

**GAZİ ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

AHMET KÖKEN

171816038

**DERS SORUMLUSU**

doç. dr. nursal arıcı

Mayıs2018

**İÇİNDEKİLER**

1.PROJENİN AÇIKLAMASI………………………………………………….1

2. FONKSİYONLAR…………………….…………………………...………..2

# PROJENİN AÇIKLAMASI

Proje c ile yazılmıştır. Tasarlanırken iki kullanıcı tipi tanımlanmıştır: Bankacı, müşteri. Bankacının şifresi kodların içinde belirlenmiş olup değiştirilmemektedir.

Müşterilerin hesabı bankacı tarafından program üzerinden oluşturulmaktadır. Hesap numaraları yine program tarafından 20090000’dan başlayarak birer birer artmaktadır. Maksimum 100 kullanıcı oluşturulabileceği için son müşterinin hesap numarası 20090099 olunca program daha fazla hesap açmayacaktır. Müşterilerin şifresi ise yine program üzerinden rastgele olarak [1000,9999] kapalı aralığında atanmaktadır. Müşteri isterse şifresini sistem üzerinden değiştirebilmektedir.

Programda 12 tane fonksiyon tanımlanmıştır. Bunlar:

1. int login(long\*)
2. void admin(void)
3. void user(long)
4. void logout(void)
5. void havale(long)
6. void para\_yatirma(long)
7. void para\_cekme(long)
8. void fatura\_odeme(long)
9. void bakiye\_goruntule(long)
10. void hesap\_olustur(void)
11. void bloke\_kaldir(void)
12. void sifre\_degistir(long)

NOT: Proje ile ilgili ayrıntılı bilgi kodların içinde yorum satırlarında bulunmaktadır.

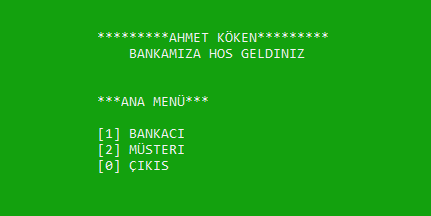
<https://ahmetkoken.com/yapisal/> adresine giderek projenin videosunu izleyebilirsiniz.

Açıklamadaki linkten de kaynak kodunu ve diğer gerekli dosyaları indirebilirsiniz.

# FONKSİYONLAR

**2.1 main Foksiyonu**

Ana fonksiyonda giriş yapma fonksiyonu çağırılmaktadır ve gelen değere göre admin ya da user fonksiyonu çağırılmaktadır. Program ilk kez çalıştırılıyor ise ekrana başlığı basmaktadır.



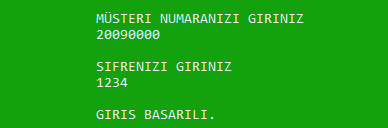
Giriş Ekranı

**2.2 login Foksiyonu**

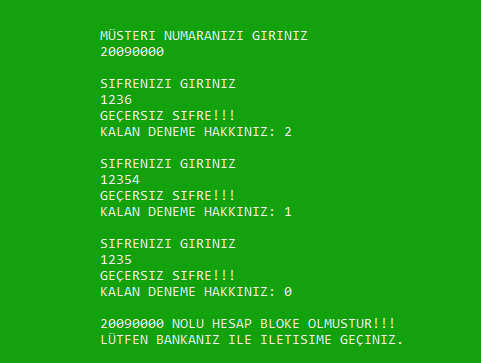
Login fonksiyonu kullanıcıya giriş yapma seçeneklerini gösteriyor. Eğer bankacı seçilirse şifre isteniyor. Girilen şifre, kodların içinden tanımlanan 7536951 şifresi ile uyuşmazsa kullanıcıya hata veriyor ve şifreyi tekrar istiyor. Admin için bloke durumu koymadım çünkü bloke olursa onu açabilecek daha yetkili birisi yok.

Eğer kullanıcı bankacı değil de müşteri seçeneğini seçerse fonksiyon önce müşteri numarasını istiyor. Müşteri numarası girildikten sonra müşterinin dosyasını okuma modunda açmaya çalışıyor. Eğer dosya açılmazsa müşteri numarası yanlış anlamına geldiği için uyarı veriyor. Dosya açılır ise dosyanın içinden müşterinin bloke durumunu (a-aktif b-bloke) ve şifresini okuyor. Sonra ise müşteriden şifresi isteniyor. Eğer müşteri şifreyi yanlış girerse uyarı veriyor. Şifre üst üste 3 kez yanlış girilirse hesabın bloke durumu b(bloke) olarak değiştiriliyor ve durum müşteriye belirtiliyor. Hesaptaki blokenin kaldırılması için bankacı hesabının aktif edilip içinden bloke kaldırma fonksiyonu çağırılmalıdır. Şifre yanlış değil de doğru girilir ise fonksiyon ana fonksiyona 2 değerini döndürüyor ve ana fonksiyon bu gelen değere göre user fonksiyonunu müşteri numarasını parametre göndererek çağırıyor.

