

Programlamaya Giriş

HAFTA 2

Yazılım Geliştirme Yaşam Döngüsü

Prof. Dr. Cemil ÖZ

Doç. Dr. Cüneyt BAYILMIŞ

Dr. Öğretim Üyesi Gülüzar ÇİT

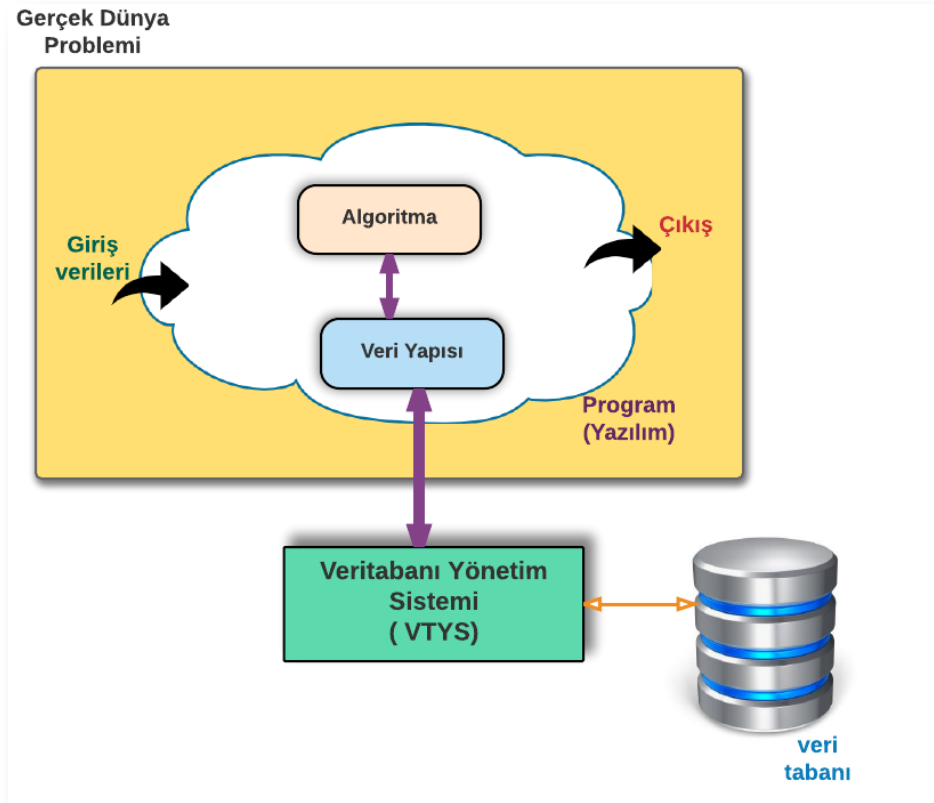
Konu & İçerik

- Büyük Resim – Algoritmik Problem
- Yazılım Geliştirme Yaşam Döngüsü
- ATM Sistemi İçin Yazılım Geliştirme
 - Gereksinim Analizi
 - Modelleme & Tasarım Aşaması
 - UML Dili
 - Kullanım Senaryosu (Use Case Diagram)
 - Etkinlik Şeması (Activity Diagram)
- Sözde Kod ve Akış Şeması
- Kaynaklar

