

**\_\_\_\_\_\_\_\_\_\_\_\_PROGRAMLAMA VE UYGULAMALARI-II\_\_\_\_\_\_\_\_\_\_**

**ÖĞRETMEN ADI: SAİT ALP**

**PROJENİN KONUSU: MOBİL BANKA OTOMASYONU**

**PROJE EKİBİ:**

**-BERK COŞKUN TÜRE**

**-BURAK YILDIRIM**

**-İBRAHİM ENES KAYA**

**-TEVFİK OLGUNOĞLU**

**TARİH 11/05/2025**

**MOBİL BANKA OTOMASYONU**

**1.1-GİRİŞ-**

Biz ekip olarak insanların atm veya şubelerden yaptıkları işleri telefon veya bilgisayarlarından yapmasını amaçlayarak mobil banka uygulamasını geliştirdik

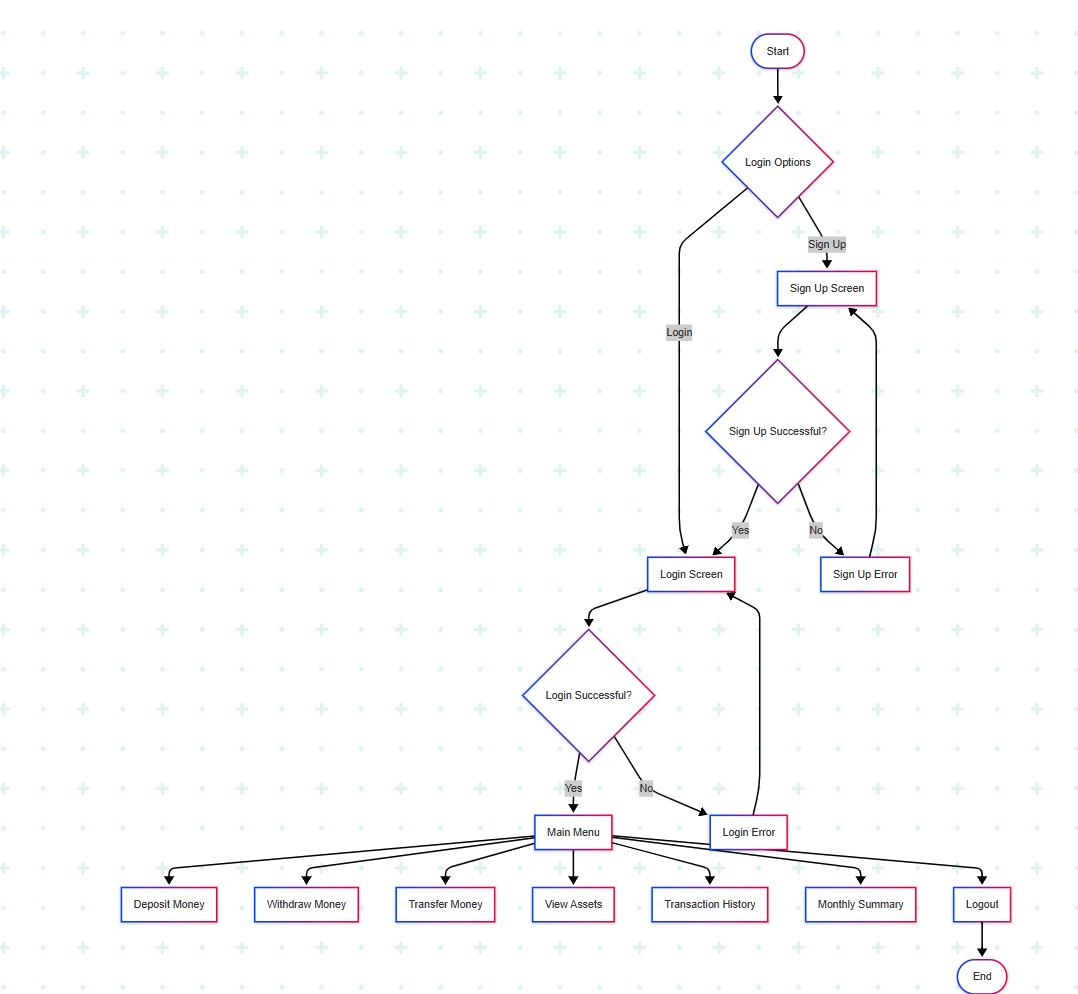
Bu uygulamayı geliştirirken iki modül oluşturduk . Bunlardan biri arayüzün arkasında çalışan fonksiyonların tutulduğu modül olan **data\_base.py** modülü diğeri ise kullanıcıların somut olarak görebildiği tıklayabildiği arayüzün sınıfsal kodlarını tutan **arayüz.py** modülüdür

