**ATM SİSTEMİ**

İbrahim Kurt, 140757009

Atatürk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü

**İÇİNDEKİLER**

1.Özet

2.Tasarım Bakışı

2.1.Class Diyagramı

2.2.Nesne Diyagramı

3.İşlem Bakışı

3.1.Aktivite Diyagram

3.2.Durum Diyagramı

3.3.Sequence Diyagramı

3.4.Collaboration Diyagramı

4.Gerçekleştirme Bakışı

3.1.Component Diyagramı

5.Dağılım Bakışı

4.1.Deployment Diyagramı

6.Use Case Diyagramı

7.Flowchart Diyagramı

8.Communication Diyagramı

9.Er Diyagramı

10. Profile Diyagram

11. Composite Diyagram

12.Package Diyagram

13.Kodlar

14.Kaynakça

**ÖZET**

ATM Sistemi para yatırma, para çekme , ödeme vb işlemleri elektronik ortama dökerek güvenilirliği arttıran iş gücünü azaltan bir sistemdir. UML (Unified Modelling Language, Birleşik Modelleme Dili) yazılım modellenmesi ve planlanması için kullanılan standart bir dildir.

UML, yazılım ağırlıklı bir sistemi ve bu sistemin parçalarını gözde canlandırmak, belirtmek, kurmak ve belgelemek için kullanılabilir. Kurumsal bilgi sistemlerinden, dağıtımlı ağ-tabanlı uygulamalara ve gerçek zamanlı gömülü sistemlere kadar birçok sistemi modellemek için uygun bir dildir. UML bir programlama ya da yazılım geliştirme dili değil, sistem modellemelerinde kullanılan yöntemlerin bir araya getirilmiş halidir.

**1.GİRİŞ**

Bu projede atm sistemini StarUML ile gerçekleyerek UML’nin faydalarını göreceğiz.Projeyi anlaşılabilirliği kolay olması için 4+1 bakış ile yapıldı. 4+1 bakışı hakkında bilgi verecek olursak 5 farklı bakıştan oluşur bunlar tasarım bakışı,işlem bakışı, gerçekleştirme bakışı, dağılım bakışı, use case bakışı olarak sıralanabilir.

Tasarım Bakışı(Desing view): Tasarım bakışı sistemin yapıla bakışı olarak tanımlayabiliriz. Sınıf(Class) ve nesne (Object) diyagramları sistemin tasarım bakışını oluşturur.

İşlem Bakışı(Process view): Sistemin dinamik davranışlarını işlem bakışında sınıflandırılır. Durum(State), aktivite(activity), ardışık(sequence) ve işbirliği(collaboration) diyagramları bu bakışta kullanılır.

Gerçekleştirme Bakışı(Component View): Bileşen(Component) diyagramları kullanılarak sistem modellemesinde modüllerin gruplandırılmasında gerçekleştirme bakışı kullanılır.

Dağılım Bakışı (Deployment View):sistemin donanımsal olarak dağılımını gösteren bakış açısıdır. Dağılım(Deployment) diyagramlar dağılım bakışında kullanılmaktadır.

Use case Bakışı(Use case View): +1 de yeralan bakış açısı ise use case bakışıdır. Use case diyagramlarda ve senaryolarda kullanılır.

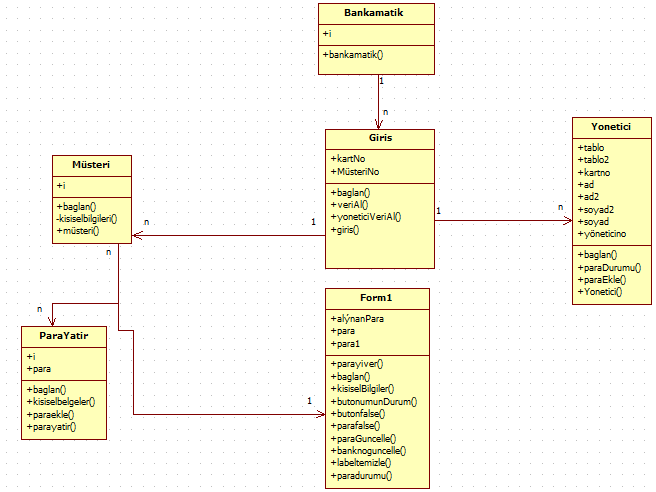
**2. GELİŞME** (Times New Roman Kalın 12pt)

### 2.1 Yapı Diyagramları

UML’de 6 tür yapı diyagramı vardır.