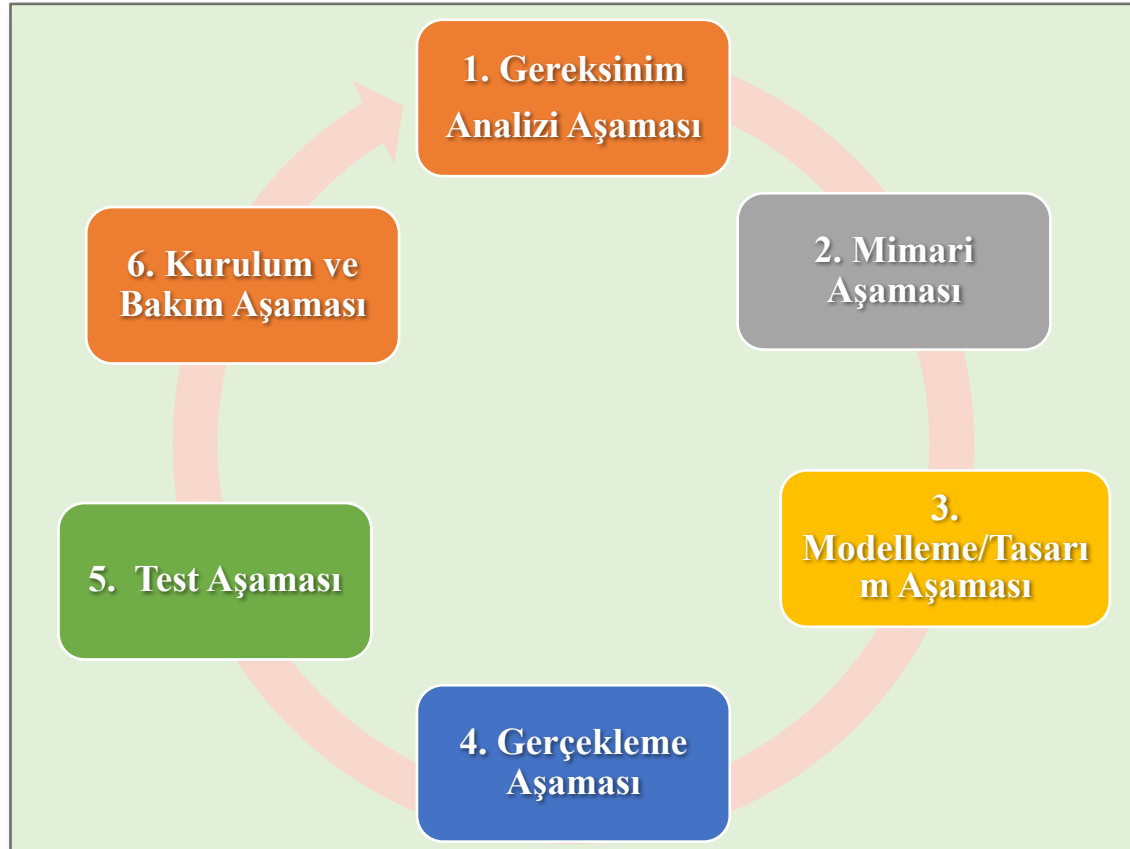


Büyük Resim – Algoritmik Problem

- Gerçek dünya problemlerini bilgisayar yardımıyla çözmek için **yazılım (program)** geliştirilir



Yazılım Geliştirme Yaşam Döngüsü



ATM Sistemi için Yazılım Geliştirme

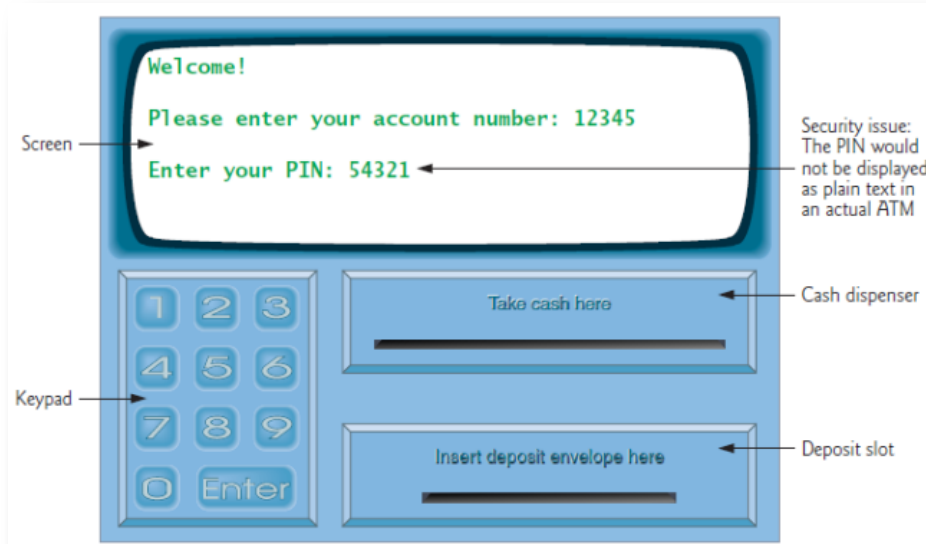
- Yerel bir banka, kullanıcılarına (yani, banka müşterilerine) temel finansal işlemlerini gerçekleştirebilmelerine izin veren yeni bir ATM yazılımı yüklemek istiyor.
- Her kullanıcının bankada sadece bir hesabı olabilir.
- ATM, kullanıcıların hesap bakiyelerini görmelerine, hesaplarından para çekmelerine ve hesaplarına para yatırmalarına izin vermelidir.