**Data\_base.py** yazılan fonksiyonları isimleri ile **arayüz.py** import ederek her fonksiyon için bir arayüz sınıfı oluşturduk

Geliştirdiğimiz mobil banka uygulamasında kullanıcılar para yatırma ,para çekme, para transferi bunların yanında aylık harcamalarını görüntüleme ve bu faturaların grafiğe dökülmüş halini farklı hesaplarda bulunan paralarının yüzdelik olarak grafiğe dökülmüş halini görebilirler

**2.1--Kodun Akış Diyagramı—**

Projemizin akış diyagramını sunmak gerekirse ,



**2.2-Data\_Base.py Kullanılan Kütüphaneler ve amaçları-**

Bütün bankaların verileri kaydettikleri bir yeri olduğu gibi bizde verileri bir dosyada kayıtlı tutmak istedik Kişilerin ismi ve soyisimleri aynı olabilme ihtimalini düşünerek her kişiye random (rastgele)ve unique(benzersiz) bir hesap numarası atayarak kişileri bu hesap numaraları temsil etmektedir

Kurmuş olduğumuz bu kayıt sistemi python dictionary(sözlük) yapısına benzediği için verilerin tutulacağı dosya formatını “json” formatı olarak belirledik Bu formatta dosya yazmak için **json** kütüphanesini **import** ettik Aynı işlemleri kişilerin işlem geçmişi tutmak amaçlıda yaptık

Bu işlemleri tanımladıktan sonra,

gecmis\_dosyasi=”işlem\_gecmisi.json”

veri\_dosyasi=”data.json” dosyalarını tanımladık

Daha sonra bu şekilde dosyaların olup olmadıklarını kontrol etme amacıyla **import os** yaptık ve dosyaların kontrolünü sağladık

**2.3-Kullanıcıları Kaydetme-**

def hesap\_olustur (isim,soyisim,sifre) fonksiyonu ile kişilerin hesap oluşturması sağladık her kişiye rastgele ve benzersiz bir hesap numarası vermek için iki adımlı bir kod yazdık Önce herkese rastgele hesap numarası vermek içi **import.random** yaptık Buradan gelecek değerin türü <<class ‘int’ >> olacaktık fakat biz hesap numarası ile herhangi bir matematiksel işlem yapmayacağımız için

hesap\_numarası=str(random.randint(100000,999999)

yaptık bu şekilde belirlenen aralıkta rastgele hesap numarası üretilecekti

Üretilen numaraların unique(benzersiz) yapmak içinde yukarıda yaptığımız işlemi bir koşula bağlayarak yaptık

while hesap\_numarası in kullanicilar:

hesap\_numarası=str(random.randint(100000,999999)

yaptık

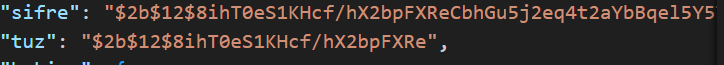
**2.4-Şifre Kontrol ve Güvenliği-**

Şifrelerin doğru şartlarda oluşmasını saplamak için **import re** yaparak şifrelerin koşullarını düzenledik Bu şekilde şifrenin içindeki karakterlere, şifrenin uzunluğuna, özel karakter olup olmadığına bakabilme imkanı bulduk Daha sonra kişilerin şifrelerini matematiksel formatta geri döndürülmesini engellemek amacıyla şifreyi **hashlemek** için **import bcrypt** yaptık

---**Hashlemek Nedir**

Hash algoritmaları (örneğin: SHA-256, bcrypt), girdiyi alır ve onu **ikili (binary)** veri olarak işler. Ancak, insanlar için okunabilir hale getirmek için bu ikili veriler genellikle:

* **16’lık sistemde (hex)** gösterilir: e3b0c44298fc1c149afbf4c8996fb924...,
* veya **base64** gibi kodlamalarla yazılır (bcrypt bu yöntemi kullanır).



