

**T.C.  
GEBZE TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**NESNE YÖNELİMLİ ve KULLANICI DOSTU  
ATM ARAYÜZ YÖNETİM SİSTEMİ**

**ORHAN ŞEN  
YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**GEBZE  
2015**

**T.C.**  
**GEBZE TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**NESNE YÖNELİMLİ ve KULLANICI DOSTU**  
**ATM ARAYÜZ YÖNETİM SİSTEMİ**

**ORHAN ŞEN**  
**YÜKSEK LİSANS TEZİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMANI**  
**DOÇ. DR. MEHMET GÖKTÜRK**

**GEBZE**  
**2015**

**T.R.**  
**GEBZE TECHNICAL UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**OBJECT ORIENTED and USER FRIENDLY ATM**  
**INTERFACE MANAGEMENT SYSTEM**

**ORHAN ŐEN**  
**A THESIS SUBMITTED FOR THE DEGREE OF**  
**MASTER OF SCIENCE**  
**DEPARTMENT OF COMPUTER ENGINEERING**

THESIS SUPERVISOR  
ASSOC. PROF. DR. MEHMET GÖKTÜRK

**GEBZE**  
**2015**



GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 21/01/2015 tarih ve 2015/05 sayılı kararıyla oluşturulan jüri tarafından 21/04/2015 tarihinde tez savunma sınavı yapılan Orhan ŞEN'in tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

**JÜRİ**

ÜYE

(TEZ DANIŞMANI) : Doç. Dr. Mehmet GÖKTÜRK

ÜYE

: Doç. Dr. Yusuf Sinan AKGÜL

ÜYE

: Öğr. Gör. Dr. Ahmet Cengizhan DİRİCAN

**ONAY**

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve ...../..... sayılı kararı.

İMZA/MÜHÜR

## ÖZET

Günümüzde artan servis ihtiyaçlarıyla birlikte Türkçede “Bankamatik” adı ile adlandırılan ATM (Automated Teller Machine) cihazlarının kullanımı giderek artmaktadır. Hizmet sağlayan servislerde görülen artış arayüz tasarımını zorlaştırmakta ve kullanıcı dostu olmaktan uzak arayüzler ortaya çıkmaktadır.

Bu çalışmada; ATM cihazlarının kişiye özel gelişmiş özelliklerle donatılarak gerek kullanıcıya kullanım kolaylığı, gerekse sistem modifikasyon ve güncelleme avantajları sağlayacak biçimde yazılım alt çerçevesinin oluşturulması ve bu çerçevede yeni bir ATM cihazı arayüz yönetim sistemi geliştirilmiştir. Sistem sayesinde kullanıcılara; ATM cihazlarında yeni nesil işlemlerin rahatlıkla tanımlanabileceği, banka personelleri tarafından hatasız yönetilebilen, arayüz değişimlerinde kullanım sorunları oluşturmayan ve kolay kullanılabilir bir arayüz yönetim ortamı sağlanmıştır. Geliştirilen sistem sayesinde; tasarımlara modülerlik ve ölçülebilir arayüz performans metriği ortaya çıkarılmış, bu metriğin deneysel sonuçları üzerine de analizler yapılmıştır. Çalışma kapsamında ayrıca senaryo adı verilen modüler tasarımlar üzerine kullanıcı merkezli işlem ölçümleri yapılmış olup, kullanıcı grupları çıkarılmıştır. Çıkan kullanıcı gruplarına özel ATM arayüz tasarımları oluşturularak da kullanıcılara daha verimli ve başarılı işlem arayüzleri sağlanmıştır.

Elde edilen sonuçlar 00794.STZ.2011-1 numaralı SAN-TEZ (Sanayi Tezleri Programı) kapsamında çalışılan banka ATM’leri üzerinde uygulanmış olup, ATM platform standartlarına ve çok dilli yapıya uygun olması sebebiyle bu sektördeki yerli ve yabancı müşterilere hitap etmesi hedeflenmiştir.

**Anahtar Kelimeler: İnsan Bilgisayar Etkileşimi, Kullanıcı Arayüz Yönetim Sistemleri, Arayüz Performansı, Arayüzde Kişiselleştirme, ATM.**

## SUMMARY

In line with the increasing service demands, the use of ATM's (Automated Teller Machine which is known as 'Bankamatik' in Turkish) has been increasing. The rise in the services provided to customer complicates the interface design and leads to unfriendly interfaces.

In this study, a new interface management system has been developed based on the software frame ensuring user friendliness to users through new personalised advanced facilities as well as some advantages of system modifications and updates of the ATM's. With this system, a new environment for interface management which the new-generation-transactions can easily be introduced and can be managed by bank personnell faultlessly and which does not create any problems during interface changing, has been provided to users as user friendly concept. With the help of this newly developed system, the modularity of designs is enabled, a measurable metrics of interface has been emerged, and the analyses based on the experimental conclusions of this metrics have also been performed. In the context of this study, the user groups have also been defined though the transaction measurements on modular designs which called 'scenarios'. Based on defined user groups, special ATM interfaces have been created for each group to ensure more effective and high performance transaction interface to users.

As the conclusions are compliant to ATM platform standards and multilingual structure, the conclusions derived from this study have been applied on the ATM's covered under 00794.STZ.2011-1 SAN-TEZ (Industry Theses Program) program to reach domestic and foreign customers.

**Keywords: Human Computer Interaction, User Interface Management Systems, Performance In Interface, Personalization In Interface, ATM.**

## TEŐEKKÜR

Yüksek lisans eğitimin ve tez çalışmam boyunca bana verdiği destek ile beni teşvik eden tez danışmanım Sayın Doç. Dr. Mehmet GÖKTÜRK'e ve edindiğim mühendislik bilgilerinin kaynağı olan Gebze Teknik Üniversitesi Bilgisayar Mühendisliği bölümü öğretim üyelerine,

SAN-TEZ proje teklifi ile tez çalışmamın gerçek hayatta kullanılabilir büyük bir projeye dönüşmesini sağlayan ve bana ATM platform bilgisi edinmeme aracılık eden şirketim Cybersoft, nezdinde müşterisi olan Şekerbank ve huzurlu bir iş ortamında güzel işler çıkardığımız ATM modülü çalışma arkadaşlarıma,

ve en önemlisi beni bu günlere getiren ve eğitim hayatım boyunca bana her koşulda tam destek veren sevgili aileme en içten dileklerle teşekkür ederim.

# İÇİNDEKİLER

	<b><u>Sayfa</u></b>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
SİMGELER ve KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
TABLolar DİZİNİ	xiii
1. GİRİŞ	1
2. BANKACILIK SEKTÖRÜNDE ATM SİSTEMLERİ	3
2.1. ATM Cihazlarının Gelişimi ve Mevcut Durum	3
2.2. Kullanıcı Dostu ATM Sistemlerinin Önemi	5
2.3. Durum Diyagramı ve Tablosu Yaklaşımları	6
2.4. NDC Protokolü	9
3. KULLANILABİLİRLİK VE KULLANICI ARAYÜZ YÖNETİM SİSTEMİ	11
3.1. Nesne Yönelimli Arayüz Yönetim Sistemi	12
3.2. Arayüz Tasarımında Performans Ölçümü	15
3.3. Kullanıcı Merkezli Arayüz Tasarımı	17
4. HİPOTEZ	21
5. ATM ARAYÜZ YÖNETİM SİSTEMİ	22
5.1. Sistem Modelleri	23
5.2. Ekran Tasarımı	25
5.3. Durum Tasarımı	30
5.4. Çıktı Tasarımı	36
5.5. Senaryo Tasarımı	39
5.6. Yöntem ve Sistem Mimarisi	47
5.7. ATM Arayüz Yönetim Sistemi Değerlendirme	49
6. ATM ARAYÜZ PERFORMANSI	59



6.1. ATM Arayüz Performans Metriđi ve Analizi	60
6.1.1. Ortalama Senaryo Bileşen Sayıları Tespiti	60
6.1.2. Tekrar Eden Senaryo Tespiti	64
6.1.3. Senaryo Tamamlanma Süreleri Tespiti	66
6.1.4. Senaryo Başarılı – Başarısız Tamamlanma Oranları	69
6.1.5. Zaman Aşımına Uđrayan Senaryo Tespiti	70
6.2. İşlem Testi	71
7. KULLANICI MERKEZLİ ATM İŞLEM ÖLÇÜMÜ	73
7.1. Kullanıcı Grubu Tespiti	74
7.2. Kullanıcı Davranışları Analizi ve Arayüzde Kişiselleştirme	74
7.3. Kullanıcı Memnuniyeti Anketi ve Deđerlendirme	78
8. SONUÇLAR	82
9. GELECEK ÇALIŞMALAR	84
KAYNAKLAR	85
ÖZGEÇMİŞ	88
EKLER	89

## SİMGELER ve KISALTMALAR DİZİNİ

### Simgeler ve Açıklamalar

#### Kısaltmalar

ATM	:	Automated Teller Machine
BDDK	:	Bankacılık Düzenleme ve Denetleme Kurumu
EFT	:	Elektronik Fon Transferi
HTML	:	Hyper Text Markup Language
IBAN	:	International Bank Account Number
ISO	:	International Organization for Standardization
NDC	:	NCR Direct Connect Protocol
SAN-TEZ	:	Sanayi Tezleri Programı
TCP	:	Transmission Control Protocol
TL	:	Türk Lirası
UML	:	Unified Modeling Language

