ve küçük program, staj süresince bunu benden araya sıkıştırıp basit bir prototip göstermemi istediler. Sadece barkodu okutma işlemi bile yeterdi. Bu yüzden anlatacağım program IronBarCode isimli bir paket kullanarak barkodu okumamıza yardımcı oluyor. 8.1 Barkod nedir? Barkod hayatımızda kullanılan birçok ürünün üzerindeki beyaz zemine siyah bir şekilde basılmış, çizgi ve sayılardan oluşan bir özel bir sistemdir. Barkod sistemleri kullanılmasıyla birlikte günlük hayatta ve iş hayatında büyük kolaylıklar sağlamıştır. En önemlisi alışverişlerin çileye dönmesini engelleyen barkod sistemi sayesinde ürünler barkod okuyucuya tutularak ürün fiyat girdisi otomatik bir şekilde ekrana yansımaktadır. Bu sayede onlarca, yüzlerce ürünü kısa sürede kasadan geçebilir ve ödemenizi yapıp alışverişlerden kurtulabilirsiniz. 8.2 Barkod Çeşitleri nedir? UPC, EAN, EAN-13, EAN-8, Code 39, Code 93, Code 128. UPC numaralama sistemi Kanada ve Amerika’da, EAN-13 numaralama sistemi ise Avrupa ve Türkiye’de kullanılmaktadır. Bizim programımızda barkod Code-128 yapısındadır. Bu sayede uzun yazılı barkodları okuyabiliyoruz.  O fabrikaya ait olan bir parça barkodu;    Bu barkodun harflere çevrilmiş hali B-XBAKIMRULMAN063’tür. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  STOK TAKİP  PROJESİNİN  YAPISI | **YAPILDIĞI TARİH**  **01 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 9. Stok Takip Projesinin Veri Tabanı   Çok basit bir veri tabanına sahip olan bu stok takip projesinde, sadece 1 tane tablo vardır. Buda parçanın kodunu, miktarını ve açıklamasını tutmaya yarar. KeyID isimli sütunumuz Primary Key’dir ve Auto-Increment olarak belirlenmiştir. Yani birer birer artmaktadır. Parça Kodu isimli sütunumuz nvarchar’dır. Bunun sebebi ise kodların sadece sayısal olmamasıdır. Kodlarda aynı zamanda harflerde bulunmaktadır. Parça miktar sayısal bir değer olacağı için int olarak belirlenmiştir. Boş bırakılabilir, çünkü henüz miktarı olmayan parçaları da buraya ekleyebiliriz. Parça açıklama kısmında açıklama kısmı çok uzun olabileceği için nvarchar değişkeninin uzunluğu maksimum olarak belirlenmiştir.  Tablomuzun ismi TBL\_ParcaKodları’dır. Buradaki veri tabanı yöneticileri, veri tabanı tablolarını oluştururken başlarına TBL eklemem gerektiğini söylediler. Bunun sebebi ise standartlara ve profesyonelliğe uygun olmasından dolayıdır. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  STOK TAKİP  PROJESİNİN  VERİ TABANI | **YAPILDIĞI TARİH**  **02 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 10. Stok Takip Projesinin Kodlarının Yazılması Kaynak kodları en sonda paylaşılacağı için, bazı kısımlardan küçük örnekler vererek ilerleyeceğim. Sadece kodların ne işe yaradığını anlatacağım. Öncelikle Ironbarcode isimli eklentiyi kullanabilmek için kütüphanesini eklememiz gerekiyor.    Using IronBarCode; yazarak bu kütüphaneyi eklemiş olduk. Bu sayede artık IronBarCode’in tüm özelliklerine erişebiliriz.    Barkodu ImageBox aracını kullanarak yüklüyoruz ve kütüphanenin bize sağladığı kodu kullanarak barkodu okutuyoruz. ReadASingleBarCode(A,B,C,D) kısmında bir takım yazılar görülüyor.  A, fotoğrafın nereden geleceğini belirtmek içindir. A kaynağı pictureBox2 isimli aracın direk resim formatının halidir. Yani pictureBox2.Image şeklindedir.  B, barkodun hangi tip barkod olduğudur. Burada barkodumuz Code-128 yapısında olduğu için BarcodeEncoding.Code128 şeklinde yazılmıştır.  C, barkodun yamukluğunun görüntü işleme ile tespitidir. Aslında bakış perspektifi olarak ta söylenebilir. Eğer gelen resim yamuk çekilmişse, yandan bakılıyorsa veya eğik ise onun kontrolünü yapmak içindir. Biz düz ve sağlam bir barkod resmi ekleyeceğimiz için Low olarak ayarlanmıştır. Gördüğümüz gibi tam yazılımı BarcodeReader.BarcodeRotationCorrection.Low şeklindedir.  D, gelen barkodun bozukluğunu düzeltme işlemidir. Eğer barkod okunamıyorsa veya barkodda ekstradan çizik, bozukluklar varsa burada düzeltme işlemi yapılabilir. Bizim barkodumuz sağlam olduğu için bunu None olarak seçtik. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  STOK TAKİP  PROJESİNİN KODLARININ  YAZILMASI | **YAPILDIĞI TARİH**  **05 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 11. Stok Takip Projesinin Test Aşamaları Öncelikle barkodun fotoğrafını bir buton aracılığı ile yüklüyoruz. Sonrasında ise barkodu okutmasını istiyoruz. Eğer barkod okuma işlemini düzgün yaparsa, okunan sonucu SQL cümleciği ile çalıştırıyor ve veri tabanından bilgiyi çekip kutucuklara yazdırıyor. 11.1 Test Birinci Aşama   Birinci test işlemimizde, Code-128 kullanılması gereken yere Code-39 bir barkod yapıştırıyoruz ve sonucu görüyoruz. Sonuçtan anlayacağımız gibi aşağıda bir takım sorunlar listelenmiştir. Barkodu okuyamadığı için null(boş) döndürdüğünden dolayı, nesne başvurusu nesnenin örneğine ayarlanamadı hatası verir. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  STOK TAKİP  PROJESİNİN TEST  AŞAMALARI | **YAPILDIĞI TARİH**  **06 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 11.2 Test İkinci Aşama İkinci aşamada ise Code-128 yapısında olup, veri tabanında bulunmayan bir barkod ekliyoruz.    