### Özetle:

* Hash işlemi matematiksel olarak ikili (0 ve 1) sistemde çalışır.
* Ama çıktı, okunabilirlik için **16’lık (hex)** veya **base64** gibi formatlarla gösterilir.

Örneğin bcrypt çıktıları base64 benzeri özel bir karakter kümesiyle şifrelenir, bu yüzden farklı görünür.

**--NEDEN BCRYPT—**

### **Yavaş ve bilinçli olarak yavaşlatılabilir**

* bcrypt kasıtlı olarak yavaştır (bu bir avantaj!).
* Bu sayede brute-force (deneme-yanılma) saldırıları **çok yavaşlatılır**.
* cost (maliyet) parametresiyle işlem süresi ayarlanabilir: işlem ne kadar uzun sürerse, saldırgan için deneme yapmak o kadar zor olur.

### **Salt kullanır (otomatik)**

* Her şifre için rastgele bir **salt** (tuz) ekler.
* Aynı şifre olsa bile her seferinde **farklı bir hash** üretir.
* Bu sayede rainbow table gibi önceden hesaplanmış saldırılar **etkisiz hale gelir**.

### **Zaman içinde güncellenebilir**

* cost değeri artırılarak sistem zamanla daha güvenli hale getirilebilir (işlem gücü arttıkça saldırıya açık hale gelmesin diye).

**3.1 –VERİ ÇEKME—**

Biz parasal işlemlerde (para yatırma/çekme/transferinde) döviz üzerindeki işlemlerin güncel olması için import request yaparak güncel USD EUR fiyatlarını çektik. Bunlara ek olarak güncel altın(g) fiyatını çekmek içinde API URL kullandık

API olarak TCBM API ‘ sini kullandık ve buradan güncel olarak gram altının değerini çektik

Kullandık

def döviz\_kurları() fonksiyonunda bize değer olarak True,usd,eur,altin döndürdük eğer veri alınamazsa sabit bir değer döndürdük Bunu yapmamızdaki mantık eğer internet problemi yaşanırsa dolar,euro,altının değerlerinin son değerlerinin sabitleyerek döndürdük Bu kişilerin internet sıkıntısı dolayısıyla verileri çekemediği durumlarda bile para yatırma /çekme işlemlerini gerçekleştirmelerini sağlamak

**3.2—İlk Giriş Tarihleri—**

Bildiğiniz üzere bazı hesap türlerinde belirlenen tarihten önce para çekme işlemi yaparsanız hesabınızdaki paradan kesinti olur (ceza tutarı kesilir ) Örnek verirsek bireysel emeklilik hesabınızdan 5 yıldan önce paranızı çekerseniz bankanızın size vermiş olduğu (%35 tutarında) faiz oranı hesabınızdaki paradan hesaplanarak kesilir Bizde bireysel emeklilik hesabının ve vadeli hesabın ne zaman açıldığını ilk ne zaman bu hesaplarda işlem yapıldığını öğrenmek için **import datetime** yaparak Kişilerinin ayrı ayrı hesaplarını açtıkları zamanı kayıtlı bir şekilde tutmamızı ve kişilere kesilen faturaların zamanını göstermemize yardım eden bir kütüphanedir

Bildiğiniz üzere para çekme işleminde vadeli hesaplarda ve bireysel emeklilik hesaplarında çekilen paranın tarihine göre ceza kesme işlemi uygulanma durumu vardır

Vadeli hesaplarda kişiler hesaplarında bulunan parayı 1 yıldan daha kısa bir sürede içerisinde çekmeye çalışırlarsa vadeli hesaplarına yatırılan faiz(%25) oranından yararlanamaz ve erken çekim cezası uygulanır

Örnek olarak kişinin hesabından kesilecek miktar 100 TL ise erken çekim sonrası kesilecek miktar 125 TL olur