## ŞEKİLLER DİZİNİ

<u>Şekil No:</u>	<u>Sayfa</u>
2.1: Yaygın kullanılan ATM arayüzü (Diebold).	4
2.2: Sık görülen bir ATM cihazı kuyruğu.	5
2.3: Örnek ATM durum diyagramı.	7
2.4: Durum değişim tablosu.	8
2.5: İlerleme diyagramı.	9
3.1: Durum-Aktarım diyagramı.	15
3.2: Kullanıcı merkezli dizayn akışı.	19
5.1: ATM arayüz yönetim sistemi kullanıcı modelleri.	24
5.2: NDC protokolü ekran formatı.	25
5.3: ATM üzerinde görülen örnek ekran görseli.	27
5.4: Matrix statik ekran tasarımı arayüzü.	28
5.5: Matrix parametrik ekran tasarımı arayüzü.	28
5.6: Ekran tasarım aracı arayüzü.	29
5.7: Günümüzde örnek bir durum tasarımı (Power Point).	32
5.8: Örnek bir durum matrisi.	33
5.9: Matrix durum tasarımı arayüzü.	34
5.10: Durum tasarım aracı arayüzü.	35
5.11: ATM'den alınan örnek çıktı görüntüsü.	37
5.12: Matrix çıktı tasarımı arayüzü.	38
5.13: Çıktı tasarım aracı arayüzü.	39
5.14: Örnek bir ATM akış diyagramı.	41
5.15: Matrix işlem tanımlama arayüzü.	42
5.16: Senaryo tasarım aracı arayüzü.	43
5.17: Örnek bakiye sorma senaryo tasarımı.	46
5.18: Veri akış diyagramı.	48
5.19: ATM sistem mimarisi.	49
5.20: Arka plana resim koyma sonuç grafiği.	52
5.21: Ekran özellikleri tanımlama sonuç grafiği.	52

5.22:	Ekran buton ekleme sonuç grafiđi.	53
5.23:	Ekran statik yazı ekleme sonuç grafiđi.	53
5.24:	Ekran parametrik yazı ekleme sonuç grafiđi.	54
5.25:	Ekran veri giriři ekleme sonuç grafiđi.	55
5.26:	Ekran arama sonuç grafiđi.	55
5.27:	Ekran deđiřikliđi sonuç grafiđi.	56
6.1:	Hesap listeleme tasarımı.	66
6.2:	İřlem test ekranı.	72
7.1:	Mevduat hesabı ve kredi kartı hesabı kullanıcısı ana menüleri.	75
7.2:	Kullanıcı memnuniyet anketi sonuç grafiđi.	80

## TABLolar DİZİNİ

<b><u>Tablo No:</u></b>	<b><u>Sayfa</u></b>
5.1 : NDC protokolü durum tipleri (A-Z).	30
5.2 : NDC protokolü durum tipleri (Diđer).	31
5.3 : Deney alıřmasına katılanların kiři bilgileri.	50
5.4 : Deneyin birinci ařama sonuları – 1.	51
5.5 : Deneyin birinci ařama sonuları – 2.	51
5.6 : Deneyin ikinci ařama sonuları – 1.	56
5.7 : Deneyin ikinci ařama sonuları – 2.	56
6.1 : Senaryo bileřen sayıları (deęiřiklik öncesi).	61
6.2 : Senaryo bileřen sayıları (deęiřiklik sonrası).	63
6.3 : Tekrar eden senaryo sayıları.	64
6.4 : Senaryo tamamlanma süreleri (deęiřiklik öncesi).	67
6.5 : Senaryo tamamlanma süreleri (deęiřiklik sonrası).	68
7.1 : Kullanıcı gruplarına göre iřlem sırası.	75
7.2 : Kullanıcı memnuniyet anketi sonuları.	79

# 1. GİRİŞ

Günümüzde artan servis ihtiyaçları ile birlikte kendi kendine bankacılık işlemlerinin oranının artması, bankaların gişe personeli sayısında indirim gitme istekleri, daha çok servisin ATM cihazlarına aktarılmasını gerektirmekte ve dolayısı ile ATM kullanımını artmaktadır [1]. Sistemin fonksiyonel karmaşıklığının artışı fakat makina donanım arayüzünün aynı kalması nedeniyle, kullanıcı tarafında da problemlerin yaşanmasına sebep olmaktadır. Bazı kullanıcılar için ATM kullanımı olanaksız derecede zor olabilmekte, diğerleri için ise gereksiz yere zaman kaybına yol açan kullanım senaryoları gözlenebilmektedir.

Kullanılmakta olan ATM cihazlarının eski tip yazılımlarla yönetimi, bankacılık sektöründeki servis çeşitliliği, görsel arabirim rekabeti ve diğer değişimler nedeniyle oldukça zorlaşmış ve yavaşlamıştır. Basit bir örnekle yeni kullanılmaya başlayan IBAN (International Bank Account Number) bile mevcut arayüz standartlarıyla ekrana sığdırılamamakta, bunun için standart dışı, yapısal olarak doğru olmayan çözümlere gidilmektedir. Hızla değişen teknolojik gelişmelere rağmen, servislerdeki hızlı artış yeni yöntemlerin uygulanabilirliğini düşürürken, bankaların odak noktalarını ATM'ye yeni fonksiyon katarak rekabet sağlama yönüne kaydırmaktadır. Fakat standart dışı tasarımla katılan her yeni fonksiyon, müşterilere getirdiği karmaşık arayüz sebebiyle de kullanılabilirliği düşürmektedir.

Tez çalışması ile hâlihazırda mevcut olan ATM cihazlarının yeni bir arayüz modeli ile hızlı ve etkin yönetilebilmesini sağlayan yeni bir yönetim sistemi ve arayüz çerçevesi yaratılmıştır. ATM arayüz yönetim sistemi sayesinde nesne yönelimli yaklaşımla, kullanıcı profillerine uygun biçimde değişken, hızlı, dinamik ve kullanıcı dostu arayüz oluşturabilen bir yapı sağlanmıştır. Bu yapı ATM arayüz tasarımında ekran, durum (state), çıktı ve senaryo tasarımı adımlarını ayrı ayrı tasarlamaya olanak sunan bir arayüz yönetim yapısıdır. Tasarımda sağlanan modülerlik yönetimi kolaylaştırmanın yanında, tasarımın birden fazla kullanıcı grubu tarafından yapılabilmesine de olanak sağlamıştır. Aynı zamanda arayüz yönetim sistemi sayesinde tasarımların, bire bir ATM üzerindeki yansımaları, ilk örnekler halinde kullanıcılara daha tasarım aşamasında sunulabilmesi hedeflenmiştir. İlk örneklerin

geliştirme aşamasına kazandıracığı hızın yanında, kullanıcı testleriyle tasarım üzerine analizler yaparak ilerde oluşacak karmaşıklığa da çok önceden engel olunabilecektir.

Günümüzde kullanılan servislerin, yazılım veya sistem performansları mevcut teknolojik yöntemlerle çoğu zaman ölçümlenebilmektedir. Fakat kullanılan pek çok sistemin arayüz performansı ölçümlenememekte ve bu metrik göz ardı edilmektedir. Servislerin performansının yüksek olması kullanan kullanıcılar için her ne kadar önemli olsa da, arayüzlerin karmaşıklığı sistemin performansını doğrudan etkileyerek kullanılabilirliği düşürmektedir. Basit bir örnek ile ATM fonksiyonlarından olan hesap listeleme işlemi, her işlem seçiminden sonra olup sunucuya tekrarlayan işlem yükü getirmektedir. Hâlbuki arayüzde her işlemin başında olmak yerine hesap listeleme işlemi ardından işlem seçimi olması durumunda, sunucuya gelen işlem sayısı yarı yarıya azalacaktır. Bu da bize arayüzün sistem performansı üzerine etkisinin ne kadar büyük olduğunu basit bir göstergesidir. Bu sebeple çalışma kapsamında, ATM arayüzlerinin performans ölçme metriği ortaya atılmış olup, deneysel olarak hesaplanmıştır. Bu metrik sayesinde çıkan sonuçlar ise arayüzün performansını değerlendirmemize olanak sağlamış ve yapılan arayüz tasarım hatalarını ortaya çıkarmıştır.

Bu çalışmayla ATM arayüz tasarımında kişiselleştirmeye olanak sunacak kullanıcı merkezli işlem ölçümü çalışması da yapılmıştır. Buna göre senaryo tasarımında oluşturulan senaryolara göre, sunucuya gelen işlem sayıları deneysel olarak hesaplanmıştır. Hesaplanan sayılar üzerinden yapılan analizlere göre ise kullanıcıların davranış biçimleri üzerine gruplamalar yapılmış olup, farklı gruplara farklı ATM arayüzleri tasarlanmıştır. Arayüzlerin kişiselleşmesinde kullanıcılara; işlemlerini daha hızlı tamamlamalarına katkıda bulunmanın yanında, kendilerini makinaya daha yakın hissetmelerini sağlamak da amaçlanmıştır. Makinayı kullanmaktan korkmayan ve işlemlerini hızlı bir şekilde yaptığını fark eden kullanıcı, bu sayede sistemi kullanma alışkanlığı da kazanacaktır.

Çalışma kapsamında geliştirilen ATM arayüz yönetim sistemi yazılımı, 00794.STZ.2011-1 numaralı SAN-TEZ projesi kapsamında Cybersoft işbirliği ile geliştirilmiş olup, verilen örnekler ve deneysel sonuçlar Cybersoft firmasının müşterisi olan ŞEKERBANK üzerinde test edilerek gerçek ortamda kullanıma sunulmuştur. Ayrıca, geliştirilen ATM arayüz yönetim sistemi ile Şekerbank'ın geçmişte kullandığı sistemin karşılaştırmalı sonuçlarına da tezin alt bölümlerinde yer verilmiştir.