**2.1.1 Sınıf (Class) Diyagramları**

Sınıf diyagramları (Class Diagram) projemizdeki sınıfların, özelliklerinin ve birbirleri arasındaki ilişkilerin gösterilmesi için kullanılır. Özelliğin tipi :byte ve :string şeklinde belirtilmiştir.TekerlekSayisi özelliğinin default değeri 4 olarak belirtilmiştir. + işareti public (diğer sınıflardan da erişilebilir) seviyesinde olduğunu gösterir.# işareti protected (sınıf içinden ve kendisinden türeyen sınıflardan erişilebilir) seviyesinde olduğunu gösterir.- işareti private (sadece sınıf içinden erişilebilir) seviyede olduğunu belirtir.



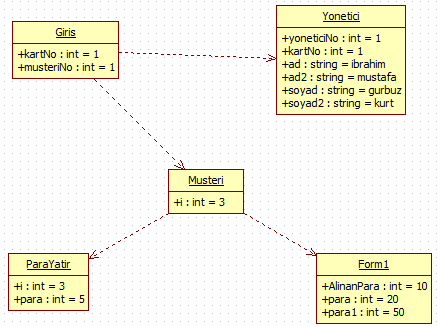
Şekil 1

**2.1.2 Nesne Diyagramları**

Bir nesne sınıfının bir örneğidir.Bu diyagramlarda sınıfın yerine gerçek nesneler kullanılır.

Modellenen sistem yapısının belirli bir andaki bütün yada kısmi görünüşünü anlatır(Run time).

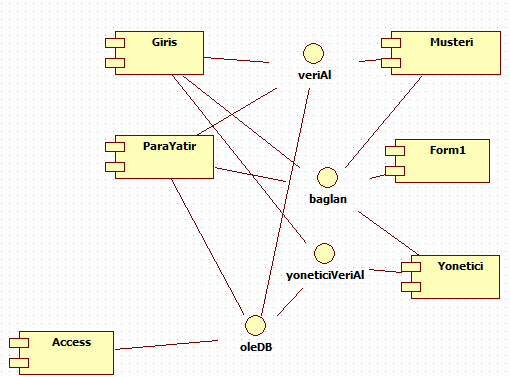
Nesne adı, sınıf adı, öznitelik ve sınıfın method parametrelerinin alacağı değerleri gösterir.



Şekil 2

**2.1.3 Bileşen (Component) Diyagramları**

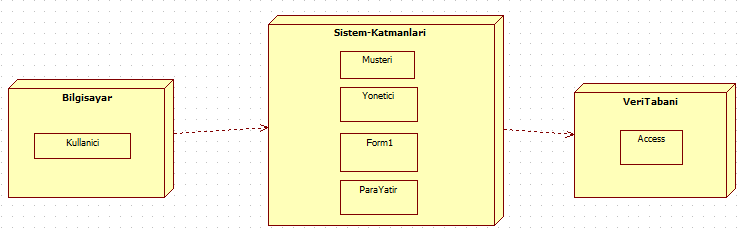
Paket diyagramların fiziksel anlatımıdır.Basitçe paketler yerine paket içinde yer alan exe,.dll dosyaları ve bunlar arası etkileşimin gösterildiği bir diyagramdır.Paket diyagramlar sistemin analizini ve tasarımını aşamasında kullanılabilirken Bileşen diyagramları programlama bittikten sonra sistemi tarif eder niteliktedir.



Şekil 3

**2.1.4 Dağıtım (Deployment) Diyagramları**

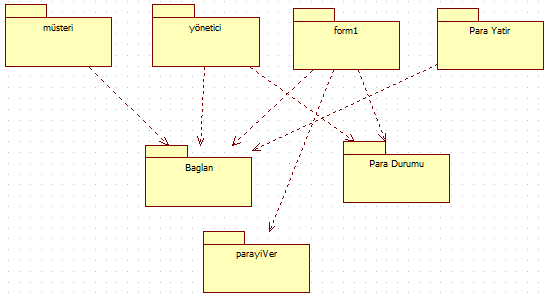
Yazılımın nasıl dağıtılacağının planlandığı aşamada yardımcı olurlar. Yazılımın kurulacağı bilgisayarların konfigürasyonu , ağ ve yazıcı bağlantıları gibi detayları kapsamaktadır.