ATM Sistemi için Yazılım Geliştirme...

➤ Gereksinim Analizi Aşaması

➤ **ATM** makinasının kullanıcı arayüzü aşağıdaki donanım bileşenlerinden oluşmalıdır:

- **Tuş takımı** kullanıcının girdiği bilgiyi okur
- **Ekran** kullanıcıya mesaj görüntüler
- **Para yatırma bölmesi** kullanıcıdan para zarfını alır
- **Para verme bölmesi** kullanıcıya para verir

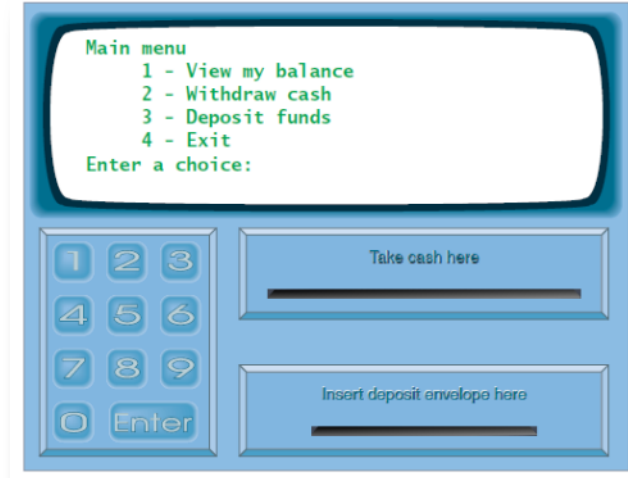


ATM Sistemi için Yazılım Geliştirme...

➤ Gereksinim Analizi Aşaması...

➤ ATM ye ilk girişte (şu anda hiç kimsenin kullanmadığı varsayılarak), kullanıcı aşağıdaki işlem adımlarını gerçekleştirir.

1. Ekran bir «Hoşgeldiniz!» mesajı görüntüler ve kullanıcıdan bir hesap numarası girmesini ister.
2. Kullanıcı tuş takımını kullanarak beş basamaklı hesap numarasını girer
3. Ekran, kullanıcıdan ilgili hesap numarası ile ilişkili PIN'i girmesini ister
4. Kullanıcı, tuş takımını kullanarak beş basamaklı PIN'i girer.
5. Eğer kullanıcı geçerli bir hesap numarası ve bu hesap için doğru PIN girdi ise ekranda ana menü görüntülenir. Eğer kullanıcı yanlış hesap numarası ve(ya) PIN girdi ise, ekranda bir mesaj görüntülenir ve ATM kimlik denetimini yeniden başlatmak için **ADIM 1**'e geri döner.



ATM Sistemi için Yazılım Geliştirme...

➤ Gereksinim Analizi Aşaması...