## 2. BANKACILIK SEKTÖRÜNDE ATM SİSTEMLERİ

### 2.1. ATM Cihazlarının Gelişimi ve Mevcut Durum

Gelişen dünyada servis sektöründe artan maliyetler, bu alanda faaliyet gösteren firmaların maliyet düşürme ve ürün kalitesini yükseltme yönünde çözümlere gitmeleri sonucunu doğurmaktadır. Kendi kendine hizmet alınabilen (Self Service) terminal uygulamaları, hizmet sektöründe gerek maliyet düşürülmesini, gerekse servislerin farklı yerlerde farklı zamanlarda kullanılabilmesini sağlamıştır. Bu tür terminaller içinde en yaygın kullanım alanı bulan sistemlerin başında ATM makineleri gelmektedir. İlk kez 1939 yılında (1905 Antep doğumlu) Luther Simjian tarafından patentleri alınan ve ilk örneği bankada (şimdiki Citibank) test edilen ATM gereken ilgiyi görememiş ve uzun bir süre kullanıma girmemiş ve bankalarca kabul görmemiştir [2], [3]. Daha sonra 1966 yılından sonra İngiltere’de (Barclays Bank) ve ABD’de (Chemical Bank) tarafından modern temelleri atılan ve hızla ilerleyen ATM teknolojisi, bankacılık alanında müşterilerin bankacılık işlemlerini kendi kendilerine yapmasını sağlamaktadır [4].

ATM sistemleri, bilişim sistemleri alanında yaygın kullanıma ilk geçenler arasındadır. İlk başlarda yalnızca çek yatırma amacıyla (bir nevi posta kutusu) kullanılan ATM sistemleri sırası ile bakiye öğrenme, para çekme, EFT (Elektronik Fon Transferi) ve diğer işlemlerin yapılması amacıyla kullanılmaya başlamıştır [5]. Teknolojinin gelişmesine paralel olarak ATM cihazlarında ekran ve etkileşim teknolojilerinin geliştiği, güvenlik alanında ise akıllı kart ya da temassız kart gibi yeniliklerin kullanılmaya başlandığı görülmektedir [6].

ATM sistemlerinden verilen servislerin sayı ve özelliklerinin sürekli gelişmesine karşın, günümüzde yaygın olarak kullanılan ATM cihazlarının arayüzlerinin göreceli olarak sabit kaldığı gözlenmektedir. Donanım üretici firmaların 1970 ve 1980’li yıllarda almış oldukları patentlere dayalı olarak tasarlanmış olan 8 tuşlu ekran ve ekran altı nümerik 4x4 klavye modeli, hâlihazırda yaygın olarak geçerli olan ATM kullanıcı arayüzünün temel bileşenidir [7].





Şekil 2.1 : Yaygın kullanılan ATM arayüzü (Diebold).

Bankacılık sistemlerinin Şekil 2.1’de gösterilen arayüzün kullanıma başlamasından sonra yıllar içinde geçirmiş olduğu evrim, mevcut ATM’ler üzerinde yoğun fonksiyon kullanımını gerekli kılmaktadır. Bu çerçevede, ilk yıllarda mainframe bilgisayar terminali temel alınarak tasarlanmış olan ATM sistemleri, daha sonraki yıllarda VM, UNIX, DOS, WindowsNT, OS/2, WindowsXP ve Windows7 işletim sistemleri üzerinde çalıştırılmıştır. Başlangıçta metin tabanlı olan arayüz gereksinimi, zaman içinde grafik nesnelere ve animasyon içerecek biçimde geliştirilme ihtiyacı duymuştur. Öte yandan ATM sistemlerinin yüksek maliyeti ve kullanıcı profiline çok değişken olması, ATM sistemlerinde hızlı ve köklü değişimlerin uygulanmasını güçleştirmektedir.

ATM üreticisi firmalar (IBM, NCR, Diebold vb.) genellikle kendi üretmiş oldukları ATM sistemlerini belirli bir iletişim ve yönetim standardı içinde tasarlamaktadırlar. Bu standart için NCR firmasının NDC protokolü (NCR Direct Connect Protocol) örnek gösterilebilir [8]. Bu protokol sayesinde, ATM sistemlerinin merkez haberleşmeleri, durum değişkenleri ve ekran yönetimleri bir standart dâhilinde yapılmaktadır. Bu protokoller, genellikle mevcut ATM sistem modelleri ile birlikte kullanılmakta ve yukarı doğru yenilenmeleri, hata temizleme haricinde planlanmamaktadır.

ATM servislerindeki sayı ve karmaşıklık artışı ile mevcut sistemlerin geleneksel yollarla yönetiminin mümkün olmadığı, yoğun performans problemine ve destek hizmeti gereksinimine ihtiyaç duyduğu görülmektedir. Kullanıcı menüsüne eklenmesi gereken yeni bir maddenin büyük problemlere yol açabildiği, günlük ya da saatlik olarak sistem menülerinin ya da animasyonların değiştirilmesinin mevcut sistemlerle mümkün olmadığı görülmektedir.

## 2.2. Kullanıcı Dostu ATM Sistemlerinin Önemi

Kullanım alanı gereği ATM sistemleri çok farklı türdeki kullanıcılara hitap etmektedir. Bankalararası Kart Merkezi verilerine göre ülkemizde 2011 itibarı ile yaklaşık 70 milyon ATM Banka kartı, 47 milyon kredi kartı bulunmaktadır [1]. Gerek farklı eğitim düzeyindeki kullanıcıların varlığı, gerekse maaş dönemlerinin özel yapısı nedeniyle çoğu zaman ATM cihazlarının önünde kuyrukların oluştuğu bilinmektedir.



Şekil 2.2 : Sık görülen bir ATM cihazı kuyruğu.

Şekil 2.2’de görülen kuyruk oluşumlarında özellikle ATM cihazının etkin ve verimli kullanımı, gereksiz adımlardan kaçınılması, kullanıcıya özel olarak hazırlanmış menülerin sunulması büyük önem taşımaktadır.

Modern bankacılıkta ATM cihazlarının zaman içinde internet bankacılığının yükselişi ardından önemini kaybedeceği öne sürülmüş ise de yapılan araştırmalarda ATM cihazlarının müşterinin Şube etkileşiminin ardından ilk tercih ettiği etkileşim

biçimi olduğu ve internet bankacılığı tercihinin önünde olduğu ortaya çıkmıştır [9]. BDDK (Bankacılık Düzenleme ve Denetleme Kurumu) kurumunun onaylaması ile birlikte bankaların ATM cihazlarından cep telefonu kredisi yükleme, şans oyunları oynama ve henüz ortaya çıkmamış türlü işlemler yapma olanakları, bu kadar güven duyulan bir sistemin bankalara getiri sağlaması bakımından dikkat çekmektedir.

Yukarıda anlatılan nedenlerle ATM cihazlarının arayüzlerinin kullanıcıların kolay ve hızlı kullanabileceği biçimde, en öne çıkan fonksiyonlarla birlikte müşteriye özel olarak hazırlanması gerekmektedir [10]. Emekliler için İngiltere’de geliştirilmiş olan bir arayüz çözümünde ATM kartını soktukten sonra şifrenin tuşlanması ardından GİRİŞ tuşu yerine hızlı para çek tuşuna basılması sağlanmış ve büyük başarı elde edildiği bildirilmiştir [13]. Özellikle ileri yaşlı kullanıcılar ile genç kullanıcıların, farklı meslek gruplarının ATM cihazı kullanım profilleri oldukça farklı olduğundan bu kullanıcılara yönelik özelleştirilmiş arayüz kullanılmasının işlem hızını artırdığı, müşteri memnuniyetini iyileştirdiği ve farklı servislerden yararlanılmasını sağladığı bilinmektedir.

BDDK kararı ile farklı bankaların ATM cihazlarının paralel kullanımının sağlanması sonucunda mevcut arayüz problemlerine bir yenisi eklenmiştir. Farklı bir bankanın ATM cihazını kullanan diğer banka kullanıcısı, tamamen yeni ve kendisine yabancı bir arayüz ile karşı karşıya kalmaktadır. Bu kullanıcılar istedikleri işlemleri tam olarak gerçekleştirememekte, bazı durumlarda işlem protokolündeki farklılıklar nedeniyle kartlarını makinada unutabilmekte ya da makinada alıkonması ile karşı karşıya kalabilmektedir. Bu problemin çözümü için, tez çalışmasında yer alan sistem ile farklı banka kullanıcılarının kullanıcı profillerine uygun olarak misafir arayüzü oluşturulması da hedeflenmiştir.

### **2.3. Durum Diyagramı ve Tablosu Yaklaşımları**

Yazılım mühendisliği alanında durum diyagramı yaklaşımı, karmaşık ve kritik problemlerin çözümünde tercih edilen bir yöntemdir [14]. ATM cihazları, durum geçiş diyagramı konularında yazılmış birçok yayında örnek problem olarak ele alınmaktadır [15]. Söz konusu durum diyagramlarından biri Şekil 2.3’te örnek olarak verilmektedir.



State Transition Table				
next current	S <sub>1</sub>	S <sub>2</sub>	...	S <sub>m</sub>
S <sub>1</sub>	-	-	...	A <sub>x</sub> /E <sub>i</sub>
S <sub>2</sub>	A <sub>y</sub> /E <sub>j</sub>	-	...	-
...	...	...	...	...
S <sub>m</sub>	-	A <sub>z</sub> /E <sub>k</sub>	...	-

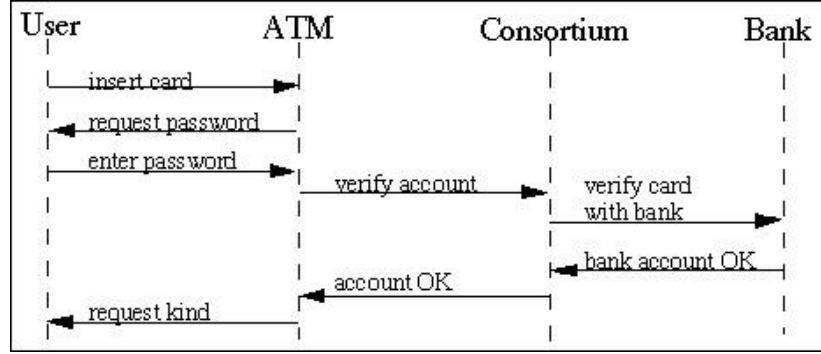
(S: state, E: event, A: action, -: impossible transition)

Şekil 2.4 : Durum deęişim tablosu.