Aynı vadeli hesapta olduğu gibi bireysel emeklilik hesabında da erken çekim ceza durumu vardır Bireysel emeklilik hesabı vadeli hesaptan farklı olarak kişinin 5 yıldan daha kısa bir süreç içerisinde para çekerse erken çekim cezası uygulanma durumu gerçekleştirilir Kişinin toplam tutarında 5 yıl boyunca yatırılan faiz oranı (%35) kesilir

**4.1—İŞLEMLER(PARA YATIRMA/ÇEKME/TRANSFERİ)—**

Ayrı ayrı her işlem için güncel bankalardaki kurallara göre fonksiyonlar yazdık para yatırmada her hesap için faiz kurallarını ,para çekmede her hesap türü için erken para çekimi durumundaki ceza miktarlarına ,para transferinde ise aynı bankalar mı ,farklı banklara arası mı (EFT mi HAVALE mi) olduğuna göre değişen para kesmelerini göz önüne alarak ayrı ayrı fonksiyonlar yazdık

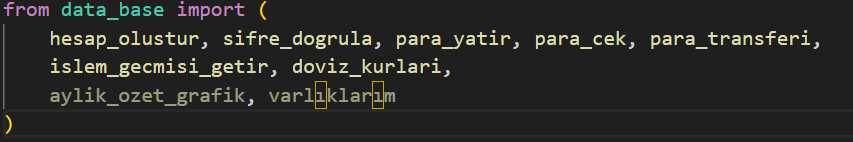
**4.2—VARLIKLARIM VE AYLIK ÖZET –**

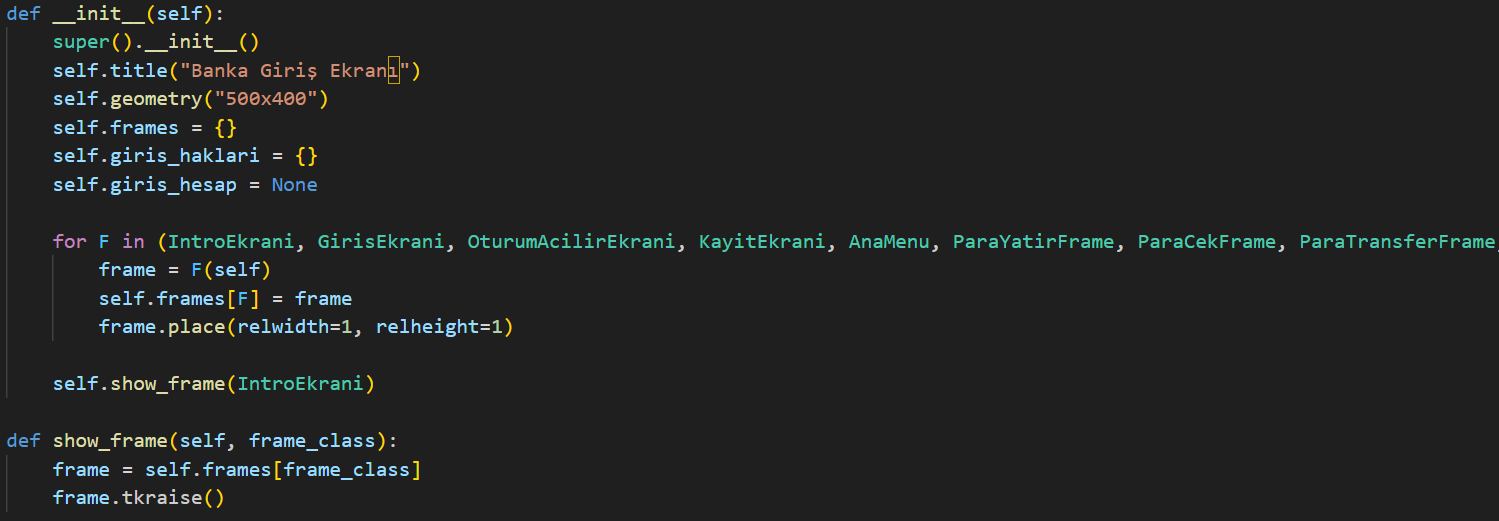
Kişilerin yatırdıkları veya çektikleri para tutarlarını işlem geçmiş adı altında bir **json** dosyasında kaydetmiştik Bu dosyadaki tutarları pasta grafiği olarak sunmak için **import matplotlib** yaptık Kişilerin işlem faturaları tutarak “Yatırılan/Çekilen” tutarlarını pasta grafiği olarak gösterilir

