**Kuru Temizleme Yönetim Sistemi Proje Analizi**

Atilla Bora Semerci

**I. Nesne Katmanı**

Uygulamanın amacı bir kuru temizleme işletmesine yardımcı olmaktır. Uygulama ile müşteri, alınan işler ve makinelerin iş yükü kontrol edilebilir.

**1. Clothes Sınıfı**

Clothes sınıfı kuru temizleme işletmesinin aldığı işler olan kıyafetleri kayıt altında tutabilmek için oluşturulmuştur. Kıyafetler türleri (t-shirt, hoodie, mont…), renk tipleri (beyaz,siyah,renkli), id numaraları, bedenleri ve işletmeye geldiği tarih kullanılarak kaydedilir. Kıyafetlerin, temizlik işleminin hangi aşamasında olduğunu belirtmek için durum değişkeni vardır.

**2. Client Sınıfı**

Client sınıfı işletmeye temizlenmesi veya ütülenmesi için ürün veren müşterileri kontrol etmek için oluşturulmuştur. Müşterinin ismi, id numarası, işletmeye ilk geldiği tarih, tipi (bireysel,kurumsal gibi), aktiflik durumu, o anda işletmede olan veya daha önce temizlenmiş kıyafetleri bu sınıf aracılığıyla tutulur. Client sınıfı kullanılarak müşterinin kıyafetleri kontrol edilebilir. Müşterinin bilgilerine erişilebilir.

**3. Machine Sınıfı**

Machine sınıfı işletmede bulunan makine tipleri için bir parent sınıftır. CleaningMachine ve IroningMachine sınıfları çoğu özelliğini Machine sınıfından miras alır. Uygulama ile makinelerdeki iş yüklerinin de kontrol edilebilmesi için Machine sınıfında id numarası, isim gibi değişkenlerin yanında iş yüklerini yani kıyafetleri tutan ArrayList de bulunur.

**4. CleaningMachine Sınıfı**

CleaningMachine sınıfı Machine sınıfının bir child sınıfıdır. Temizleme makinelerinin özelliklerini tutabilmek için oluşturulmuştur. CleaningMachine sınıfı Machine sınıfından farklı olarak iş yükündeki kıyafetlerin renk tiplerini tutan bir değişkene sahiptir. Bu özelliği renkli ve beyaz veya beyaz ve siyah kıyafetlerin bir arada temizlenmesini engeller.

**5. IroningMachine Sınıfı**

IroningMachine sınıfı Machine sınıfının bir child sınıfıdır. Ütüleme makinelerinin özelliklerini tutabilmek için oluşturulmuştur. İş yükünü tutan bir ArrayList’e sahiptir.

**6.MachineManager**

MachineManager sistemdeki makineleri kontrol edebilmek için oluşturulmuş bir sınıftır. MachineManager sınıfı ile sisteme makine eklenebilir veya silinebilir. MachineManager nesneleri sahip oldukları makineleri belirli özelliklere göre sınıflandırır. Temizleme makinelerini iş yüklerindeki genel renk tipine göre sınıflandırır. Bir iş eklendiğinde MachineManager temizleme makinelerini önce işin renk tipine göre sınıflandırır. Uygun renk tipindeki makineler belirlendikten sonra aralarından en az iş yüküne sahip olanı seçilir ve iş yüklenir. Ütüleme makineleri için renk ayrımı gerekmediğinden iş direkt olarak en az işe sahip olan makineye atılır.

**7. ClientManager**

ClientManager sistemdeki müşterileri kontrol etmek için oluşturulmuş bir sınıftır. ClientManager’ın asıl görevi işletmede kıyafete sahip olan müşteriler ile eski müşterileri birbirinden ayırmaktır. Müşterilerin kıyafetlerinin durum bilgilerini tarayarak o an işletmede kıyafete sahip olmayan müşterilerin aktiflik bilgisini pasif olarak değiştirir.

**8. MainManager**

MainManager sistemin diğer fonksiyonlarını soyutlamak için oluşturulmuş bir sınıftır. Constructor’ı çağırıldığında ClientManager ve MachineManager sınıfından nesneler oluşturur ve bu nesneler kullanılarak sistemin diğer özellikleri kullanılabilir. Polymorphism sayesinde istenilen tipte makinenin basit özelliklerini ekrana yazdıran bir fonksiyona da sahiptir.

**II. Veri Katmanı**

**1. DataManager Sınıfı**

DataManager sınıfı veritabanları ile bağlantı kurmak; veri eklemek, silmek veya güncellemek amacıyla oluşturulmuş bir sınıftır. İstenilen tablolara erişimi ve nesne katmanı ile veri alışverişini sağlayan fonksiyonları bünyesinde barındırır. MySQL 8.0 veritabanına MySQL Connector Java 8.0 kullanılarak bağlanıldı.