Hâlihazırda kullanılan birçok uygulamada bir adet büyük durum deęişim tablosu bulunmakta, bu durum tablosu yönetici tarafından deęiştirilmek sureti ile yeni ekranlar ve fonksiyonlar eklenip çıkarılabilmekte, sistem yönetilebilmektedir. Çalışma kapsamında birlikte çalışılan firmanın kullanmakta olduęu ATM cihazları yaygın kullanılan NCR üreticisine aittir. Her durum için gösterilmesi planlanan verilerin sisteme önceden girişi yapılır. Daha sonra durum deęişkenlerine baęlı olarak ilgili ekran ve ekran öğeleri gösterilir ve sistem bu biçimde çalışmasını sürdürür. Söz konusu cihazların NDC protokolü çerçevesinde 32x16 çözünürlüęünde bir metin ekran matrisi mevcuttur. Temel olarak NCR cihazlar HTML (Hyper Text Markup Language) gibi tam grafik özellikleri destekliyor görünseler de mevcut durumda NDC protokolü için geliştirilmiş ve kullanılmakta olan yazılım uygulanabilir bir yapıya sahip deęildir. Bu nedenle bankaların kendi HTML uyumlu tasarım ve yönetim yazılımlarını geliştirdikleri gözlenmektedir [17].

Nesneye yönelik yaklaşımların 2000'li yıllarda gelişmesi ile beraber birçok kurumsal ve endüstriyel yazılımda nesneye yönelik programlama dilleri kullanılmaya başlanmıştır. Özellikle UML (Unified Modeling Language) ile tanımlanıp gerek durum diyagramı gerekse Şekil 2.5'de gösterilen ilerleme diyagramı (Sequence diagram) kullanılarak yazılan modern yazılımlarda hem kalite ve performans yükselmiş, hem de

yazılım geliştirme ve yürütme verimliliği artmıştır [17]. ATM sektöründe ise her ne kadar ATM sistemleri nesneye yönelik programlama dokümanlarının vazgeçilmez örneği olsa da, ürünlerin ve kullanılan sistemlerin uzun yıllardır aynı kalması nedeniyle nesneye yönelik yaklaşımlar ve UML ile sistem tasarımının yapıldığı örnekler oldukça azdır.



Şekil 2.5 : İlerleme diyagramı.

## 2.4. NDC Protokolü

Günümüzde ATM üreticisi firmaların genellikle kendi üretmiş oldukları protokolleri kullandığı bilinirken, bunların arasında ilk ATM üreticilerinden olan NCR firmasının NDC protokolünün ise birden fazla firma tarafından da tercih edildiği ve ATM sektöründe standartlaşmış en büyük protokol olduğu bilinmektedir [18]. Tez çalışması kapsamı da bu sebeple NDC protokolü standartlarına göre belirlenmiş olup, proje çıktısı olarak hazırlanan “ATM Arayüz Yönetim Sistemi” de bu protokol standartlarına göre arayüz tasarımları oluşturan bir sistemdir.

NDC protokolü, ATM sistemlerini belirli bir iletişim ve yönetim standartları içinde tasarlanmasını amaçlayan bir protokoldür [19]. Zayıf istemci (thin client) mimarisine göre çalışan bu protokole göre zekâ tamamen sunucu tarafındadır. Sunucu, ATM’ye sadece ne yapması gerektiğini bildirir. ATM’de kullanıcının işlemi gerçekleştirip gerçekleştirilememeye durumlarına göre sunucu tarafında oluşacak akışın bilgisini iletir. Bunun yanında sunucu tarafına; işlem akışı, zaman aşımı, ekranda gösterilecek bilgiler, hangi kartların kabul edileceği, ünite arıza ve kullanım durumları

ve ATM'nin hangi kipte beklemesi gerektiği gibi birçok bilgi aktarımı da NDC mesaj yapısı ile sağlanır.

Literatürde durum makinasının (state machine) en iyi örneklerinden olan ATM, üzerinde oluşturduğu durumları NDC protokolünün durum tasarımı standartlarına göre oluşturur [18]. Buradaki her bir durum NDC işlem (transaction request) mesajı ile sunucuya gelirken, bu duruma karşılık gösterilecek ekran, verilecek çıktı ve geçilecek yeni durum işlem cevap (transaction reply) mesajı ile ATM'ye iletilir. ATM ise bu mesajı almasıyla birlikte mesajın içindeki durum akışına geçerek, kullanıcıdan yeni bir işlem gelene kadar bekleme konumuna geçer. Ayrıca ATM tarafında, mesajın işlendiğini gösteren bir istenen durum (solicited status) mesajı da sunucu tarafına gönderilir. Bu mesaj ile birlikte üç yönlü ve güvenli bir iletişim gerçekleşmiş olur. ATM'ler bu iletişimde üzerindeki ekran veya oluşan durumu kendi içinde saklamazken, tüm bilgilerin sunucuda tutulacağını garanti eder. Bu da NDC protokolünü, ATM gibi dış mekân bir sistem için güvenilir kılar [20].

ATM üzerinde kullanıcıya sunulan ekran, durum ve çıktı tasarımları günümüzde NDC protokolünün standartlarına göre hali hazırda bankaların yazılım birimleri tarafından oluşturulurken, her bir yeni tasarımın hayata geçirilmesi için NDC protokolün tüm standartlarının öğrenilmesi gereklidir. Bu sebeple de protokolü anlayıp, protokolden izole bir arayüz yönetim sistemi olmadan, yönetilebilir ve kullanıcı dostu arayüzler tasarlamak çok mümkün olmayacaktır. NDC protokolünün ekran, durum ve çıktı tasarımlarına getirdiği standartlar beşinci bölümde yer alan ATM Arayüz Yönetim Sistemi'nin tasarım ekranları bölümlerinde detaylı anlatılmıştır.

### 3. KULLANILABİLİRLİK VE KULLANICI ARAYÜZ YÖNETİM SİSTEMİ

İnsan bilgisayar etkileşimi alanının temel konusu olan kullanılabilirlik, ISO (International Organization for Standardization) standartlarında; “Hedef kullanıcıların gerekli görevleri etkili, verimli ve tatmin edici bir şekilde yerine getirebilmeleri” olarak tanımlanmaktadır [22].

İnsanlara ait bir takım görevlerin gün geçtikçe bilgisayarlara aktarılması ile kullanılabilirlik daha da önem kazanmıştır. Görevlerin doğru tamamlanabilmesi, bilgisayarları kullanan kullanıcıların görevleri doğru anlamasından geçer. Sistemin doğru anlaşılmasındaki en temel unsurda kullanıcılara sunulan arayüz tasarımının doğru olmasıdır. Tasarımlardaki karmaşa kullanılabilirliği düşürürken zamanla kullanıcının sistemi kullanmaktan vazgeçmesine de sebep olmaktadır.

ATM sistemlerinde kullanılabilirlik ise; kullanıcının yapacağı işlemi hızlı ve tatmin edici bir şekilde yapabilmesiyle doğru orantılıdır. Bir biri ardına gelen ekranlar arasında kaybolan ve istediği menüye kolayca gelemeyen kullanıcı, işlemini gerçekleştiremeyecek ve ATM’yi kullanmaktan da zamanla vazgeçecektir.

Yapılan araştırmalara göre günümüzde ATM arayüz tasarımlarında simge temelli (icon-based) arayüzlerin, yazı temelli (text-based) arayüzlere göre daha kullanılabilir olduğu görülmektedir. Özellikle okuryazarlığın az olduğu bölgelerde bu daha fazla öneme sahip olmaktadır. Yapılan incelemelere göre simge temelli arayüzlerin tanınma ve öğrenme hızının da diğerlerine göre fazla olduğu görülmektedir. Bunun yanında yazı + simge temelli (icon and text based) tasarımların ise bazı durumlarda simge temelli tasarımlara göre daha kullanılabilir olduğu görülmektedir. Bunda kuşkusuz seçilen simgelerin önemi büyüktür. Kullanıcı tarafından anlaşılmayan simgelerin yazı ile desteklenmesi kullanılabilirliği arttıran temel faktör olarak karşımıza çıkmaktadır. Simgelerin farklı kullanıcılarda farklı algılar yaratabilir olması, yazı ile desteklenmenin gerekliliğini ifade etmektedir. Bu da bize tasarımlarda kullanılan simge seçimlerinin önemini daha açık göstermektedir [23].

Günümüzde ATM arayüz tasarımında kullanıcılar düşünülürken çoğu zaman engelli vatandaşlar göz önünde bulundurulmamaktadır. Sadece ekran görselinden



yapılan seçimlerle kullanılan bir ATM arayüzü, görme engelli kullanıcılar için ATM'nin kullanılmamasına sebep olacaktır. Bu sebeple bu tip kullanıcılar için de tasarımların ses, parmak okuyucu, kabartmalı ekran görseli, çağrı merkezi hizmeti gibi arayüzler tasarlanması, kullanıcı çeşitliliğini ve dolayısıyla da kullanılabilirliği arttıracaktır [24], [25].

### 3.1. Nesne Yönelimli Arayüz Yönetim Sistemi

Kullanıcı arayüzü oluşturmaya yarayan sistemler olarak karşımıza çıkan arayüz yönetim sistemleri (UIMS: User Interface Management System) literatürde şu şekilde tanımlanmaktadır [26];

*“Kullanıcı arayüzü oluşturmak için özel araç ve teknikler içeren sistemlerdir.”*

*“Uygulama çerçevesinde kullanıcı arayüzü tasarımını gerçekleştiren ve kullanıcı ile etkileşimlerini yöneten sistemlerdir.”*

Bu sistemler yazılımsal gereksinimleri en etkin bir şekilde temsil edecek arayüzler tasarlamaya yardımcı olan, yazılıma ve platforma bağlı kalmaksızın kullanılabilen sistemler olarak karşımıza çıkar. Kullanıcı arayüz yönetim sistemleri sayesinde ayrıca arayüz tasarım kuralları kolayca uygulanabilirken, arayüzde bir standartta sağlanmış olunur.