Eğer kullanıcı müşteri seçeneğini seçip de daha önce bloke olmuş bir hesaba girmeye çalışırsa hesabın bloke olduğuna dair bir yazı ile karşılaşıyor.



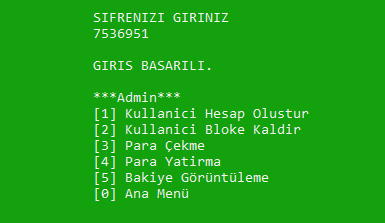
Hatasız Giriş



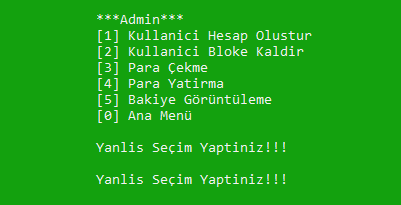
Hatalı Giriş

**2.3 admin Foksiyonu**

Admin fonksiyonu adminin yapabileceği seçenekleri ekrana yazdırıyor. Sonra sadece istenen durumda döngüden çıkacak şekilde bir while döngüsüne giriyor. Admin menüden istediği işleme ait tuşa bastığında kontrollere göre ilgili fonksiyonları çağırıyor. Ana menü seçilirse ise tekrar main fonksiyonuna döndürüyor ve ekrana bankacı ya da müşteri seçme ekranı geliyor. Eğer admin yanlış seçim yaparsa default değere girerek ekrana hata bastırıyor ve düzgün seçene kadar seçimi istiyor.



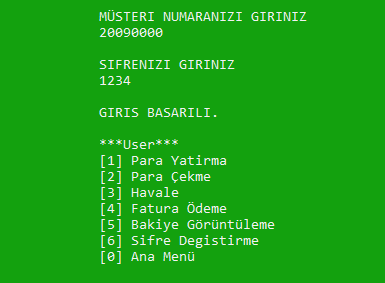
Admin Menüsü



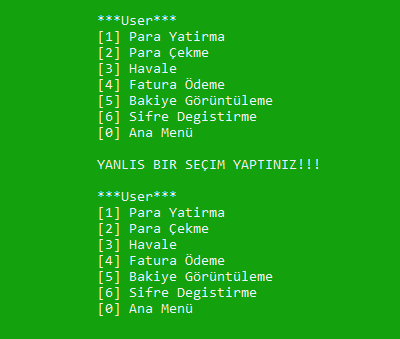
1-2-3-4-5-0 Seçilmezse

**2.4 user Foksiyonu**

User fonksiyonu da aynı admin fonksiyonu gibi kullanıcıya yapabileceği işlemleri listeliyor. Ancak admin fonksiyonundan farklı olarak kullanıcının müşteri numarasını alıyor. Müşterinin kullanıcı numarasını almaktaki sebep diğer fonksiyonları çağırırken onlara da parametre olarak göndermek. Müşteri hatalı seçim yaparsa da program müşteriye hatalı seçim yaptığı konusunda uyarıyor.



Müşteri Menüsü



Hatalı Seçim

**2.5 logout Foksiyonu**

Bu fonksiyonun tek amacı istenen zamanda çağırılarak exit(1); komutuyla programın sonlandırılmasını sağlamak.



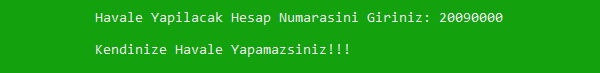
Çıkış Ekranı

**2.6 havale Foksiyonu**

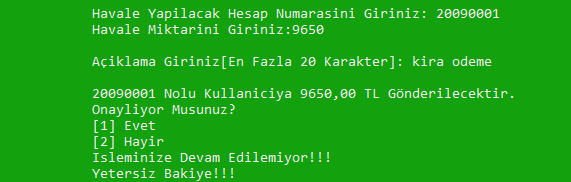
Havale fonksiyonu parametre olarak havale yapacak kişinin yani sisteme giriş yapmış kullanıcının müşteri numarasını parametre olarak alıyor.