Şekil 4

**2.1.5 Paket Diyagramları**

Paket diyagramları büyük yazılımlarda sistemi oluşturan alt yazılımlar veya etkileşimde bulunan yan sistemler olduğu durumda bu paketler arası etkileşimi gösteren kısaca sistem mimarisini özetleyen kavramsal bir diyagramdır.Aşağıda ses kaydedicide alt sistemleri belirtilen paket diyagramı örneği verilmektedir.

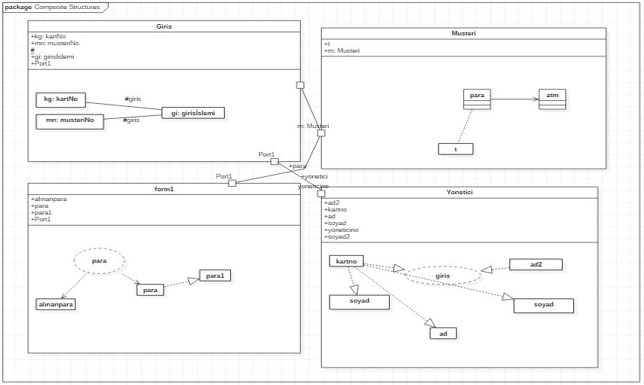


Şekil 5

**2.1.6 Birleşik Yapı (Composite Structure)**

Birleşik yapıların iki ayrı türü vardır. Toplama, bir araya getirme (Aggregation), Meydana gelme (Composition). Toplama (Aggregation) İçerilen nesneler (alt parçalar) kendi başlarına da kullanılırlar. Sadece o nesneye ait parçalar değillerdir. Örneğin havaalanında uçaklar vardır. Meydana gelme (Composition) Alt parçalar o nesneyi meydana getirmek için oluşturulmuşlardır; kendi başlarına kullanılmazlar.

Örneğin otomobilin motoru vardır.



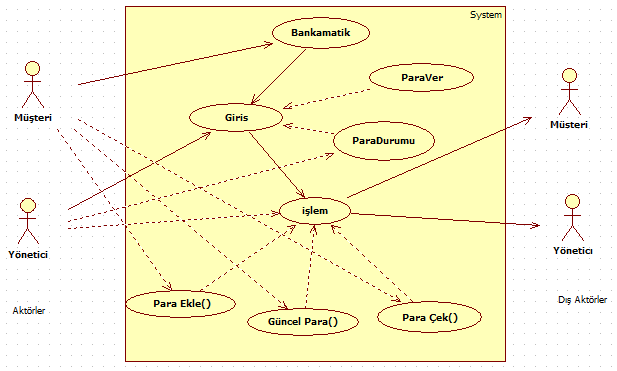
Şekil 6

**2.2 Eylem Diyagramları**

UML’de 3 tür eylem diyagramı vardır.

**2.2.1 Kullanım Durumu (UseCase) Diyagramları**

Bir uygulamanın yada sistemin kimler tarafından kullanıldığını ve söz konusu kullanıcıların bu uygulama yada sistemle neler yapabildiklerini özetlemek amacıyla kullanılan bir UML diyagramıdır.

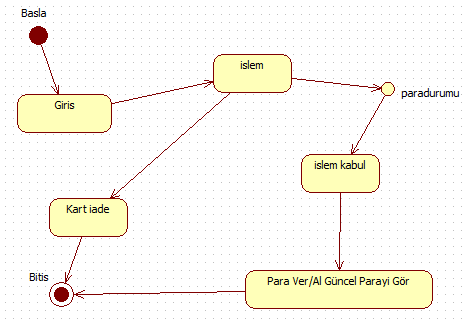


Şekil 7

**2.2.2 Durum(State) Diyagramları**

Bir nesnenin ömrü boyunca sahip olduğu durumları modelleyen diyagramlardır.Örneğin:Bir lamba yaşam süresince 3 farklı durumda olabilir.Birinci durum:Lambanın kapalı olması İkinci Durum:Lambanın açık olması Üçüncü durum:Lambanın düşük enerji harcayarak yanması olsun.Bu üç durum arasında geçişler mümkündür.Bu geçişleri sağlayan ise çeşitli olaylardır.Örneğin lambanın kapatmak için anahtarı çevirmek gerekir.Açmak içinde anahtarı çevirmek gerekir.Düşük enerjili halde yakmak içinde anahtarı yarım çevirmek gerekir.Lamba bizim nesnemiz durum değişirken yaptığı işlere de action bu işleri sağlayan mekanizmayada event(olay) denir.