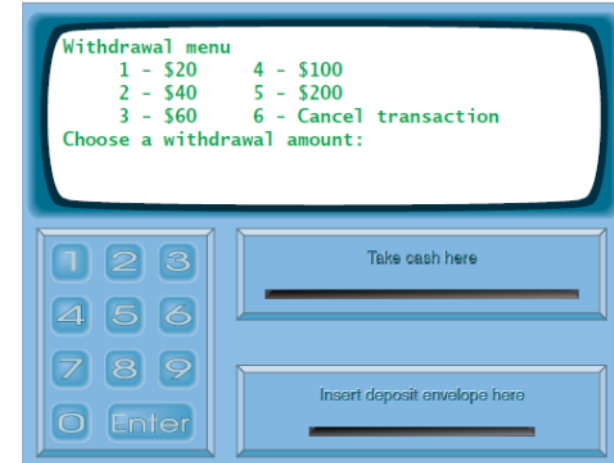
- ATM, kullanıcının kimlik denetimini tamamladıktan sonra, ana menü, bakiye sorgulama (seçenek 1), para çekme (seçenek 2) ve para yatırma (seçenek 3) olmak üzere üç tip işlemin her biri için numaralandırılmış bir seçenek sunar.
- Ayrıca, kullanıcının sistemden çıkabilmesi için de bir seçenek içermelidir (seçenek 4)
- Kullanıcı, daha sonra gerçekleştireceği işlemi (1, 2 veya 3'ü tuşlayarak) veya sistemden çıkmayı (4'ü tuşlayarak) seçer.
- Eğer kullanıcı bakiye görüntülemek için 1'i tuşlarsa, ekran kullanıcının hesabını görüntüler. Bunu gerçekleştirmek için, ATM'nin bankanın veri tabanından hesap bilgilerini alması gerekir.

ATM Sistemi için Yazılım Geliştirme...

➤ Gereksinim Analizi Aşaması...

➤ Aşağıdaki adımlar kullanıcının para çekmek için 2'yi tuşlamasından sonra gerçekleşecek olanları tanımlar:

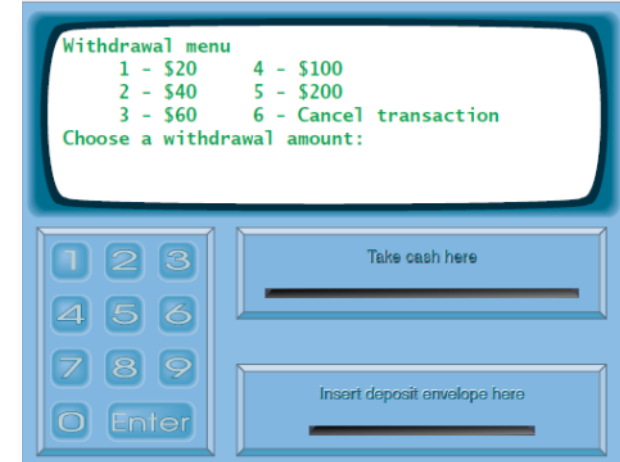
1. Ekranda, standart para çekme miktarlarını seçmek için belirlenen tuşları gösteren bir menü görüntülenir. Menü ayrıca kullanıcının işlemi iptal edebilmesine izin veren bir seçenek (seçenek 6) de içerir.
2. Kullanıcı, tuş takımını kullanarak bir menü seçimi yapar.



ATM Sistemi için Yazılım Geliştirme...

➤ Gereksinim Analizi Aşaması...