Fonksiyon ilk olarak havale yapılacak kişinin müşteri numarasını istiyor. Girilen numara müşterinin kendi numarası ise müşteriye kendine havale yapamayacağını söylüyor. Eğer aynı değilse girilen numaraya ait dosyayı açmaya çalışıyor. Dosya açılmaz ise havale yapılacak numaranın hatalı olduğunu söylüyor. Dosya açılır ise müşteriden havale yapılacak miktarı istiyor. Havale yapılacak miktarın pozitif olması gerektiği için girilen miktarın pozitif olup olmadığı kontrol ediliyor. Pozitif değilse uyarı veriyor ve müşteri pozitif sayı girene kadar tekrar istiyor. Havale yapılacak miktar pozitif girildiğinde ise müşteriden 20 karakterlik açıklama isteniyor. Daha sonra havale yapılacak kişinin dosyası açılarak tüm bilgileri ilgili değişkenlere alıyor. Sonra para gönderecek kişinin dosyasını açıp oradaki bilgileri de ilgili değişkenlere alıyor.

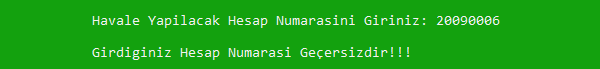
Bu işlemlerden sonra kullanıcıya havale miktarını onaylayıp onaylamadığı soruluyor. Eğer onaylamazsa dosyalardaki bilgiler değiştirilmiyor ve işlem iptal ediliyor. Onaylar ise gönderen kişinin hesabındaki bakiye ile girilen miktar kontrol ediliyor. Eğer havale miktarı bakiyeden fazla ise işleme devam edilmiyor ve yetersiz bakiye uyarısı veriliyor. Fazla değil ise gönderen kişinin bakiyesi azaltılarak, alıcının bakiyesi artırılarak dosyaya kaydediliyor ve dosyalar kapatılıyor. Transactions dosyasına da gönderen kişinin numarası alıcının numarası açıklama ve havale miktarı yazılıyor.



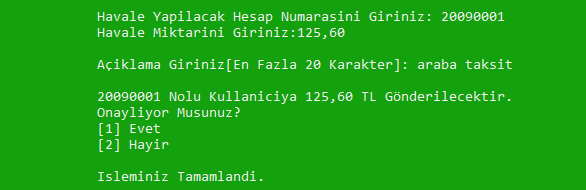
Kendine Havale



Yetersiz Bakiye



Geçersiz Müşteri Numarası



Başarılı Havale İşlemi

**2.7para\_yatirma Foksiyonu**

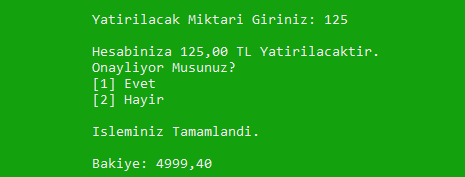
Para yatırma fonksiyonu da aynı diğer müşteri fonksiyonları gibi parametre olarak müşteri numarasını alıyor. Müşteriden yatırılacak para miktarını istiyor. Yatırılacak miktar pozitif olmadığında hesaptaki para miktarı azaltılacağı için ve bu durum istenmediğinden dolayı müşterinin para miktarını pozitif girmesi sağlanıyor.

Müşteri yatırılacak miktarı girdiğinde ise onay ekranı geliyor. Müşteri evet seçeneğini seçerse işlemler dosyaya kaydediliyor ve dosya kapatılıyor. Ekrana da müşterinin bakiyesi otomatik olarak bastırılıyor. Hayır seçeneği seçildiğinde ise işlem iptal ediliyor ve dosyaya hiçbir şey yazılmıyor.

Para yatırma işlemi transactions dosyasına da yazılmıyor. Yani para yatırma geçmişi görüntülenemiyor.



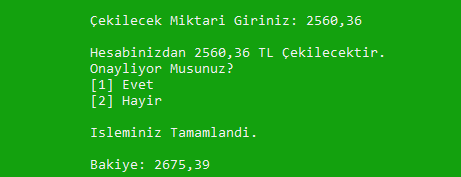
Negatif Miktar Girilirse



Başarılı Para Yatırma

**2.8para\_cekme Foksiyonu**

Para yatırma fonksiyonu ile aynı işlemi gerçekleştirmektedir. Tek farkı parayı artırma işlemi yerine parayı azaltmaktadır.