Varlıklarım fonksiyonunda ise kişinin hesap türlerinde bulunan paraları pasta grafiğine döktük

**5.1—ARAYÜZ –**

İlk olarak arayüz.py dosyasına data\_base.py dosyasını import ederek data\_base.py dosyasındaki fonksiyonları arayüz.py dosyasına çağırdık ve hepsi için ayrı ayrı frame(çerçeve) açtık Bunun sebebi butonlara basıldığında yeni bir pencere açılması yerine aynı pencerede farklı arayüzler olmasını istememizden dolayı bu şekilde yaptık





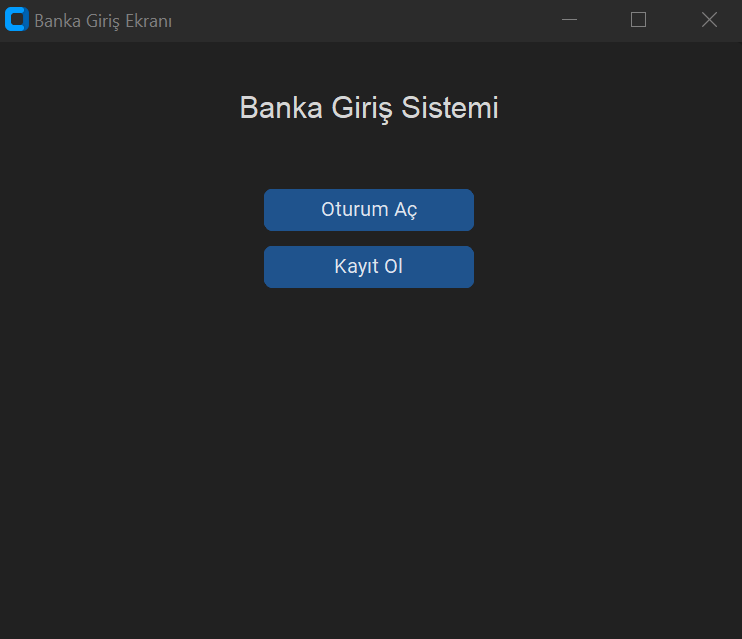
Daha sonra her bir frame(çerçeve) için ayrı ayrı sınıf oluşturduk Bunun nedeni sınıf kavramının içinde hem veriyi hem de kodu aynı yerde tutmasından dolayı kullandık

**5.2--İntro Ekranı—**

Uygulamamızı ilk açtığımızda bizi İntro Ekranı karşılıyor Bu sınıfta arka plana koyacağımız resmin uzantını yazarak resmin boyutunu ve ekrandaki x ve y koordinatlarına göreyerini belirledik Eğer bir sorun sonucu resim yüklenemezse “Resim yüklenemedi” yazıp arka plan “dark” butonun rengi ise “dark-blue” (Diğer frame(çerçeveler) ile aynı şekilde) ayarlanır