**2. ClientData, ClothesData, PaymentData ve EmployeeData Sınıfları**

Bu sınıflar veritabanından elde edilen verilerin UI tarafından erişilebilmesi için kullanılan basit sınıflardır. Tablolardaki verilerin istenilen şekilde tutulması ve aktarılmasından sorumludur.

**3. GenerateFromData**

GenerateFromData sınıfı veritabanından elde edilen veriler ile Client ve Clothes sınıfı nesneleri oluşturup ClientManager nesnesine eklenmesi amacıyla oluşturulmuştur. Oluşturulan Client nesnelerine sahip oldukları Clothes nesneleri eklenir ve ClientManager nesnesi sayesinde fiyat hesaplama fonksiyonu çalışır.

**III. UI Katmanı**

**1. LoginScreen Interface ve Controller’ı**

LoginScreen interface’i uygulamaya giriş için kullanıcı adı ve şifre isteyen ekrandır. Doğru kullanıcı adı ve şifre girişi ile kullanıcı ana ekrana yönlendirilir. Yanlış girişlerde ise kullanıcı uyarılır.

**2. MainLayout Interface ve Controller’ı**

Uygulamanın ana ekranını oluşturur. Bu interface kullanıcının istediği ekranları ana ekrana yükler. Ana ekrana, Clients ekranına, Payment History, Employees ekranına ve geriye gitmeyi sağlayan menüye sahiptir.

**3. MainPage Interface ve Controller’ı**

İlk girişte MainLayout’ın yüklediği arayüzdür. Bu arayüzde yeni müşteri ve kıyafet eklemeyi, ödeme almayı sağlayan pencereleri oluşturan butonlar bulunur. Sistemdeki kıyafetleri gösteren bir tablo nesnesi içerir.

**4. AddClient Interface ve Controller’ı**

Kullanıcıdan yeni müşteriden isim, telefon numarası ve müşteri tipi için bilgiler alan arayüzdür. Add butonuna basılması ile gerekli bilgiler alınır ve DataManager nesnesi aracılığıyla veritabanının clients tablosunda yeni bir kayıt oluşturulur.

**5. AddClothes Interface ve Controller’ı**

İstenilen kullanıcıya yeni kıyafet ekleyen arayüzdür. Kıyafet kaydı oluşturmak için gerekli bilgiler ChoiceBox nesneleri ile alınır ve DataManager nesnesi aracılığıyla veritabanının clothes tablosunda yeni bir kayıt oluşturulur.

**6. Payment Interface ve Controller**

Sistemdeki hesapları yükleyen, ödenmesi gereken miktarları hesaplayan, ödeme yapmayı sağlayan arayüzdür. DataManager ve GenerateFromData nesneleri aracılığıyla veritabanından Client ve Clothes nesneleri oluşturulup ClientManager nesnesine eklenir ve müşterilerin kıyafetleri ve kıyafetlerinin tiplerine göre fiyat hesaplaması yapılıp ekrana yansıtılır. Close the Bill butonuna basıldığında tablodan seçilen müşterinin ücreti sıfırlanır.

**7. ClientsPage Interface ve Controller**

Sistemdeki kullanıcıları ve kullanıcıların sahip oldukları kıyafetleri göstermek için oluşturulmuş arayüzdür. Tablodan seçilen kullanıcının bilgilerini ve kıyafetlerini TabPane içerisine gösterir. Kullanıcı ve kıyafet bilgileri güncellenebilir. Kıyafet seçilerek silinebilir. Yeni müşteri eklenebilir. Verilerin ekrana yansıtılabilmesi için ClientData ve ClothesData nesneleri kullanılır.

**8. PaymentHistory Interface ve Controller**

Sistemden yapılan ödeme bilgilerini DataManager ve PaymentData nesneleri aracılığıyla ekrana yansıtan arayüzdür.

**9. Employees Interface ve Controller**

Sistemden işletmedeki çalışanların görüntülenebilmesi ve yeni çalışan eklenip silinebilmesi için oluşturulmuş arayüzdür. ChoiceBox ve TextField nesneleri ile sisteme yeni çalışan eklemek için gerekli bilgiler alınır ve EmployeeData ve DataManager nesneleri kullanılarak veritabanındaki employees tablosuna yeni kayıt eklenir. ChoiceBox ve List butonu kullanılarak istenilen pozisyondaki çalışanlar listelenir. Seçilen çalışan sistemden silinebilir.