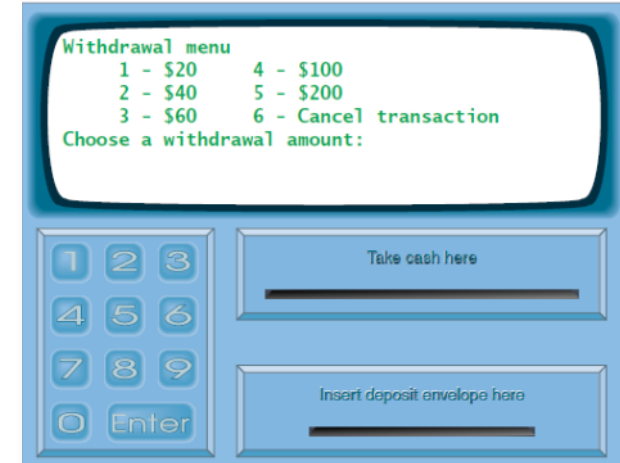
3. Eğer, çekilmek istenen para miktarı, kullanıcının hesap bakiyesinden büyükse, ekranda bu durumu belirten ve kullanıcıya daha düşük bir miktar girmesini söyleyen bir mesaj görüntülenir. ATM, daha sonra **Adım 1**'e döner. Eğer çekilmek istenen para miktarı, kullanıcının hesap bakiyesinden küçük veya eşitse (yani, kabul edilebilir bir miktarsa), ATM, **Adım 4**'e gider. Eğer kullanıcı, işlemi iptal etmeyi seçerse (seçenek 6), ATM, ana ekranı görüntüler ve kullanıcı girişini bekler.
4. Eğer, bankamatik yeterli para içeriyorsa, ATM, **Adım 5**'e gider. Aksi halde, ekranda, bankamatikte yetersiz para olduğunu belirten ve kullanıcıya daha düşük bir para çekme miktarı girmesini söyleyen bir mesaj görüntülenir. ATM, **Adım 1**'e geri döner.



ATM Sistemi için Yazılım Geliştirme...

➤ Gereksinim Analizi Aşaması...

5. ATM, çekilen para miktarını bankanın veritabanındaki kullanıcı hesabından düşer (yani, kullanıcının hesap bakiyesinden çekilen para miktarını çıkartır).
6. Bankamatik, hesaplanan para miktarını kullanıcıya verir.
7. Ekranda, kullanıcıya para almasını hatırlatan bir mesaj görüntülenir.



ATM Sistemi için Yazılım Geliştirme...

➤ Gereksinim Analizi Aşaması...

➤ Aşağıdaki adımlar kullanıcı para yatırmak amacıyla 3'ü tuşlayınca gerçekleşen adımları tanımlar:

1. Ekran, kullanıcının yatıracağı para miktarını sorar veya iptal etmek için 0'ı tuşlamasını ister.
2. Kullanıcı, tuş takımını kullanarak yatırılacak para miktarını girer veya 0'ı tuşlar.
3. Eğer kullanıcı yatırılacak parayı girerse, ATM, Adım 4'e gider. Eğer kullanıcı işlemi iptal etmeyi seçerse (0 tuşlayarak), ATM ana ekranı görüntüler ve kullanıcı girişi için bekler.
4. Ekran, yatırılacak parayı yerleştirmesini söyleyen bir mesaj görüntüler.
5. Eğer, para yatırma bölmesi iki dakika içerisinde parayı alırsa, ATM , para miktarını bankanın veritabanındaki kullanıcının hesabına ekler. Eğer para yatırma bölmesi bu zaman diliminde yatırılacak parayı alamazsa, ekran, sistemin eylemsizlikten dolayı işlemi iptal ettiğini belirten bir mesaj görüntüler. ATM, daha sonra ana menüyü görüntüler ve kullanıcı girişi için bekler.