Nesne yönelimli tasarım ise, problemin çözümünde kullanılan nesne modelinin farklı bir bakış açısıyla arayüz tasarımında kullanılmasıdır. Modern nesne yönelimli çözümler ve tasarım yöntemlerinin çoğu; gereksinim, tasarım, uygulama, test ve dağıtım süreçleri boyunca kullanım senaryoları üzerine kuruludur. Kullanım senaryoları, nesne yönelimli programlama ile birlikte ortaya çıkmış olmasına rağmen yordamsal programlanan sistemlerde de işe yaramaktadır [27].

Nesne yönelimli tasarım, nesne yönelimli çözümlemenin çıktısı olan kavramsal modeli alır ve tasarımın temeli olarak seçilen mimari kararlarla birlikte ortamdan, programlama dilinden ve seçilen araçlardan gelen uygulama kısıtlamalarını bu model üzerine uygular. Kavramsal modeldeki kavramlar, -niteliklerine göre- sınıflara,

uygulamaların soyut arayüzlerine ve nesnelerin değişik durumlarda gerçekleştirecekleri rollere yansıtılır [27].

Nesne yönelimli çözümlerdeki değişmez kavramlar; tekrar kullanılabilir arayüzleri ve uygulamaları oluştururken, değişken kavramlar; karar veren, ortama özgü veya duruma özgü mantık veya algoritmaları uygulayan sınıfları oluşturur. Nesne yönelimli tasarım sonucunda, nesnelere kullanarak sistemin nasıl yapılandırılacağına detaylı bir tanım verilir.

Nesne yönelimli arayüz yönetim sistemi ise; bir sistemin işlem gereksinimlerinin çözümünde kullanılan tasarımı, platformdan bağımsız nesne yönelimli tasarım ile tasarlamaya olanak sunan arayüz yönetim sistemi olarak tanımlanabilir.

Nesne yönelimli arayüz sisteminde, kullanıcıya sunulan tüm bileşenler bir sınıf (class) yapısındadır. Her bileşenin yani sınıfın bir niteliği (attribute) olduğu gibi, bir ilişkisi (relationship) ve bir fonksiyonu (method) da sistemde tanımlanır[29]. Bu sayede bileşenler tasarıma kendi özellikleri dâhilinde görsellik katarken, tasarıma da standart getirmiş olur. Örneğin;

- Sınıf (Class): Grafikselle bir obje olan buton olsun.
- Davranışı: Rengi, yazı tipi, görünürlüğü, konumu.
- İlişkisi: Diğer butonlar veya diğer grafikselle objelerin listesi.
- Fonksiyonu: Görünür veya görünmez olma, konum değiştirme, renk değiştirme.

Nesne yönelimli arayüz yönetim sisteminin avantajları;

- Tutarlılık

İçerdiği tasarım kuralları sayesinde her ne kadar tasarımda esnek olabilsede arayüzde bir standart oluşturur. Bu standartlarda arayüzde tutarlılığı sağlar.

- En acemiden en uzmana kadar tüm kullanıcı tiplerine uygunluk

Tasarım yapmak için yazılımsal bilgi birikimi gerektirmeden her tipte kullanıcı için yönetimin sağlanabilmesi.

- Kolay arayüz tasarımı ve deęiřtirme

Saęladıęı modülerlik ve yönetim kolaylıęı sayesinde arayüz üzerinde deęiřiklikler ve yeni tasarımlar hızlı ve kolaydır.

- Hata ayıklama ve kurtarma desteęi

Arayüzdeki hatalar yazılımdan baęımsız tespit edilebilir ve geri dönüş senaryoları kolayca uygulanabilir.

- Yazılımdan ve karmařık yapıdan izole

Odak noktasını yazılıma vermeden kullanıcı davranıřlarına yoğunlařtırarak daha iyi arayüzler tasarlamaya olanak sunar.

- Anında görebilme ve test etme

Tasarımsal geri bildirim çok hızlı. Bu sebeple yapılan tasarım anında görülebildięinden geliřtirmek kolaylařır. Ayrıca özellikle bilgisayar ortamından baęımsız sistem tasarımları için, benzetim ortamı oluřturduęundan tasarımlara zaman kazancı saęlar. Örnek; ATM.

Nesne yönelimli arayüz yönetim sisteminin dezavantajları;

- Kullanması öğrenme gerektiriyor

Normal sistem dıřında birde arayüz yönetim sistemin öğrenilmesi gerekiyor. Bu sebeple küçük sistemler için ekstra zaman kaybı.

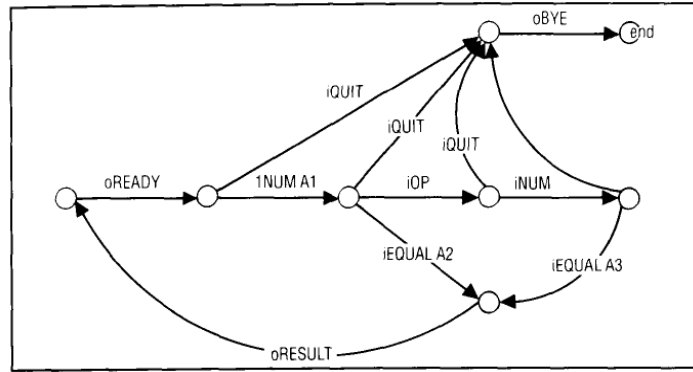
- Yaratılabilecek arayüzler sınırlı

Sadece arayüz yönetim sistemi modelleri üzerinden tasarım mümkün. Bu da tasarıma bir sınır getiriyor. (Fakat bu bazı gerekli durumlar için tutarlılık saęlayacaęından avantaj olmaktadır.)

- Geniř bir kullanıma sahip deęil

Sadece, odak noktasındaki sistem tasarımı üzerine kurulu bir yapı. Bu sebeple küçük sistemler için ekstra maliyet gerektirdiğinden geniş bir kullanım alanına sahip değil.

Nesne yönelimli kullanıcı arayüz yönetim sistemleri özellikle girdi-olay (input-event) ilişkisi içeren sistemler için oldukça gereklidir. Çünkü kullanıcı her gördüğü durum için bir girdi oluşturabilmekte, sisteminde bu girdilerin her biri için bir olay davranışı sergilemek zorundadır. Çok fazla işlem gerektiren büyük sistemlerde bu ilişkiyi içeren bir sistemin tasarımını, yönetim sistemi olmadan yapmak maliyet ve zaman açısından da oldukça büyüktür. Literatürde bu tarz yapıları şematize etmek için Şekil 3.1'de görülen durum-aktarım diyagramları (state-transmission diagrams) çizilmektedir. Çok fazla işlem gerektiren sistemler için bu diyagram sayfalar halinde olabilmekte ve bu karmaşık yapıdan her hangi bir tasarım ürünü çıkarmak oldukça zorlaşmaktadır. Bu sebeple de büyük sistemler için bir nesne yönelimli kullanıcı arayüz yönetim sistemine ihtiyaç vardır.



Şekil 3.1 : Durum-Aktarım diyagramı.

### 3.2. Arayüz Tasarımında Performans Ölçümü

Günümüzde artan teknolojik gelişmeler her ne kadar bilgisayarların donanımsal performanslarını arttırsa da, yoğun işlem gören sistemler için sadece donanımsal performansın artması çoğu zaman yeterli olmamaktadır. Bu tarz sistemleri kullanan firmaların donanımsal yeterliliği arttırmanın yanında, bu sistemler üzerinde çalışan

yazılımlara da performans kıstasları uygulayarak işlem hacmini arttırma yoluna gittikleri gözlemlenmektedir.

Günümüzde çoğu sistem için yazılımsal ve donanımsal performans kıstasları daha sistem kurulmadan oluşturulmaktadır. Sistem gerçek hayatta çalışmaya başlamadan bu kıstasları sağlayıp sağlamadığı kontrol edilmekte ve çeşitli testlerden başarısız olan durumlar üzerine analizler yapıp sistem modifikasyonları yapıldığı bilinmektedir. Sistem performans testleri genelde yazılımsal hataları bulmayı amaçlamaz ancak sistemdeki dar boğazları ortaya çıkardığından, işlem hacmi büyük sistemlerde hayati önem taşımaktadırlar. Kullanıcı merkezli olmayan sistem performans testlerinde bilinen ve en yaygın kullanılan testler; yük testi, yükleme (load) testi, stres testi gibi örneklerdir [30].

Donanımsal ve yazılımsal performansın yeterli olduğu, fakat işlem hacminin çok yüksek olması sebebiyle hala yeterli performans kabulüne ulaşamayan sistemler için ise en büyük engelin arayüz tasarımındaki problemler olduğu görülmektedir. Buna sebep olan temel faktörlerden biri kuşkusuz tasarım prensiplerine uymayan tasarımların kullanılabilirliği düşürmesidir [31]. Günümüzde kullanıcı merkezli testler genelde; alfa testi, beta testi ve kullanıcı kabul testi şeklinde üç gruptan oluşur. Fakat bu testler sadece, kullanıcının istediği fonksiyonların arayüzde sağlanıp sağlanmadığı üzerinedir ve arayüzdeki tasarım problemlerini ortaya çıkarmamaktadır. Arayüz tasarımındaki problemler her ne kadar sistem özelliklerinin arttırılması ile çözülüyor gibi dursa da aslında tasarımdaki problemlerin giderilmesi ile çok daha az sistem gereksinimi ile daha büyük işlem hacmine ulaşılacağı öngörülebilir.

Peki, arayüz tasarımının performansı nasıl ölçülebilir? Bu soruya tüm sistemler için ortak bir kıstasla cevap vermek oldukça zordur. Fakat genelleme yaparak şu sorulara cevap verecek bir arayüz performans kıstası bütün sistemler için oluşturulabilir;

- Ortalama bir ana işlem ardı ardına kaç işlem ekranından oluşuyor?
- Bir alt ekran kaç farklı işlem ekranında kullanılıyor?
- Tekrar eden işlemler, üst bir ana işlemde birleştirilebilir mi?
- Kullanıcı genelde hangi senaryoları başarılı hangi senaryoları başarısız sonuçlandırmış?