Okutmaya çalıştığımız barkod Code-128 yapısına sahip bir barkoddu. Fakat veri tabanında böyle bir veri olmadığı için tekrardan hata verdi. Eğer veri tabanında bu barkodun yazıya döküldüğü değerin karşılığı olsaydı böyle bir hata vermeyecekti. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  STOK TAKİP  PROJESİNİN TEST  AŞAMALARI | **YAPILDIĞI TARİH**  **06 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 11.3 Test Üçüncü Aşama Üçüncü ve son test aşamamızda doğru barkodu okuttuğumuz takdirde, gelen değeri veri tabanında bulduğu için bilgileri kutucuklara yazdıracaktır.    Resimdeki Code-128 barkodu okuttuğumuzda başarılı sonuca ulaşıyoruz. Barkodun harflere çevrilmesi sonucunda ortaya çıkan yazı B-XBAKIMRULMAN063’tür. Bu gelen sonucu SQL cümleciği ile veri tabanında aratıyor ve sonucu bulursa sağdaki kutucuklara bilgileri yazdırıyor.    Buradaki cümlede, TBL\_ParcaKodlari isimli tablodan tüm bilgiler getirme işlemi yapılmıştır. Bu işlem parça koduna göre yapılmaktadır. Buradaki kullanım parametreli kullanımdır. SQL cümleciklerinde güvenlik açısından her zaman parametreli kullanılmalıdır. MAS sisteminde gösterdiğim SELECT cümleciklerinde WHERE OperatorAd=”+ string\_operatoradi +” şeklinde kullanılması güvenlik açıkları oluşmasına sebep olabilir ve SQL Injection yapılmasına olanak verir. Bu yüzden parametreli kullanım en güvenli yollardan birisidir. Örnek olarak parametresiz kullanımda parça koduna tırnaklar ile ‘’DROP database’’ yazarsak veri tabanını silebilir. Bu çok tehlikeli bir yöntemdir.  Küçük bir kısa bir projeyle barkod sistemini böylece tasarlamış olduk. Barkod okuyucum olmadığı donanımsal olarak yapamadım. Barkod okuyucu olsaydı, Serial Port eklentisi kullanılarak bir sonuç gelecektir. Gelen String değerini döndürerek yine barkodu okutmayı başarabilirdik. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  STOK TAKİP  PROJESİNİN TEST  AŞAMALARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| KAYNAKLAR Not: Bu sayfadan sonraki her yazı kaynaklar kısmına girer. Hepsinde başlık yazmıyordur.  Ek-A: <https://ironsoftware.com/csharp/barcode/>  Ek-B: <https://dotnetdrizzles.blogspot.com/2015/05/remove-or-delete-datagridview-selected.html>  Ek-C: <https://www.c-sharpcorner.com/UploadFile/1e050f/insert-update-and-delete-record-in-datagridview-C-Sharp/>  Ek-D: Tasarladığım MES projesinin kaynak kodları: <https://github.com/batur123>  Ek-E: <https://stackoverflow.com/questions/47526546/delete-command-c-sharp-sql-server>  Ek-F: <http://www.csharpnedir.com/articles/read/?id=34>  Ek-G: <https://www.c-sharpcorner.com/blogs/textbox-only-accepting-numbers-in-c-sharp-windows-application1>  Not: Sayfalarda kaynak olarak belirtilen köşeli parantez içindeki [1] sayılar kaynaklar kısmından bulunabilir. Üstteki Ek-A kısmı ise koda dahil olmayan ve kullandığım kaynakların linklerinin bulunduğu yerdir.  Devamı arka sayfadadır. | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 1. Admin DLL.cs **[1]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 1. Kullanıcı DLL.cs **[2]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 3-Operator DLL.cs **[3]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| 4-MAS Genel Sınıf DLL.cs Parça 1 **[4]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 5-MAS Genel Sınıf DLL.cs Parça 2 **[5]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| 6-MAS Genel Sınıf DLL.cs Parça 3 **[6]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| MAS Database.cs **[7.1]**    MAS Program.cs **[7.2]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| MAS AdminGiris.cs Parça 1 **[8.1]**    MAS AdminGiris.cs Parça 2 **[8.2]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| MAS AdminGiris.cs Parça 3 **[9]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| MAS Metotlar.cs **[10]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| MAS SQLIslemleri.cs Makine Ekleme **[11]**    MAS SQLIslemleri.cs Makine Varmı Kontrolü | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Makine Silme 1 **[12]**    MAS SQLIslemleri.cs Makine Silme 2 | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Makine Güncelleme **[13]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Operatör Varmı Kontrolü **[14]**    MAS SQLIslemleri.cs Operatör Silme **[15]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Operatör Ekleme **[16]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Operatör Güncelleme **[17]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Ürün Varmı Kontrolü **[18]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Ürün Ekleme **[19]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Ürün Silme **[20]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Ürün Güncelleme **[21]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Operasyon Varmı Kontrolü **[22]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Operasyon Ekleme **[23]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Operasyon Güncelleme **[24]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Operasyon Silme **[24]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Duruş