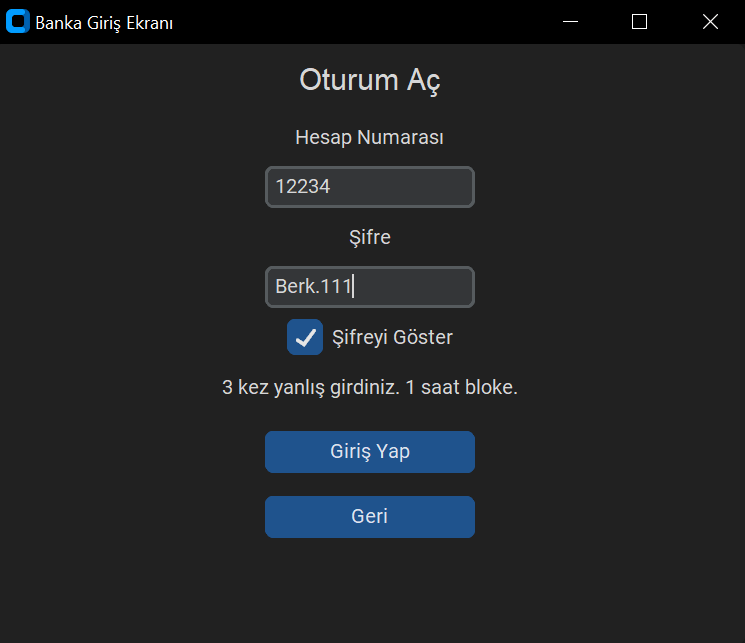
**5.3—Giriş Ekranı—**

Giriş ekranı için yazdığımız sınıfta ise bizi iki farklı buton karşılıyor

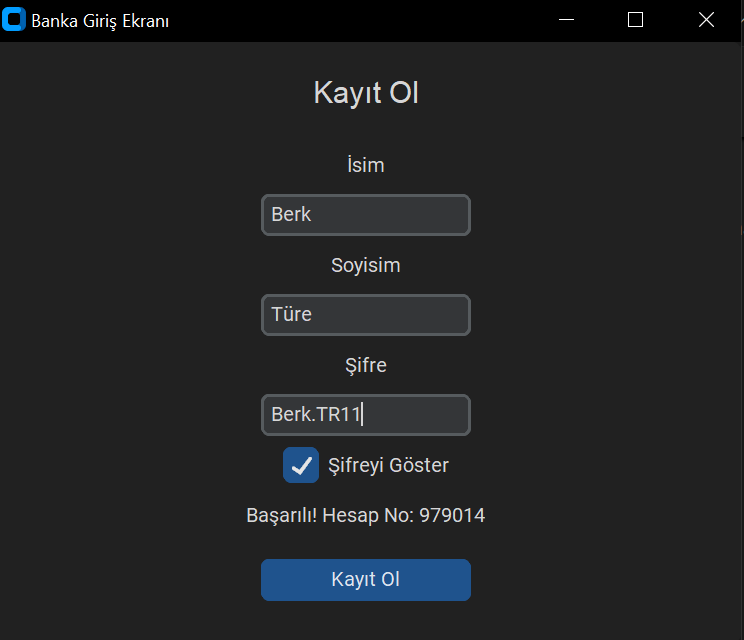
****

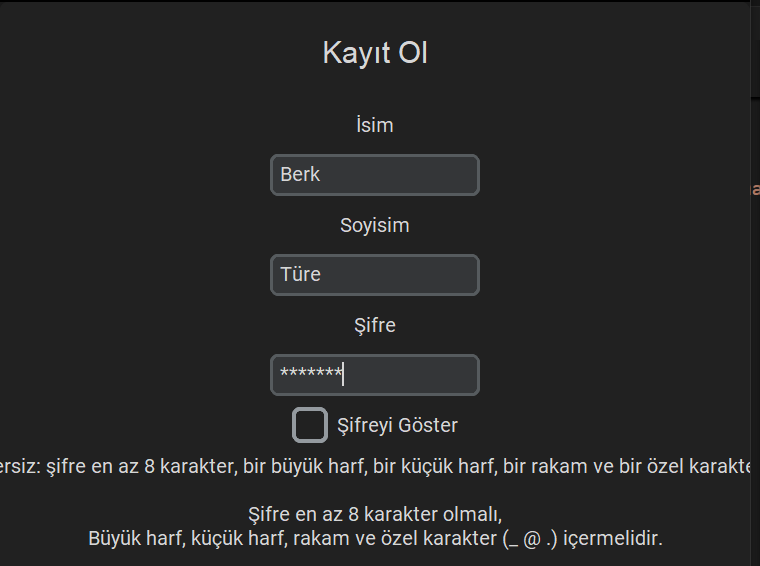
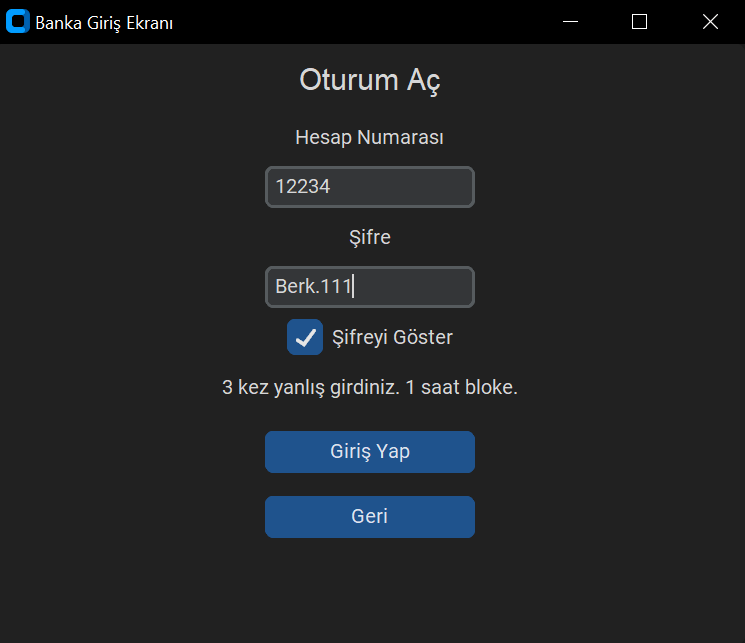
Butonlardan ilki oturum açma butonu