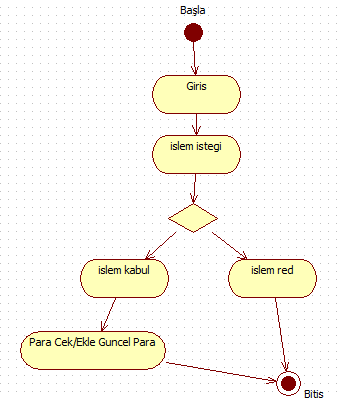
Bir durum köşeleri yuvarlanmış dörtgen şeklinde gösterilir.Bir durumun 3 ana bileşeni vardır.Birinci bileşeni Durumun Adı,ikinci bileşeni Durumun değişkenleri sonuncu ise durumla ilgili çeşitli aktiviteleri belirlemek için kullanılan yapı.



Şekil 8

**2.2.3 Aktivite Diyagramları**

Sistemin yaşamı boyunca alabileceği tüm durumlar durum diyagramları ile belirtilirken ilgili durumda yapılması gereken aktivite aktivite diyagramları sayesinde ortaya konur. Aktivite diyagramları bir davranışın giriş ve çıkışlarını göstermektedir.Oluşturulacak modelde giriş ve çıkışlar önemliyse aktivite diyagramlarını tercih etmek gerekir.



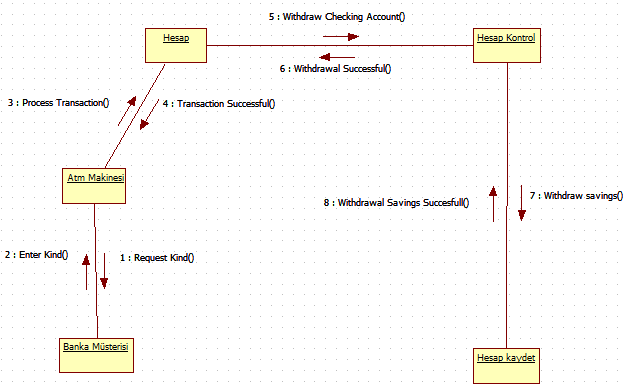
Şekil 9

**2.3 Etkileşim Diyagramları**

UML' de iki tür etkileşim diyagramı vardır. İletişim Diyagramları (Collaboration Diagrams), Ardışıl Diyagramlar (Sequence Diagrams)

**2.3.1 İletişim(Collaboration) Diyagramları**

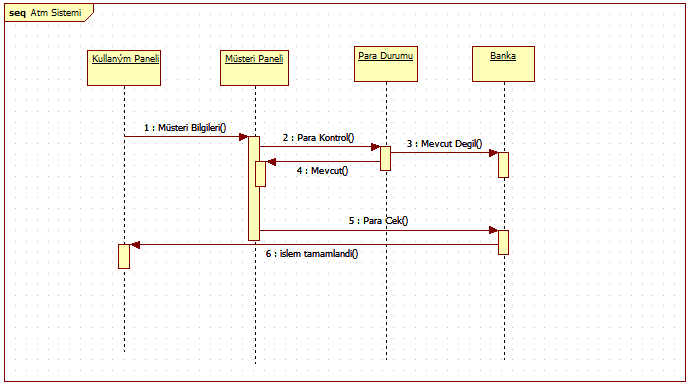
Nesneler arası etkileşim bir graf şeklinde gösterilir. Özellikleri az yer kaplar, mesajların dallanmalarını göstermek kolaydır , mesajların sıralarını anlamak zordur.