Varmı Kontrolü **[25]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Duruş Ekleme **[26]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Duruş Silme **[27]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Duruş Güncelleme **[28]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS SQLIslemleri.cs Tüm Sınıf **[29]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Giris\_Admin.cs Giriş Yap Buton **[30]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Giris\_Admin.cs Tamamı **[31]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Giris\_SecimEkran.cs Tamamı **[32]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Giris\_SecimEkran.cs Form Load **[33]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Giris\_SecimEkran.cs Değişkenler **[34]**    MAS Giris\_SecimEkran.cs ComboBox Selected Index Changed Eventi **[35]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Giris\_SecimEkran.cs Operator Ekranı Buton **[36]**    MAS Giris\_SecimEkran.cs Admin Ekranı Buton **[37]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Tamamı **[38]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Operatörler Listesi GridView Doldurma **[39]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Makineler Listesi GridView Doldurma **[40]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Operasyonlar Listesi GridView Doldurma **[41]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Ürünler Listesi GridView Doldurma **[42]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Duruş Listesi GridView Doldurma **[43]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Operatör Ekleme Buton **[44]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Operatör Silme Buton **[45]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Operatör Güncelleme Buton Parça 1 **[46]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Operatör Güncelleme Buton Parça 2 **[47]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Ürün Ekleme Buton **[48]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Ürün Silme Buton **[49]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Ürün Güncelleme Buton **[50]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Makine Ekleme Buton **[51]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Makine Silme Buton **[52]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| MAS Admin\_AnaMenu.cs Makine Güncelleme Buton **[53]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| MAS Admin\_AnaMenu.cs Operasyon Ekleme Buton **[54]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Operasyon Silme Buton **[55]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Operasyon Güncelleme Buton **[56]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Duruş Ekleme Buton **[57]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Duruş Silme Buton **[58]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Duruş Güncelleme Buton **[59]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs Giriş Ekranına Dönme Buton **[60]**    MAS Admin\_AnaMenu.cs Programı Yenileme, Programı Kapatma Buton ve Operator Durum CheckBox Click Eventi **[61]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs TextBox’ları Temizleme Butonları **[62]**    MAS Admin\_AnaMenu.cs TextBox’ları Temizleme Fonksiyonu **[63]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs TextBox’ların KeyPress Eventleri Parça 1 **[64]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs TextBox’ların KeyPress Eventleri Parça 2 **[65]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs TextBox’ların KeyPress Eventleri Parça 3 **[66]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs TextBox’ların KeyPress Eventleri Parça 4 **[67]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs DataGridView’ların CellClick Eventleri Parça 1 **[68]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs DataGridView’ların CellClick Eventleri Parça 2 **[69]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs TabPage Click Eventleri Parça 1 **[70]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs TabPage Click Eventleri Parça 2 **[71]**    MAS Admin\_AnaMenu.cs TabPage Click Eventleri Parça 3 **[72]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| MAS Admin\_AnaMenu.cs TabPage Click Eventleri Parça 4 **[73]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| Admin\_AnaMenu.cs TabPage Click Eventleri Parça 5 **[74]** | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| SQLSERVER MASDATABASE STORED PROCEDURES Parça 1 | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |

|  |  |  |
| --- | --- | --- |
| SQLSERVER MASDATABASE STORED PROCEDURES Parça 2 | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| SQLSERVER MASDATABASE STORED PROCEDURES Parça 3 | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| SQLSERVER MASDATABASE STORED PROCEDURES Parça 4 | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |
| SQLSERVER MASDATABASE STORED PROCEDURES Parça 5 | | |
| **ÇALIŞMANIN** | | **Tasdik edenin imza ve mührü** |
| **KONUSU**  KAYNAKLAR VE  KAYNAK KODLARI | **YAPILDIĞI TARİH**  **07 / 08 / 2019** |  |