- Kullanıcının en çok işlem yapmakta beklediği ekran veya işlem hangisi?
- Kullanıcı ortalama yapacağı bir işlemi ne kadar sürede tamamlamış?
- Başarısız sonuçlanan işlemler ortak mı? Kullanıcı neden başarısız sonuçlandırmış?

Tüm sistemler için bu sorulara verilecek cevaplar aslında sistemin arayüz performans metriğini ortaya çıkaracaktır. Bu metrik sayesinde ölçümlenecek değerler üzerine yapılacak analizlerle de arayüz tasarımındaki hatalar ortaya çıkarılacaktır. Tasarım hataların giderilmesi, sistemin kullanılabilirliğini de olumlu yönde etkileyecektir. Çünkü kullanıcının arayüzde aradığını bulamaması ya da bir işlemi tasarım yanlışlığı yüzünden çok uzun sürede yapması o sistemi kullanmayı bırakmasına sebep olacaktır. Ayrıca işlem hacmi yoğun sistemler için sistem gereksinimlerini donanımsal olarak arttırmak yerine tasarımdaki hataları gidermek çok daha düşük maliyetlidir. Yapılacak analiz sonrası arayüz tasarımındaki hataların giderilmesi ile mevcut sisteme gereksiz yük binmesi engellenecek ve sistemin işlem kapasitesinin artırılması hedeflenecektir.

Tez kapsamında; ATM arayüz performans metrikleri, gerçekleştirilen ATM Arayüz yönetim sistemi üzerine sorulacak sorular ve yapılan analizler altıncı bölümde detaylı anlatılmıştır.

### **3.3. Kullanıcı Merkezli Arayüz Tasarımı**

Günümüz teknolojisinde artan teknolojik yatırımlar ile birlikte birden fazla fonksiyonu içinde barındıran sistemler artık hayatımızın bir parçası olmaktadır [32]. Televizyon üzerinden internete girmek, cep telefonundan akıllı ev sistemlerini yönetmek, araba konsolu üzerinden mesajlaşmak vb. gibi daha birçok teknolojik ürün bir biri ile bütünleşmiş hale gelmiştir. Artan fonksiyonlar kullanıcılara yeni özellikler getirmenin yanında, öncelerde kolayca kullandığımız sistemlerin artık kullanımı daha karmaşık birer ürün haline gelmesi kullanıcı çeşitliliğini düşürmektedir [33]. Bunda en güzel örneğin akıllı telefonlarda olduğunu söylemek sanırım yanlış olmaz. Daha önceden sadece arama ve mesaj atma özelliğine sahip telefonlar 7-70 yaş grubunda

çoğu kullanıcıya hitap ederken, şu anda akıllı telefonlar 50 yaş üzeri kullanıcılar tarafından çoğunlukla tercih edilmemekte ya da sadece arama özelliği kullanılmaktadır. 2013 yılında yapılan bir ankete göre Türkiye’de akıllı telefon kullanım oranları; 18-40 yaş arası %84 iken, 50 yaş üstünde bu oran %2,4’dür [34].

Birden fazla fonksiyonu içeren karmaşık sistemlerin kullanımını etkileyen en temel faktörlerden biri de o sistemin arayüzüdür. Artan fonksiyonların arayüzü karmaşık bir yapı haline getirmesi, kullanıcı çeşitliliğinin düşmesine ve sistemin sadece belli bir kullanıcı kitlesine hitap etmesine sebep olmaktadır. Bu sebeple arayüz tasarımı yapılırken sistemin kullanıcı kitlesi de düşünülmeli ve farklı bilgi ve eğitim seviyesindeki tüm kullanıcılara hitap edecek arayüzler tasarlanmalıdır.

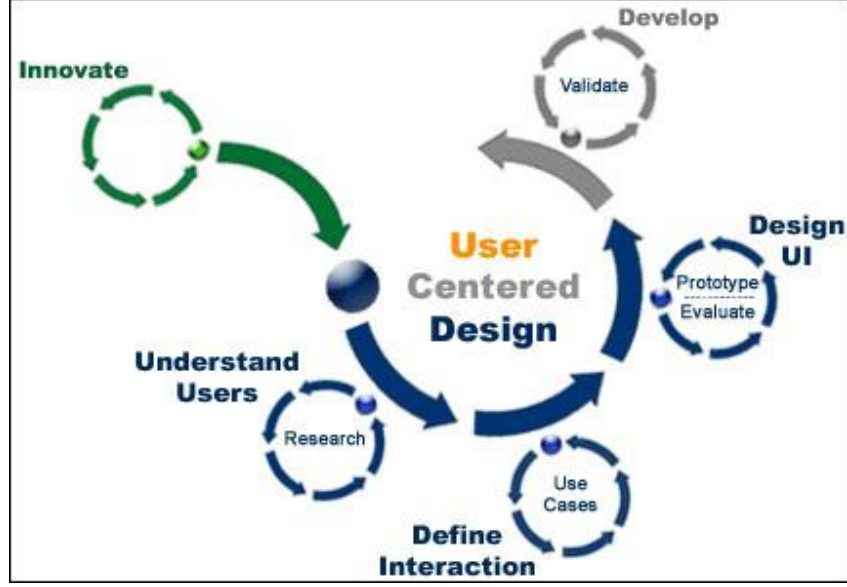
Herkes tarafından kolay kullanılabilen, kullanıcı dostu arayüz tasarlamak bilindiği gibi oldukça zordur. Ayrıca bazen ortaya çıkarılan sistemlerin kullanıcı grupları daha tasarım aşamasında tahmin edilememekte ve daha sonradan ortaya çıkmaktadır. Bu sebeple geniş kitlelere hitap eden arayüz tasarlamak için; kullanıcı merkezli tasarım ilkelerini kullanmak daha doğru bir seçim olabilir.

Kullanıcı merkezli tasarım 1980 yılında California San Diego üniversitesinde Donald Norman Laboratuvarı’nda, İnsan Bilgisayar Etkileşimi alanında yeni bir bakış açısı olarak ortaya atılmıştır [37]. Geliştirdiği bu bakış açısını 1988’de yazdığı The Psychology Of Everyday Things (POET) kitabında şu şekilde özetlemektedir;

- Sistemin kavramsal modelini, alternatif hareketleri ve bu hareketlerin sonuçlarını içerecek görünürlükte olmalı.
- Nesnelere her an değiştirilebilir ve saptanabilir kolaylıkta olmalı.
- Sistemin mevcut durumunu her an değerlendirebilecek kolaylıkta olmalı.
- Gereksinimler ve amaç arasında doğal eşleşme gözlemlenmeli. Bunlar sistemin anlık durumları ve o durumların sonuç etkileri arasındaki farkı verir.

Kullanıcı merkezli tasarımı bir başka şekilde; kullanıcının, tasarımın odağında olması olarak da tanımlayabiliriz. Kullanıcılar tasarımın en başından itibaren sürekli göz önünde bulundurulur ve onlardan gelen geri bildirimler sayesinde ortaya çıkan sistemin en verimli hale getirilmesi amaçlanır.

Kullanıcı merkezli tasarımın yaşam döngüsü Şekil 3.2’de görüldüğü üzere, kullanıcının yaşam döngüsünün merkezinde yaşaması ve ilk örnek, tasarım, geliştirme, değerlendirme ve değişiklik adımlarının hepsinde kullanıcının hesaba katılması anlamını taşımaktadır [38].



Şekil 3.2 : Kullanıcı merkezli dizayn akışı.

Kullanıcı merkezli tasarımda örnek olarak şu sorulara cevap bulma amaçlarını [39];

- Sitemizin kullanıcıları veya kullanıcı grupları kimler?
- Bu kullanıcıların amaçları neler?
- Kullanıcıların sistem veya benzer sistemler hakkındaki bilgi ve deneyimleri ne seviyede?
- Kullanıcıların ihtiyaç duydukları fonksiyonlar neler? Bu fonksiyonlar gereksinimlerle örtüşüyor mu?
- Kullanıcılar hangi bilgilere ihtiyaç duyuyorlar ve bu bilgiyi nasıl bir formatta istiyorlar?
- Kullanıcılar sistemin nasıl işlemesi gerektiği hakkında ne düşünüyorlar?



Kullanıcı merkezli tasarım, ilke olarak odak noktasına kullanıcıyı alarak, daha kullanılabilir arayüzler tasarlamayı hedeflese de; yaşam döngüsünde kullanıcı geri bildirimleri ile “arayüzde kişiselleştirme” hedeflerine de uygun bir tasarım biçimi olabileceği görülmektedir. Buna göre farklı eğitim ve deneyime sahip kullanıcı gruplarının hepsine hitap eden bir arayüz tasarlama fikri, hem tasarım yaşam döngüsünü uzatmakta hem de ortaya çıkan tasarımın karmaşıklığını arttırmaktadır. Bu sebeple özellikle kullanıcı grupları belirli sistemler için hedef gruplara uygun farklı arayüzler tasarlamak, doğru ve daha verimli sonuçlar doğuracaktır. Tez kapsamında ATM arayüzlerinde yapılan analizler sonucu çıkarılan kullanıcı grupları ve bu gruplara özgü ortaya çıkarılan kişiselleştirilmiş arayüzler yedinci bölümde detaylı anlatılmıştır.