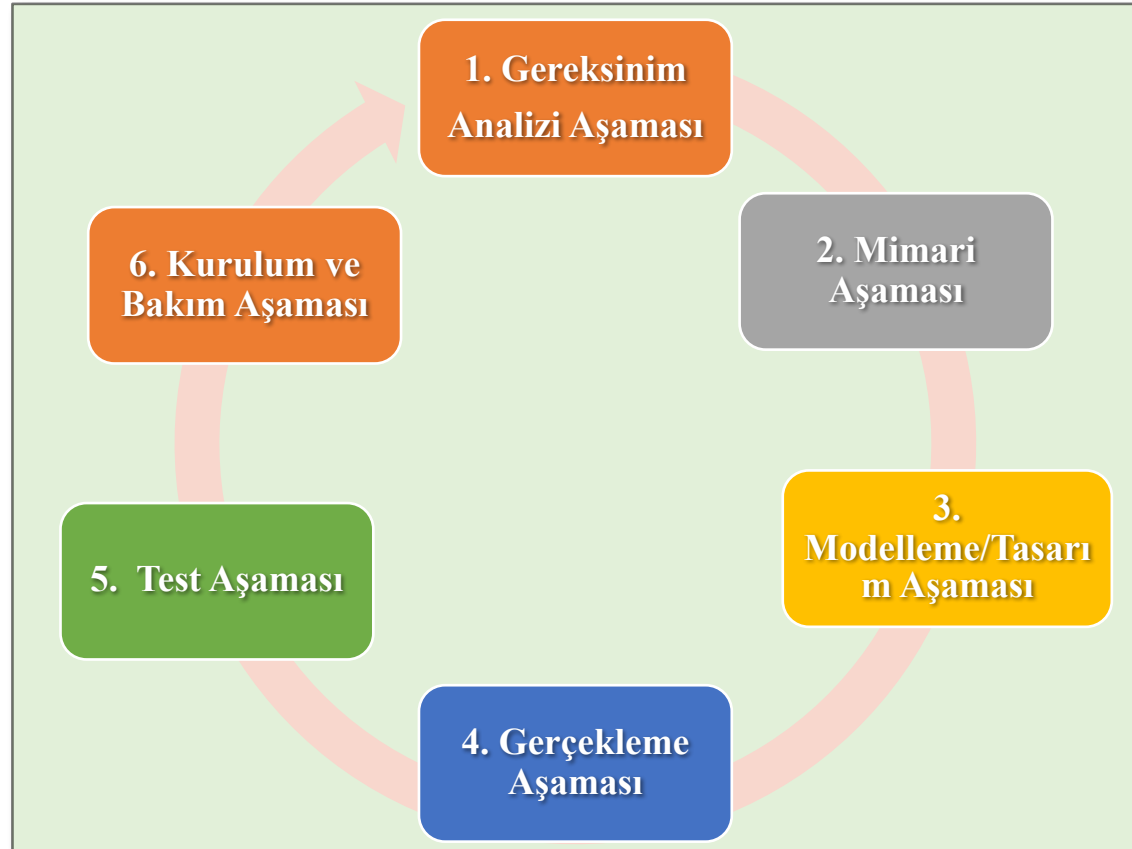
ATM Sistemi için Yazılım Geliştirme...

➤ Gereksinim Analizi Aşaması...

- Sistem, başarılı şekilde bir işlemi gerçekleştirdikten sonra, ana menüye dönmelidir ki kullanıcı başka işlemler de yapabilsin. Eğer kullanıcı sistemden çıkarsa, ekran bir teşekkür mesajı görüntüler ve daha sonra sonraki yeni kullanıcı için bir hoş geldin mesajı görüntüler.

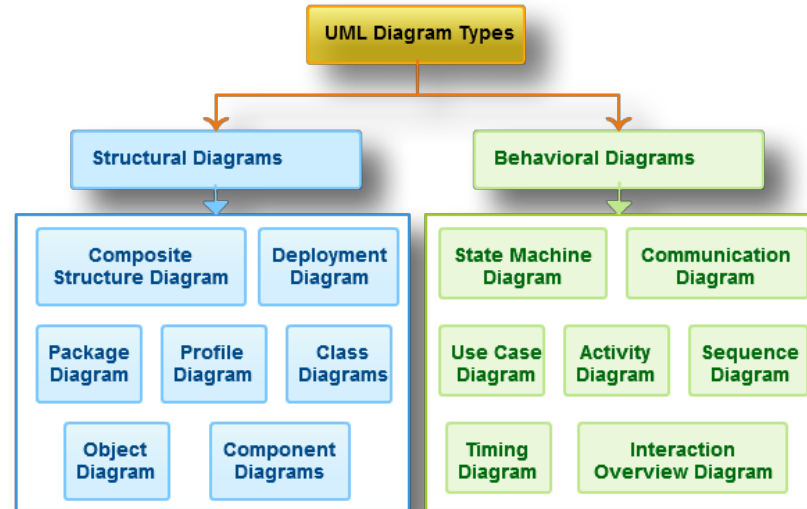
Yazılım Geliştirme Yaşam Döngüsü

➤ Yazılım Geliştirme Yaşam Döngüsü



Modelleme/Tasarım Aşaması...