Şekil 10

**2.3.2 Ardışıl(Sequence) Diyagramlar**

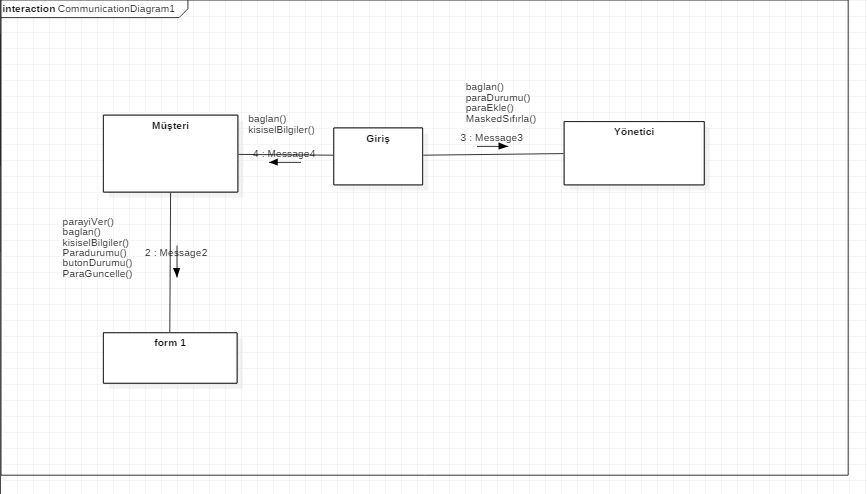
Nesneler yan yana gösterilir. Etkileşimler oluşturdukları sıra ile yukarıdan aşağıya doğru çizilir. Her yeni nesne çizimin sağına eklenir, fazla yer kaplar, mesajların zaman içinde sıralarını anlamak daha kolaydır.



Şekil 11

**2.4 Haberleşme (Communication) Diyagramları**

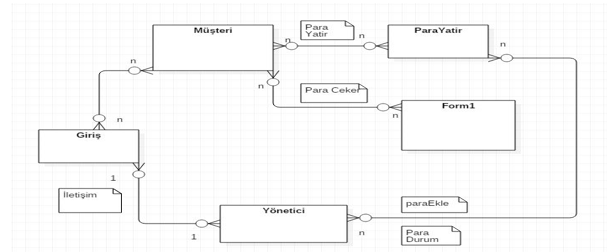
Nesneler arası etkileşim bir graf şeklinde gösterilir. Özellikleri az yer kaplar, mesajların dallanmalarını göstermek kolaydır , mesajların sıralarını anlamak zordur.



Şekil 12

**2.5 ER Diyagramları**

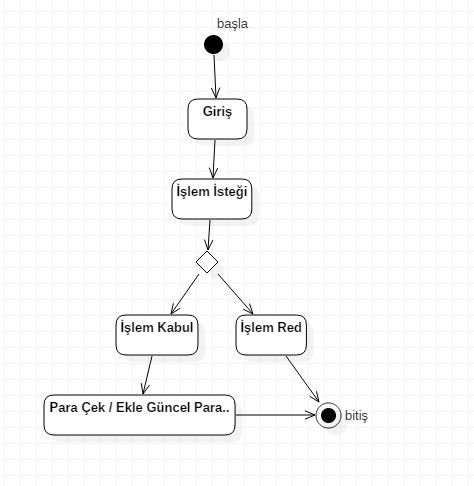
Bu diyagramda, [veritabanı](http://bilisim-kulubu.com/sozluk/sozluk.php?e=veritaban%FD) içinde bulunan bütün [tablo](http://bilisim-kulubu.com/sozluk/sozluk.php?e=tablo) ve ilişkiler belirtilir. Veritabanının genel yapısı buradan kolaylıkla anlaşılabilir. Büyük sistemler için oldukça gereklidir. Bu sayede sisteme hakimiyet sağlanması daha kolaydır. Şema olduğu için herhangi veri tabanı için özel değildir. Bu şema ile istenen veri tabanı için oluşturulabilir.