Günümüzde çok fonksiyonlu sistemlerin özellikle üst yaş grupları tarafından kullanılmamasının en büyük sebebinin, fonksiyonların arayüzü çok fazla karmaşık hale getirmesi olduğundan bahsetmişim [33]. Hâlbuki basit mantıkla yaş seviyelerine göre gruplanan kullanıcılar için farklı arayüz tasarımı düşünüldüğünde, üst yaş grubu kullanıcıların ihtiyaçlarına uygun bir arayüz tasarımının kullanılabilirliği arttıracığı söylenebilir. Arayüzde kişiselleştirme, kullanıcıları sistemin ihtiyaçlarına göre; yaş gruplarına, cinsiyete, bilgi ve deneyim seviyelerine, kullandığı işlem tiplerine, en sık yapılan işlemlere, üye veya misafir olma durumlarına göre gruplara ayırabilir. Bu gruplara göre de farklılaşan arayüzlerle sistem kullanılabilirliğinin artması hedeflenir.

İnsan makine etkileşiminde kullanılabilirliğin temel ilkeleri açısından arayüzde kişiselleştirmeyi inceleyecek olursak; farklı arayüzlerin farklı kullanıcı gruplarına sağladığı kolaylıkla, kullanıcının istediği işi daha kısa sürede yapacağını ve aradığını daha kolay bulacağını düşünürsek “etkin” ve kendi ihtiyaçlarına uygun tasarlanmış, kendine ait hissedeceği bir arayüzle de “memnun edici” olacağı söylenebilir.

## 4. HİPOTEZ

Günümüzde artan servis ihtiyacının tasarımcılara getirdiği zorluklar yüzünden istenilen fonksiyonların hızlı bir şekilde sağlanamadığından ve bununla beraber tasarımlardaki problemlerin arayüzlere getirdiği karmaşa yüzünden sistemin kullanılabilirliğinin de ne denli etkilendiğinden bahsetmiştik. Bu problemler düşüldüğünde tez kapsamında ATM cihazları kullanıcıları için kullanılabilirliği direk etkilediğini düşündüğümüz arayüzlerin tasarımında kullanılacak bir “Nesne Yönelimli ve Kullanıcı Dostu ATM Arayüz Yönetim Sistemi” çalışması yapılmıştır.

Bu çalışmayla ile birlikte ATM cihazlarında yeni nesil işlemlerin rahatlıkla tanımlanabileceği, banka personelleri tarafından en az hata ile yönetilebilen, arayüz değişimlerinde kullanım sorunlarını en aza indiren ve kullanıcı tarafından kolay kullanılabilen bir ATM arayüz yönetim sistemi geliştirilmiştir. Geliştirilen sistemin arayüz tasarımına getirdiği modülerlik ve yönetilebilirlik sayesinde arayüz tasarımları, daha hızlı ve tasarım kaynaklı performans sorunu yaratmayacak (yaratsa da geri bildirim kolay olacak) kalitede gerçekleştirilebilecektir.

Ayrıca günümüzde yazılım veya donanım performansları bir takım bilinen testler ile ölçümlenebilirken, kullanılabilirliğe doğrudan etkisi olan arayüz tasarım performansının ölçümlenmeleri genellikle yapılamamaktadır. Bu sebeple, ATM arayüz yönetim sistemi sayesinde ölçülebilir ATM arayüz performansı metrikleri ortaya çıkarılmıştır. Bu metriklerle ATM'nin arayüz performansının deneysel ölçümlenmeleri yapılmış ve iddia edildiği üzere tasarımdaki hatalar daha net ortaya çıkarılmıştır.

Arayüz kullanılabilirliğinin en temel faktörlerinden biri de kullanıcı memnuniyetidir. Arayüzü kendine ait hisseden kullanıcılar için sistem kullanımı artmakta, fakat arayüz ile kullanıcının arasındaki mesafe arttıkça kullanım oranı da düşmektedir [42]. Bu sebeple ATM arayüz yönetim sisteminin sağladığı tekrar tekrar kullanılabilir senaryo tasarımı sayesinde arayüzlerde kişiselleştirme yapılabilecektir. Oluşturulacak kullanıcı gruplarına göre tasarlanabilecek farklı arayüzler ile de işlem hızının, sistem verimliliğinin ve kullanıcı memnuniyetinin artması hedeflenmiştir.

## 5. ATM ARAYÜZ YÖNETİM SİSTEMİ

Bankacılık sektöründe hızlı değişen pazar ihtiyaçlarına paralel olarak artan servis ihtiyaçlarına karşın, hizmet sağlayan servislerin arayüzlerin sabit kaldığı görülmektedir. Bunda en büyük sebep sektördeki işlem hacmi ve sistemlerin oldukça büyük, geliştirmesi ve yönetilmesinin zor olmasıdır. Sektördeki artan rekabetin getirdiği talep ile birlikte, değişime ayak uydurmayarak aynı kalan arayüzlerin olduğu sistemler, pazarda değerini kaybetmekte ve kullanıcılar tarafından tercih edilmez hale gelmektedir. Bu sebeple de değişime ayak uydurmak, yenilikleri sisteme kolay adapte ederek geliştirmek önemlidir. Fakat hızla değişen pazarda ATM gibi kısıtlı ve kullanıcı etkileşimi sınırlı cihazların arayüzlerini geliştirmek nasıl hızlı olacaktır? Bu sorunun cevabı kuşkusuz iyi bir arayüz tasarımı geliştirme ve yönetim ortamı sayesinde olmaktadır. Büyük sistemlerdeki yazılım karmaşasından uzak sadece tasarıma odaklı bir sistemin, arayüz gelişimine sağladığı verim oldukça fazla olurken, nesneye dayalı yaklaşım ile arayüzler modüler ve tekrar kullanılabilir olacaktır.

ATM sistemlerinin çoğu günümüzde, NDC protokolünün getirdiği standartlar dâhilinde oluşturulan arayüzler ile yönetilmektedir. Bu arayüzler, yine tasarım şablonu olarak ATM kurulumunda yüklenen durum(state) tablosunun kullanıcıya yansımalarından oluşur [18]. Mevcut yapıya eklenecek bir ekranın bile tasarlanması için bu protokolün tamamının öğrenilmesi ve yapılacak ekranın hem durumsal hem de görsel olarak ATM üzerinde test edilmesi gerekmektedir. Arayüz yönetim ve geliştirme sisteminin olmadığı düşünüldüğünde, bu süreç bir ekran için bile günler süren uzun bir süreç olmaktadır. Bu da arayüzün gelişiminin ve pazarın ihtiyaçlarına karşılık vermesinin önünde büyük engel teşkil etmektedir.

Tez kapsamında yapılan çalışmada oluşturulan ATM arayüz yönetim ve geliştirme sistemi en temelde tasarımdaki zorlukları gidererek, kullanıcı dostu tasarımlar oluşturabilmeyi amaçlamıştır. Bu sebeple de öncelikle NDC protokolünün getirdiği standartları öğrenme ve bunu sisteme adapte etme süreci geçirilmiştir. NDC protokolü standartlarının tasarımı ne kadar kısıtladığına ve zorlaştırdığına, sistemin tasarım ekranlarını anlatan alt bölümlerde detaylı bir şekilde yer verilmiştir. Fakat sistemin genel olarak sağladığı avantajları maddeler halinde belirtecek olursak;

- ATM arayüz geliştirmeyi kolaylaştırma ve yeni taleplere hızlı cevap verebilme.
- Her birimden kullanıcı için ortak bir platform oluşturma.
- Tasarımı anında görme, test etme ve geri bildirim kolaylığı sağlama.
- Kod karmaşasından uzak, sistemden izole arayüz üzerine odaklanabilme.
- Standartlar oluşturarak arayüzde uyumluluğu sağlama.
- Hızlı değişen çalışan profiline karşın öğrenme ve adaptasyon kolaylığı ile her tipte kullanıcıya uygunluk.
- Nesne yönelimli ve sürükle bırak tasarım ile tasarımı tekrar tekrar kullanılabilme.
- Ölçümlenebilir performans metriği ile arayüz performansını ölçme ve bu sayede tasarımındaki yanlışları tespit edebilme.
- Farklı kullanıcı tiplerine ayrı arayüzler tasarlanmasının kolaylığı ve bu sayede arayüzde kişiselleştirmeye olanak sağlama.

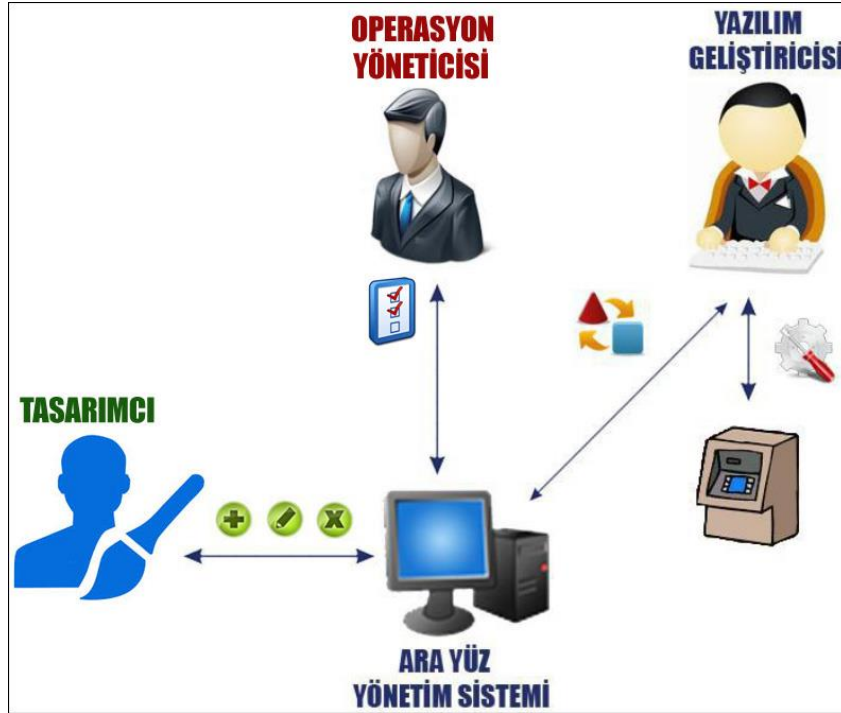
ATM arayüz yönetim sistemi; ATM kullanıcısı ile etkileşimde kullanılan ekran ve makbuz tasarımı ve kullanıcıdan alınan aksiyonlara verilecek çıktıları yöneten durum ve senaryo tasarımı olmak üzere dört farklı tasarım ekranından oluşmaktadır.