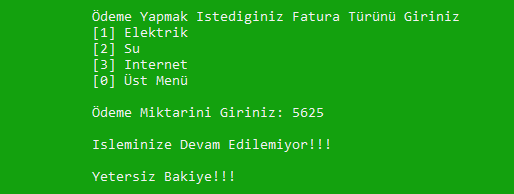


Başarılı Para Çekme

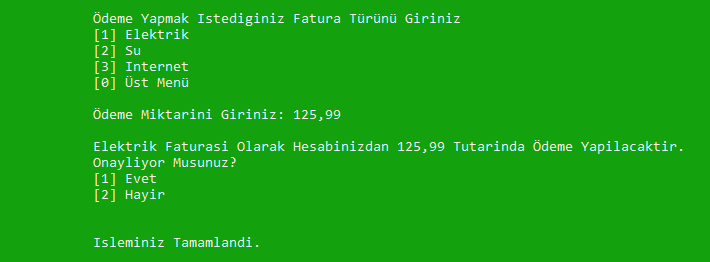
**2.9fatura\_odeme Foksiyonu**

Parametre olarak müşteri numarasını alıyor. Fonksiyon temel olarak aynı kodun üç kere tekrar etmesinden oluşuyor. Üç kez tekrar etmesinin sebebi elektrik su ve internet için ayrı ayrı yazıyı ekrana yazdırması.

Fonksiyon ilk önce fatura türünün belirlenmesini istiyor. Fatura türü belirlendikten sonra ödeme miktarı müşteriden isteniyor. Para yatırma ve para çekme fonksiyonundaki gibi ödeme miktarının pozitif olması isteniyor. Eğer pozitif değilse müşteri pozitif değer girene kadar uyarı veriyor. Müşteri pozitif değer girerse müşterinin bakiyesi dosyadan okunuyor ve ödeme miktarı ile karşılaştırılıyor. Bakiyeden fazla ödeme miktarı girildiyse işleme devam etmiyor ve yetersiz bakiye uyarısı veriyor. Bakiye yeterli ise müşteriye ödeme miktarını onaylayıp onaylamadığı soruluyor. Müşteri hayır derse işlem iptal ediliyor ve dosyaya hiçbir şey yazılmıyor. Evet derse müşterinin dosyası w modda açılıyor yani eski bilgiler silinerek açılıyor. Açtıktan sonra bakiye azaltılıyor ve dosyaya tüm bilgiler baştan yazılıyor. Daha sonra transactions dosyası açılarak faturayı ödenen kişinin müşteri numarası fatura ödeme türü ve miktarı yazılıyor ve dosya kapatılıyor.



Yetersiz Bakiye



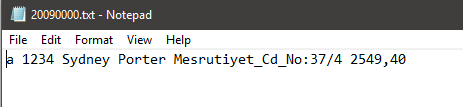
Başarılı Fatura Ödeme

**2.10bakiye\_goruntuleme Foksiyonu**

Bakiye görüntüleme fonksiyonu parametre olarak aldığı müşteri numarasına ait dosyayı açarak tüm bilgileri ilgili değişkenlere atıyor. Daha sonra ise dosyayı kapatıp bakiye değişkenini Ekrana yazdırıyor.



Bakiye Görüntüleme

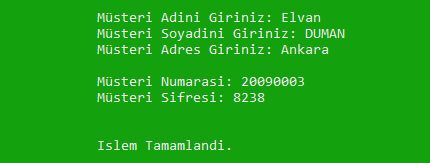


Dosyaya Kayıt Formatı

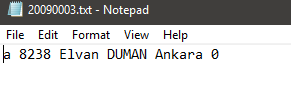
Dosyaya bilgiler "%c %d %s %s %s %f" formatında kayıt edilmektedir. Fotoğraftaki a karakteri hesabın bloke durumunu belirtmektedir. a olduğu için aktiftir, b olsaydı bloke anlamına gelecekti. 1234 int değeri ise kullanıcın şifresidir. Sydney müşterinin adı, Porter soyadı, Mesrutiyet\_Cd\_No:37/4 adresi ve 2549,40 müşterinin bakiyesidir. Bakiye değişkenine ulaşmamız için fonksiyon içinde diğer bilgileri de uygun tipteki değişkenlere atmamız gerekmektedir. Yani bakiye görüntülerken kullanıcının şifresini ihtiyacımız olmasa bile şifre değişkenine o değeri atamadan bakiye değerine ulaşamıyoruz.