- Yazılımlar iki temel amaç için modellenirler:
 - Modelden kodlamaya geçmek daha kolaydır.
 - Farklı aktörlerin (yazılım geliştiriciler, kullanıcılar, proje yöneticileri, şirket yöneticileri v.s.) haberleşebilmeleri kolaylaşır.
- Günümüzdeki yazılımları (özellikle de nesne yönelimli olarak geliştirilenleri) modellemek için **UML** (Unified Modelling Language – Birleşik Modelleme Dili) kullanılır.

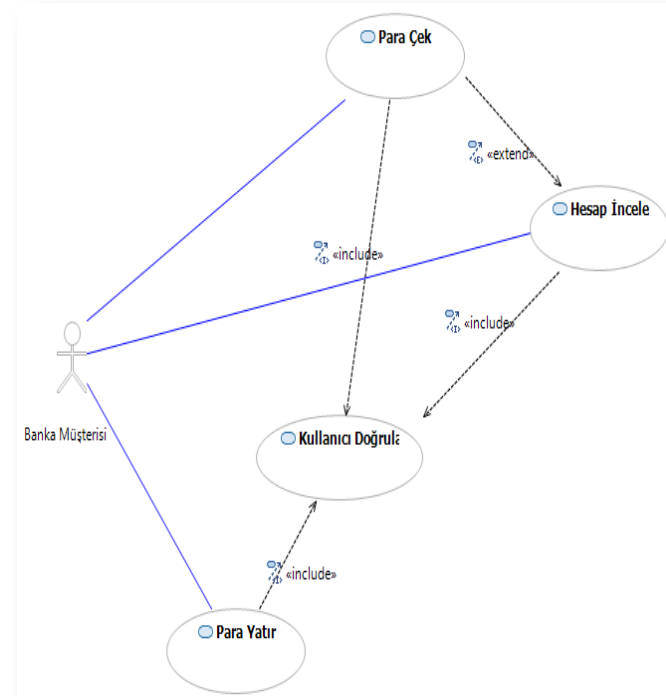


[online diagramming & design] creately.com

Modelleme/Tasarım Aşaması...

➤ Kullanım Senaryoları (Use Case Diagram)

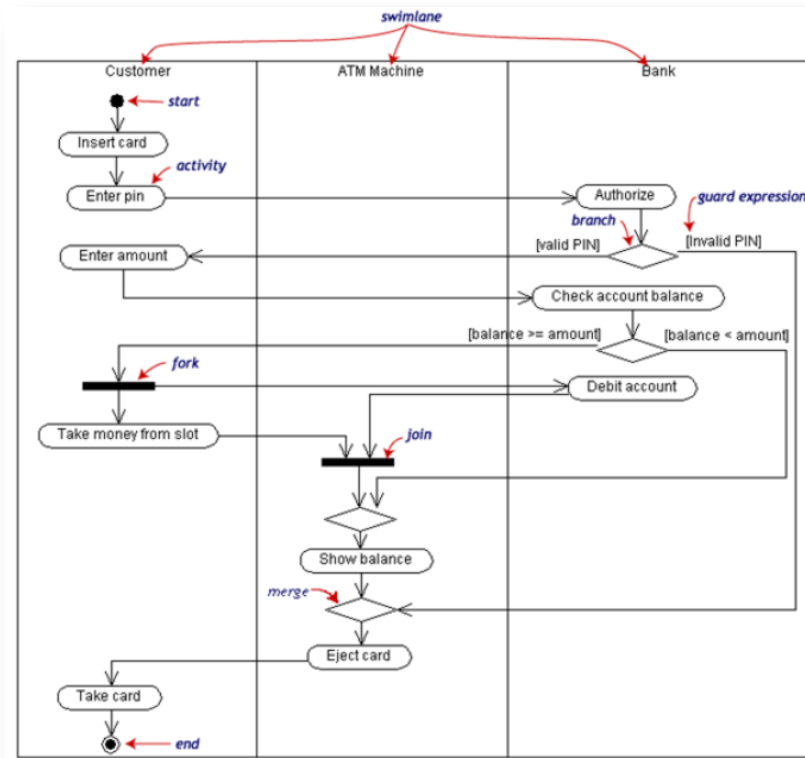
- Bir kullanıcı ve bir sistem arasındaki etkileşimi anlatan senaryo topluluğu
- Davranışsal şemalardandır.
- Aktör belirli bir görev için olayları başlatan nesnedir. (Banka müşterisi, operatör, banka v.s.)
- Kullanım senaryolarında sistem aktör açısından ne yapar ile ilgilenir. Nasıl yapar ile ilgilenmez.
- Senaryo; aktör sistem ile etkileşime girdiğinde neler olduğunu anlatır.
- **Use case**; tek bir görev ya da hedef için senaryonun özetlenmiş halidir. Use case'ler birleşerek senaryoyu oluşturur. (Para çekme bir use case dir)



