

Proje Raporu - Banka Sistemi

Uygulama dizini: BankSystem\bin\Release\BankSystem.exe

Proje raporu dizini: proje_raporu.pdf

Proje raporu linki: https://github.com/mertkan-k/BankaSistemiDemo/blob/master/proje_raporu.pdf

Proje video linki: <https://youtu.be/KnP-qOekXGw>

Kullanılan tasarim desenleri: Singleton Pattern ve Factory Pattern (rapor sonunda açıklamaları bulunuyor)

Genel olarak kullanıcı ve yöneticilerin giriş yapıp birtakım işlemler gerçekleştirebildikleri basit ve anlaşılır arayüzlü bir uygulama.

Veri deposu olarak iki adet json dosyası üzerinden işlemler yapılıyor, bu dosyaları bu program dışında bir araçla editlemenizi tavsiye etmiyoruz, zira dosyalarda bozulma olması halinde sistem hata verip kapanacaktır.

Eğer bu gibi bir sorun yaşarsanız dosyaları silip sistemin default olarak bu dosyaları tekrar oluşturmasını sağlayabilirsiniz.

Sistem, dosyaları kendisi oluşturduğunda ve ya yönetici hesabını bulamadığında otomatik olarak 'root' isminde ve '12345' şifresiyle bir yönetici hesabı oluşturacaktır.

Uygulamaya giriş yaptığınızda sizden kullanıcı ve şifre istemektedir, henüz bir hesabını yoksa istediğiniz bilgileri girerek 'Kayıt ol' butonuna tıklayarak kayıt olabilir ve ardından 'Giriş Yap' butonuna tıklayarak giriş yapabilirsiniz.

Kayıt olurken sistem otomatik olarak sitemdeki en büyük kullanıcı numarasına göre size bir kullanıcı numarası atamaktadır.

Eğer halihazırda kullanılan bir kullanıcı adı girerseniz hata alırsınız.

Hesabınıza her giriş yaptığınızda(doğrudan giriş, yönetim panelinden değil) girişiniz kayıt altına alınır.

Yeni bir kullanıcı oluşturulduğunda, bu kullanıcıya sistem tarafından belli bir miktar(ilgili .cs dosyasında) bakiye verilir.

Normal(Yönetici olmayan) bi hesaba girdiğinizde yapabileceğiniz işlemler:

1- Havale/Eft:

Sistemde default olarak bir banka tanımlanmış ve tüm hesaplar bu bankaya yönlendirilmiştir.

Bu sayfada öncelikle geçerli bir kullanıcı numarası girmelisiniz.

Girdiğiniz numara bakiye aktaracağınız hesabın numarasıdır ve bu numara, hesabınıza giriş yaptığınızda sol üstte '#' den sonra bölümde bulunmaktadır.

Kendi numaranızı girmeniz durumda sistem bunu algılayıp devam etmenize izin vermeyecektir.

Güvenlik amaçlı olarak, geçerli bir numara girdiğinizde bu numaraya ait kullanıcının isminin sadece ilk ve son hanesini görebilirsiniz.

Daha sonrasında işleme gönderilecek miktarı girerek devam ediyorsunuz.

Girdiğiniz miktar geçerli ise(bakiyenize uygun) son olarak hesap şifrenizi girip 'Havale Yap' butonuna tıklayarak işlemi bitirebilirsiniz.

Eğer şifreniz yanlış ise sistem bunu belirtip sizi tekrar şifre girme alanına yönlendirecektir.

Bazı durumlarda(karşı tarafın bakiyesinin ulong long'u aşması vs.) işlem başarısız olur ve bilgilendirme metninde belirtilir.

İşlem başarılı bir şekilde gerçekleştiğinde hesap bakiyeniz yenilenir ve havale işlemi her iki hesaba da kayıt alınır.

2- Ödeme Yap/Fatura Öde:

Bir kullanıcıya değil de bir şirkete ödeme yapacağınız alan.

Öncelikle işmerkezinin numarasını girmelisiniz.

Sistemde varsayılan olarak iki adet tanımlanmıştır: 1-N11 2-Hepsiburada 3-Trendyol 4-Amazon

Bu veriler ilgili .cs dosyasında kolayca değiştirilebilir.