Eğer biz bankaya daha önceden bir kayıt oluşturduysak oturum açma butonuna tıklarız Daha sonra bize bankanın random(rastgele) ve unique(benzersiz) bir şekilde atamış olduğu hesap numarası ve şifremizi gireriz Fonksiyonun kendi içinde girilen şifre hesap numarasına ait mi diye kontrol edilip bir dönüt verilir Eğer doğru ise kişiyi bir sonraki frame(çerçeve) aktarır Eğer yanlış ise kişinin şifreyi doğru girmesi için 3 hak verir Kişi 3 hak sonucunda da yanlış girerse kişinin hesabı 1 saatlik bloke edilir

****

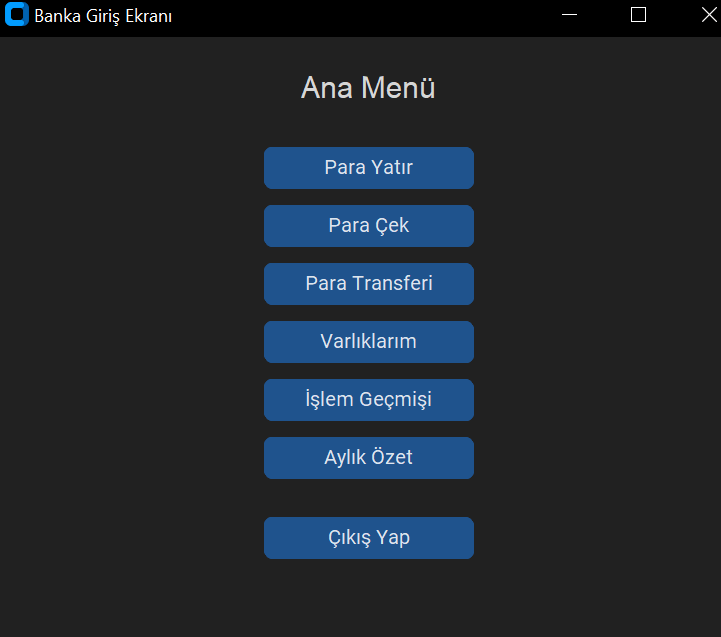
İkici butonumuz ise kayıt ol butonudur Bu butonun arkasında yatan fonksiyonda kişiden entry() ile isim soyisim ve şifresi istenir Kişi eğer koşullara uygun bir şifre girerse hesap numarası atanır Eğer koşullara uygun olmayan bir şifre girerse şifrenin hatalı olduğunu ve yeniden girilmesi gerektiğini bildirir Her iki frame (çerçevede)gözüktüğü gibi şifreyi göster butonu vardır Kişiler eğer bu butona tıklarlarsa girdikleri karakterler \* yerine “ “ olarak gösterilir ve şifreyi ne olarak girdikleri gözükür

****

** **

**5.4--Ana Menü—**

Kişiler oturumları doğru açtıktan sonra ana menüye yönlendirilirler Ana menüde birbirinden farklı butonlar ve her butonun arka planında çalışan fonksiyonlar vardır