Şekil 13

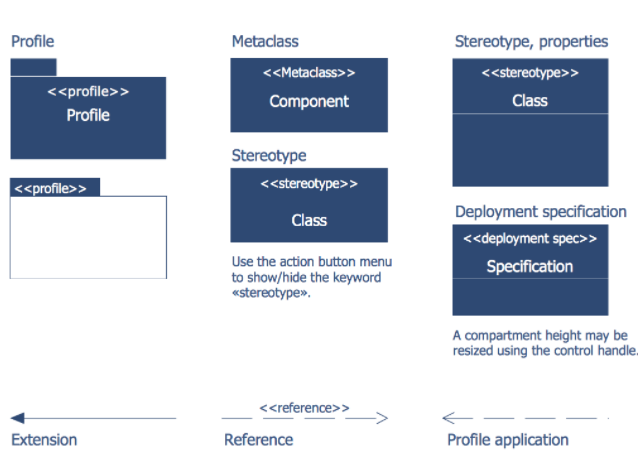
**2.6 Flow Diyagramları**

Sistemin yaşamı boyunca alabileceği tüm durumlar durum diyagramları ile belirtilirken ilgili durumda yapılması gereken aktivite aktivite diyagramları sayesinde ortaya konur. Aktivite diyagramları bir davranışın giriş ve çıkışlarını göstermektedir.Oluşturulacak modelde giriş ve çıkışlar önemliyse aktivite diyagramlarını tercih etmek gerekir.



Şekil 14

**2.7 Profile Diyagramları**



Şekil 15

**3. SONUÇLAR**

Bir bileşen kendi ortamında değiştirilebilen modüler bir birimdir. Kendi iç yapıları gizlidir, ancak olduğu bir veya daha iyi tanımlanmış sağlanan arayüzler işlevleri erişilebilen üzerinden. Bir bileşen ayrıca sahip gerekli arabirimleri. Gerekli arabirim, diğer bileşenlerden hangi işlevleri veya hizmetleri beklediğini tanımlar. Birçok bileşenin sağlanan ve gerekli arabirimleri bağlanarak daha büyük bir bileşen oluşturulabilir. Tam bir yazılım sistemi bir bileşen olarak anlaşılabilir.

**4.KAYNAKLAR**

1.[http://www.sparxsystems.eu/resources/project-development-with-uml-and-ea/interaction-diagram/](http://l.facebook.com/l.php?u=http%3A%2F%2Fwww.sparxsystems.eu%2Fresources%2Fproject-development-with-uml-and-ea%2Finteraction-diagram%2F&h=lAQHlWyO8AQH2o8Vxlt_BYVUFdoXm93Yml3i_VUVduj86IQ)  
2. [http://www.csharpnedir.com/articles/read/?filter&author&cat=uml&id=462&title=UML+ile+Aktivite+%28Activity%29+Diyagramlar%C4%B1](http://l.facebook.com/l.php?u=http%3A%2F%2Fwww.csharpnedir.com%2Farticles%2Fread%2F%3Ffilter%26author%26cat%3Duml%26id%3D462%26title%3DUML%2520ile%2520Aktivite%2520(Activity)%2520Diyagramlar%25C4%25B1&h=rAQE0yGzyAQFZoz9S6bDtXb1wZHEoVUrMiWaFiHJ17eYU6w)  
3. [http://www.uml-diagrams.org/index-examples.html](http://l.facebook.com/l.php?u=http%3A%2F%2Fwww.uml-diagrams.org%2Findex-examples.html&h=xAQGGsFmhAQGHRThA51spPpHfwiZlTZEHMyQzWPNtLeEAEQ)  
4. [http://web.bilkent.edu.tr/isbilkent/](http://l.facebook.com/l.php?u=http%3A%2F%2Fweb.bilkent.edu.tr%2Fisbilkent%2F&h=5AQHdH-lAAQHdVmt5fj_E0cDdjX6jJrRdBrDovi5I1KmLPg)  
  
5. [http://univera-ng.blogspot.com.tr/2010/04/uml-ve-modelleme-bolum-11-41-baks.html](http://l.facebook.com/l.php?u=http%3A%2F%2Funivera-ng.blogspot.com.tr%2F2010%2F04%2Fuml-ve-modelleme-bolum-11-41-baks.html&h=5AQHdH-lAAQFRpRxLufEihjUwWz2F9yax-LFs6tNaoLGNyQ)  
  
6. [https://tr.wikipedia.org/wiki/UML](https://l.facebook.com/l.php?u=https%3A%2F%2Ftr.wikipedia.org%2Fwiki%2FUML&h=KAQH-CbomAQHkea9x8DBAsBkSh8q6qHVZD-xsJz8obIa06A)

7. <http://www.conceptdraw.com/>