Geçerli bir işmerkezi numarası girdiğinizde logosunun ve isminin yazdığını, ayrıca ödeme miktarının aktif hale geldiğini görebilirsiniz.

İşlemlere Havale işlemlerinde olduğu gibi devam edip işlemleri sonlandırabilirsiniz.

Başarılı bir işlem sonrasında hesabınızda kayıt oluşturulur.

3- İşlem Geçmişi:

Hesabınızda yaptığınız tüm işlemleri tarihi ve işlem sonrası bakiyesiyle birlikte görüntüleyebilirsiniz.

İsterseniz siz kontrol ederken değişiklik olma ihtimaline karşı 'Yenile' butonuna tıklayarak kayıtları sistemden tekrar çekebilirsiniz.

4- Yenile:

Siz hesabınıza giriş yaptıktan sonra dışarıdan işlemler gerçekleşmişse bazı durumlarda ekranda görüntülediğiniz veriler yenilenmemiş olabilir.

Bu durumlara karşı bu butonu kullanabilirsiniz.

5- Şifre Değiştir:

Şifrenizi değiştirebileceğiniz alana yönlendirilirsiniz.

Bu alanda öncelikle eski şifrenizi girmelisiniz.

Şifrenizi girdikten sonra yeni şifrenizi iki defa girmelisiniz.

Yeni girdiğiniz şifreler aynı değil ise sistem devam etmenize izin vermeyecektir.

Başarılı bir işlem sonrasında hesabınızda kayıt oluşturulur.

Yönetici hesabıyla giriş yaptığınızda ise sistem sizi otomatik olarak yönetici paneline yönlendirecektir.

Bu panelde yapabileceğiniz işlemler:

1- Tüm Kullanıcılar:

Yöneticiler de dahil olmak üzere tüm kullanıcıların bilgilerini açık şekilde görüntüleyebileceğiniz ve değiştirebileceğiniz alan.

Sadece istediğiniz kullanıcı seçin ve bilgilerini değiştirip 'Kaydet' butonuna tıklayın.

Değişiklikleri yapmaktan vazgeçtiyseniz 'Geri Al' butonuna tıklayarak ilgili alanların eski haline gelmesini sağlayabilirsiniz.

İsterseniz 'Hesabına Gir' butonuna tıklayarak seçtiğiniz kullanıcının paneline doğrudan giriş yapabilirsiniz. Bu durum kayıt altına alınacaktır.

2- İşlem Geçmişi:

Yöneticiler de dahil tüm kullanıcılara ait tüm kayıtları görüntüleyebileceğiniz alan.

3- Kullanıcı Girişi:

Hesabınıza yönetici olarak değil de standart bir kullanıcı gibi giriş yapın.

4- Logları Sıfırla:

Yönetici de dahil tüm hesaplara ait kayıtları temizler.

Bu işlem, hesabınızda kayıt altına alınır.

5- Kullanıcıları Sıfırla:

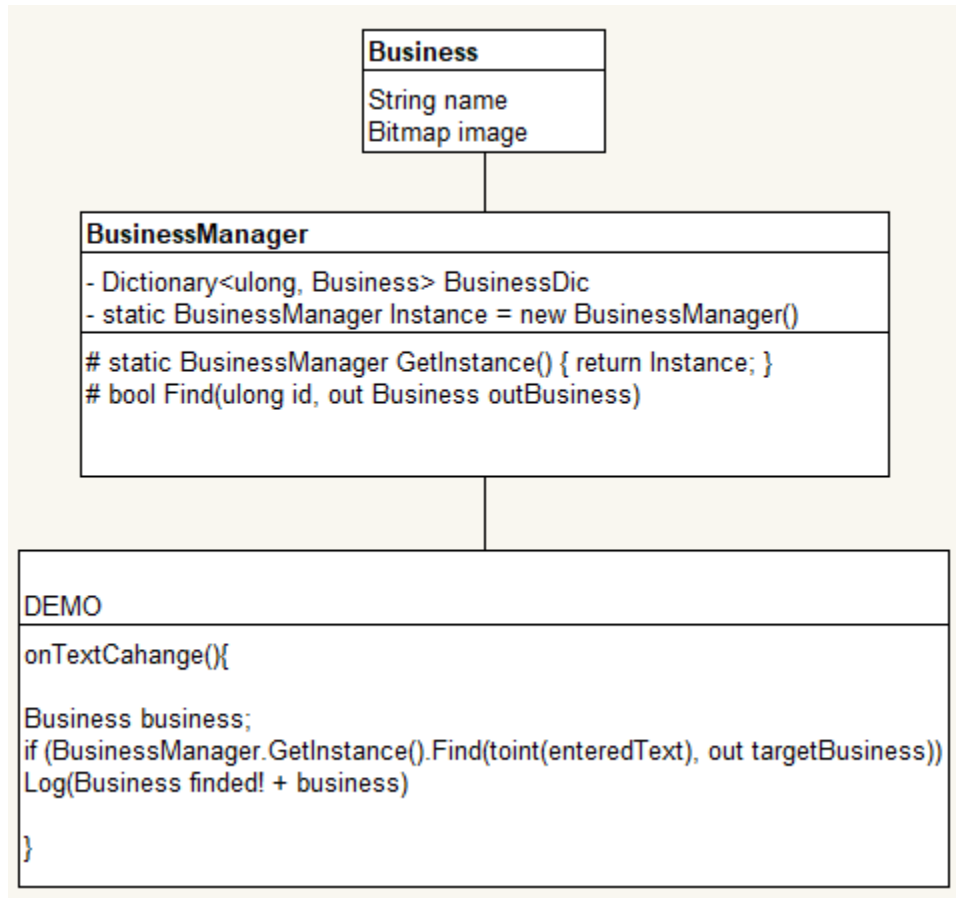
Yönetici dışındaki tüm hesapları ve bu hesaplara ait tüm logların silinmesini sağlar.

Bu işlem, hesabınızda kayıt altına alınır.

Kullanmadan önce bir kere daha düşünün.

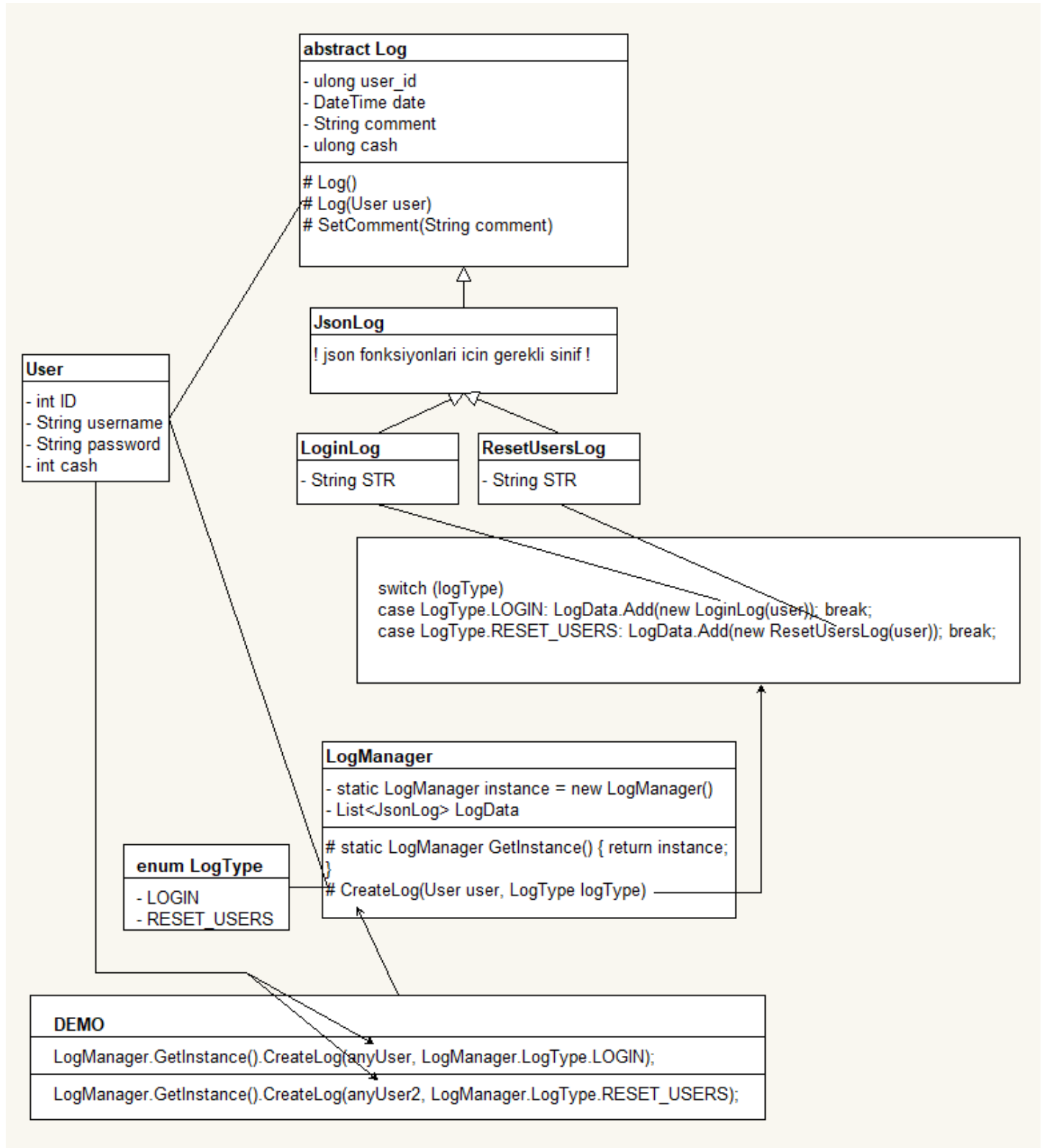
Kullanılan Tasarım Desenleri:

1- Singleton Pattern:



BusinessManager isminde bir sınıfımız var. Bu sınıfın instance isminde ve BusinessManager tipinde statik bir özelliği(attribute) bulunuyor. Bu özelliği statik yapmamızın sebebi bu sınıftan sadece bir tane oluşturulmasını istememiz. Bu özelliğe ilk olarak new keyword ünü kullanarak yeni bir BusinessManager atıyoruz. Sınıfımıza dışarıdan erişebilmek için GetInstance isminde bir methodumuz bulunuyor, bu method bize üstte bahsettiğimiz instance özelliğini return ediyor. Bu sınıfımızın bir de Find isminde bir methodu bulunuyor. Bu method sayesinde BusinessDic isminde ve map tipinde olan özelliğimiz içerisinde bize verilen key'ı(1.argüman) arıyoruz ve eğer bulursak out(c#'da fonksiyon üstündeki bir değere referans verilebilmesi için kullanılan keyword) olarak verilen targetBusiness'e bulduğumuz değeri referans verip true return ediyoruz. Eğer bulamazsak default keywordü ile referansayıp false return ediyoruz.

2- Factory Pattern:



Bu yapıda aslında hem singleton hem de factory pattern'ı birlikte kullanıyoruz. Singleton yapısı üsttekiyle aynı. Factory yapısı için de öncelikle Log isminde ve diyagramda gösterildiği üzere belli özellikleri ve methodları olan bir sınıfımız var. Daha sonra Bu sınıfımızdan kalıtım alan bir JsonLog isminde sınıfımız var. Bu sınıfın Log'dan hiçbir farkı yok, sadece json ile dosya okuma ve yazma işlemleri için abstract olmayan bir sınıfa ihtiyacımız olduğu için böyle bir sınıf kullandık. Daha sonra bu sınıftan kalıtım alan iki farklı Log sınıfımız var. Bu sınıfların (şimdilik sadece)STR isminde

static özellikleri bulunuyor. Bu özellikler, sınıf oluşturulurken Log sınıfının SetComment methoduyla Log sınıfındaki comment değerini değiştiriyor. Böylelikle Log için bir string belirlemiş oluyoruz.

Bu sınıfları oluşturmak için LogManager isminde singleton yapısı kullanan bir sınıfımız var. Bı sınıfımızda CreateLog isminde public bir method bulunuyor. Bu methodu programımızın herhangi bir yerinde log oluşturmak için kullanıyoruz. Bu methodu kullanırken öncelikle User sınıfından bir kullanıcı vermeliyiz. Daha sonra da log tipimiz için belirlenen LogType tipiniveriyoruz. Methodumuz verilen logType'e göre (switch case) uygun sınıfı oluşturup bilgileri kaydediyor.

Merkan KEÇECİ

181906086

Bilgisayar Mühendisliği