****

**5.4.1—Para Yatırma—**

Kişiler ana menüye geldikten sonra eğer para yatırma tuşuna tıklarlarsa para yatır() fonksiyonu çalışır Bu fonksiyonda kişiden paranın hangi hesabına yatırılacağı ve yatıracağı miktarı girmesi istenir(girilen miktar str() olarak girileceğinden onu float(miktar) yapmalıyız) Bazı hesap türlerine göre kurallar vardır Örneğin vadeli hesaba 1000 Tl den az bir miktar yatırılamaz Eğer yatırılmaya çalışılırsa uyarı verir



**5.4.2--Para Çekme—**

Para çekme fonksiyonu da aynı para yatırma fonksiyonu mantığında çalışır kişi önce hesabın türünü daha sonra çekilecek olan paranın miktarını(str() olarak) girer

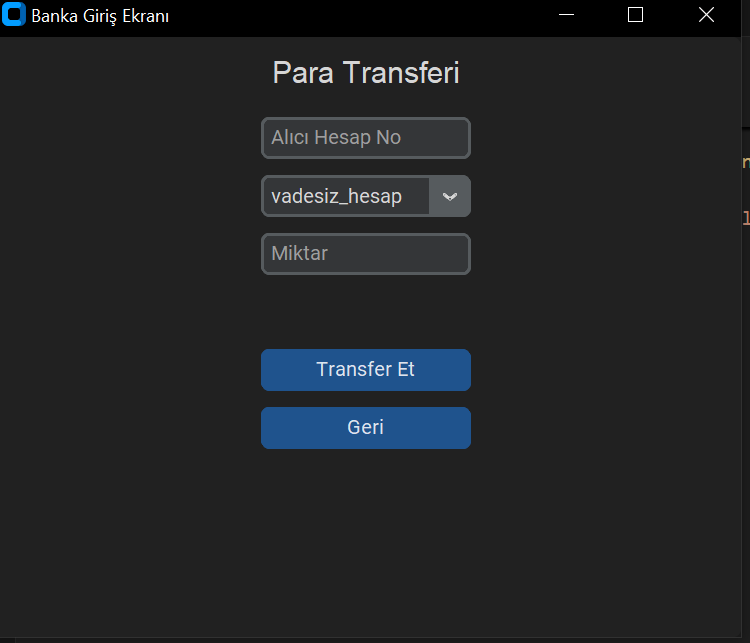
Eğer hesap türü vadeli veya bireysel emeklilik ise özel kurallar geçerlidir Bu kurallara uyulmaz ise ceza durumu oluşabilir

****

**5.4.3—Para Transferi—**

Para transferi olayı da aynı diğer iki fonksiyon mantığında çalışır Alıcının hesap numarası paranın çekilip aktarılacağı hesabın türü ve miktarı alınır(str() olarak)

Eğer aynı banka ise HAVALE farklı banka ise EFT işlemi uygulanır

****

**5.4.4—Varlıklarım/İşlem Geçmişi/Aylık Özet—**

Kişilerin hesap türlerinde bulunan paraların pasta grafiğinde yüzde olarak gösterilmesini sağlayan fonksiyon varlıklarım fonksiyonudur Grafik çizmek için matplotlib kütüphanesini kullandık

Kişilerin yapmış olduğu işlemlerin saat gün ay ve yıl olarak faturalandırılmış bir şekilde tutulmasını sağlayan fonksiyon ise işlem geçmişi fonksiyonudur İşlem geçmişinde tutulan yatırılan miktar ile çekilen miktarın yüzdelik oranlarını kişiye her ay otomatik güncellenerek bir pasta grafiği gibi verilmesini sağlayan fonksiyon ise aylık özet fonksiyonudur

**5.4.5—Çıkış—**

Çıkış butonunun fonksiyonu hazır fonksiyondur Butona basılınca uygulama kapanır



**TRABZON ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM FAKÜLTESİ**

**YAPAY ZEKA MÜHENDİSLİĞİ**

**VİZE PROJESİ**

**RAPORUMUZU OKUDUĞUNUZ İÇİN TEŞEKKÜR EDERİZ**

**BERK COŞKUN TÜRE (242301037)**

**BURAK YILDIRIM (242301038)**

**İBRAHİM ENES KAYA**

**TEVFİK OLGUNOĞLU**

**11/05/2025**