Modelleme/Tasarım Aşaması...

➤ Etkinlik Şeması (Activity Diagram)

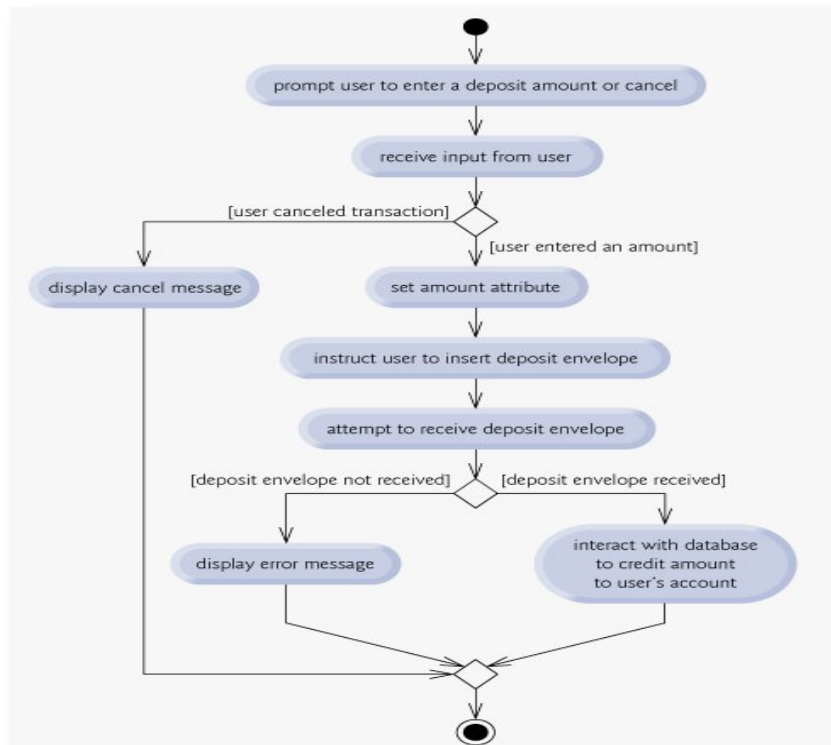
- Sistemin dinamik görünümünün modellendiği diyagramlar
 - Para çekme senaryosuyla ilgili şema



Modelleme/Tasarım Aşaması...

➤ Etkinlik Şeması (Activity Diagram)

- Sistemin dinamik görünümünün modellendiği diyagramlar
 - Para yatırma senaryosuyla ilgili şema



Modelleme/Tasarım Aşaması...

- Sözde kod ve akış şemaları da yazılımların algoritmalarını ifade etmek için kullanılan kısıtlı modelleme yaklaşımlarıdır.
- Akış şemaları, UML'deki etkinlik şemaları gibidir. Etkinlik şeması, yapısal programlamadaki modelleme standartlarından olan veri akış şemaları (dfd – data flow diagram) ve akış şemalarının birleşiminden oluşur.

Kaynaklar

- Deitel, P., Deitel, H., Java How To Program, Prentice Hall, Ninth Edition, 2012.