**2.11 hesap\_olustur Foksiyonu**

Hesap oluşturma fonksiyonu parametre almamaktadır. Sadece admin arayüzünden erişilmektedir. Fonksiyon içinde müşteri\_no dosyası açılarak en son kullanılmış müşteri numarası alınır. Müşteri numarasının 20090099 olup olmadığı kontrol edilir. Eğer 20090099 ise 100 kullanıcı sınırına ulaşıldığı için daha fazla kullanıcı oluşturulmaz ve fonksiyon sonlanır. Eşit değil ise hesap oluşturma işlemi devam eder. musteri\_no dosyası açılarak oradaki değer bir artırılır. Daha sonra müşterinin adı, soyadı ve adresi istenir. Şifre fonksiyon içindeki kodlar sayesinde [1000-9999] aralığında otomatik olarak üretilir. Üretilen şifre ve müşterinin hesap numarası ekranda görüntülenir. Müşteri için hesap numarası ile dosya açılır, bilgileri kaydedilir ve bakiyesine ise 0 yazılır. Fonksiyon sonlanır.



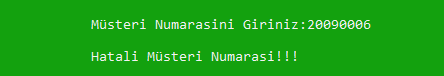
Örnek Kullanıcı Oluşturma



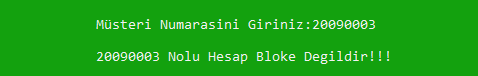
Müşteri İçin Açılan Dosya

**2.12 bloke\_kaldir Foksiyonu**

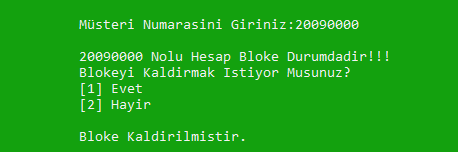
Bloke kaldırma fonksiyonu parametre almayan bir fonksiyon. Çağırıldığında adminden müşteri blokesi kaldırılacak müşteri numarasının girilmesi isteniyor. Girildikten sonra müşteri numarasına ait txt dosyası açılmaya çalışılıyor. Eğer açılmazsa öyle bir müşteri numarası yok anlamına geldiği için hata ver. Dosya açıldıysa da dosyanın bloke durumunu(a ya da b) oku. Eğer bloke durumu a ise hesabın bloke olmadığını ekrana yazdır ve fonksiyonu bitir. Bloke durumu b ise hesap bloke demektir. Admine blokeyi kaldırmak isteyip istemediği soruluyor. Evet seçilirse pointer rewind(pointer) ile başa alınıyor ve müşterinin dosyasında b harfinin üstüne a harfi yazılıyor, blokenin kaldırıldığı söyleniyor ve dosya kapatılıyor. Hayır seçilirse bloke kaldırma işlemi iptal ediliyor ve dosyada herhangi bir işlem yapılmamış oluyor.



Yanlış Müşteri Numarası Girilirse



Bloke Olmayan Hesap Girilirse

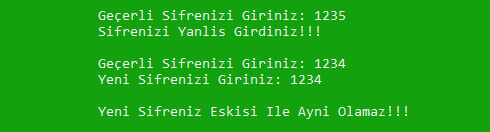


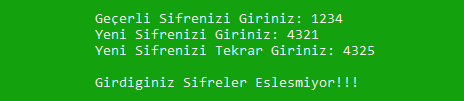
Başarılı Bloke Kaldırma

**2.12 sifre\_degistir Foksiyonu**

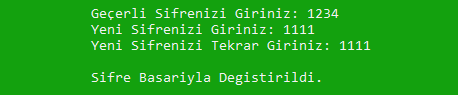
Şifre değiştirme fonksiyonu parametre olarak şifresi değişecek müşterinin müşteri numarasını almaktadır. Daha sonra müşteriye ait dosyayı okuma modunda açıyor. Tüm bilgileri ilgili değişkenlere attıktan sonra dosyayı kapatıyor.

Kullanıcıdan kullanmakta olduğu şifreyi istiyor. Şifreyi yanlış girerse tekrar istiyor. Girilen şifre ile kullanmakta olduğu şifre aynı ise yeni şifreyi istiyor. Burada önemli olan nokta yeni şifrenin eskisi ile aynı olmaması. Eski şifre ile aynı girilirse hata veriyor. Eski şifre ile yeni şifre farklı ise yeni şifreyi kontrol amaçlı tekrar istiyor. Eğer yeni şifre ile kontrol için tekrar istenen şifre farklı ise yine hata veriyor. Aynı ise Şifreyi değiştirip dosyayı w modda açıp tüm bilgileri ve yeni şifreyi dosyaya yazıyor. Dosyayı kapatıyor ve şifrenin başarıyla değiştirildiğini söylüyor.





Hatalar



Başarılı Şifre Değiştirme