Tez kapsamında gerçekleştirilen ATM arayüz yönetim sistemi yazılımının SAN-TEZ kapsamında çalışılan banka olan ŞEKERBANK tarafından kullanıma alındığından bahsetmiştim. Bankanın, gerçekleştirdiğimiz sistemi kullanmaya başlamadan önce kullanmakta olduğu ve yönetilebilme problemleri yüzünden artık ihtiyaçlarını karşılamadığı “Matrix-ATM” yazılımı ile karşılaştırmalı sonuçlar tasarım ekranlarını anlatan alt başlıklarda anlatılmıştır [43].

## 5.1. Sistem Modelleri

Hâlihazırda bulunan mevcut ATM cihazlarının çoğu, NDC standartlarına uygun olarak, bankaların ilgili birimlerindeki geliştiriciler tarafından veya dışarıdan satın alınan hizmet yazılımları tarafından kontrol edildiği bilinmektedir. Bu yazılımların geliştirilmesi, yeni taleplerin eklenmesi yazılımcılar tarafından yapılırken; arayüzün değiştirilmesi, iyileştirmeler vb. gibi çalışmalar tasarımcılar tarafından yapılır. Yazılımı ve tasarımı tamamlanan geliştirmeler ise daha sonra; taleplerde bulunan, operasyon

yöneticileri tarafından test edilerek arayüz ve geliştirmeler son halini alır. Bu süreçte tasarımcı, tasarladığı ekranı ATM'ye atmadan ekranda nasıl görüldüğünü bilememekte, yazılımcı ise tasarımcının ekranla ilgili işi bitmeden yazılıma başlayamamaktadır. Ayrıca geliştirmelerin son hali tamamlandıktan sonra da operasyon yöneticisinin testinde farklı anlaşılmalardan doğan hatalar yüzünden, ATM üzerinde görülen tasarımlar tekrar tasarım aşamasına geri dönmektedir. Bu süreç düşünüldüğünde; üç farklı birimin farklı işler yapmasına rağmen bir birine bağlı ve testinin bu kadar zor olduğu bir ortamda hızlı geliştirmeler yapmasının imkânsız olduğu görülür. Ayrıca oluşabilecek tasarım hatalarında geri dönüşlerin zor oluşu; çoğu zaman baştan tasarım yapılmasına neden olmakta, bazen de zaman kısıtları yüzünden hatalı tasarımlar olsa bile o şekilde yoluna devam etmesi gibi doğru olmayan çözümler doğurmaktadır. Bu sonuçlar düşünüldüğünde de bir arayüz yönetim sisteminin eksikliğini, kullanıcı kullanılabilirliğini ne denli düşürebileceği daha net görülmektedir.



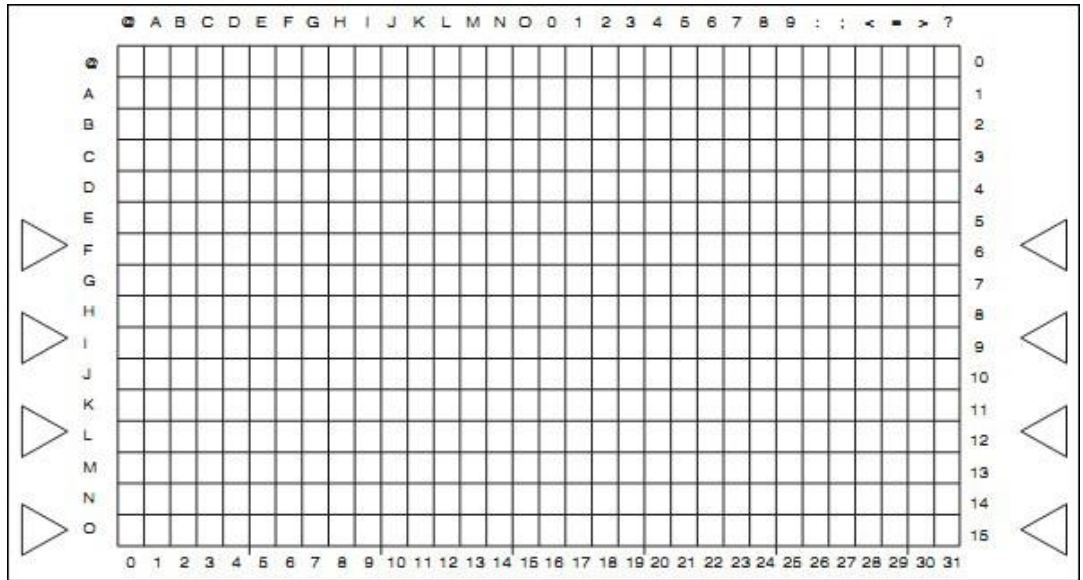
Şekil 5.1 : ATM arayüz yönetim sistemi kullanıcı modelleri.

Bu sebeple geliştirilen ATM arayüz yönetim sistemi ile farklı birimlerin aynı platform üzerinde paralel çalışabilmesi de amaçlanmıştır. Buna göre ATM arayüz

yönetim sistemi üzerinde tasarımcı ekran veya çıktı tasarımını yaparken, aynı anda yazılımcı da durum ve senaryo tasarımını yapabilmektedir. Ayrıca oluşan tasarımlar modüler ve sürükle bırak olduğundan, daha tasarım aşamasında operasyon yöneticisinin ekrana veya senaryoya müdahale etmesi kolaylıkla sağlanabilmektedir. Bu şekilde tasarım hatalarının geri dönüşü kolaylıkla yapılabilmekte ve test için ATM'ye gitme gereksinimi ortadan kalkmaktadır. ATM arayüz yönetim sisteminin sağladığı izole yapı sayesinde de tasarımcı, NDC standartlarını öğrenmek zorunda kalmamakta ve değişen durumlarda birden fazla tasarımcı ile birlikte hızlı ve kullanıcı dostu arayüzler tasarlayabilmektedir.

## 5.2. Ekran Tasarımı

Tez kapsamında yapılan çalışmada kullanılan NDC protokolüne göre ekranlar 32x16'lık bir ekran matrisi üzerine tasarlanabilmektedir [18]. Bu ekranlar üzerine gelebilecek her bir bileşen, bu matrisin herhangi bir noktasından başlamak üzere soldan sağa ve yukarıdan aşağı olmak üzere ATM tarafından konumlandırılır.



Şekil 5.2: NDC protokolü ekran formatı.

ATM'de gösterilebilecek dört tip bileşen vardır. Bunlar resim, sabit yazı, parametrik yazı ve veri girişi alanlarıdır. Ekranlar üzerinde; bu bileşenlerin her biri için

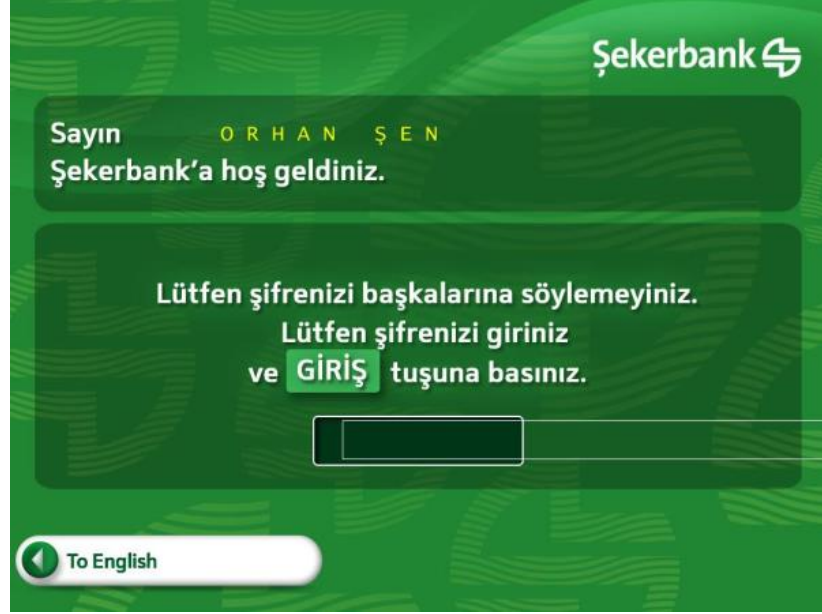
bir karakter setinin yanında ekranı temizleme, ekran arka plan rengi, ekran yazı tipi ve rengi, imleci verilen koordinata çekme vb. gibi durumların her biri için de oluşturulmuş karakter seti protokolde belirtilmiştir. Bir ekran tasarlamak için bu karakter setinin tamamının bilinmesi ve aynı zamanda bir biri üzerine gelecek resimler için koordinatların doğru formatta yapılması gerekmektedir. ATM üreticisi firmalar ekran tasarımında firmalara bir tasarım aracı sağlamamakla birlikte, tasarımın doğruluğu da ATM üzerinde test edilmeden bilinmemektedir. Örnek bir ekran tasarımı verisini inceleyecek olursak;

```
"<ESC>[11;23;80m<FF><SI>@@<ESC>P2011<ESC>\><SI>@N<ESC>P2300<ESC>\><SI>CH<TRACK1><SI>KM"
```

Şeklindeki bir ekran tasarımı şu anlama gelmektedir;

- <ESC>[11;23;80m : Arka plan rengi mavi, sabit yazı rengi sarı ve font tipi Lucida Console olan bir ekran yarat.
- <FF> : Ekranı temizle. (ekranlar üst üste gelen resimler şeklinde olduğundan, yeni bir ekrana geçmeden önce ekranın temizlenmesi gerekiyor.)
- <SI>@@ : İmleci @@ koordinatına getir.
- <ESC>P2011<ESC>\> : Diskte bulunan PIC011.jpg resmini @@ koordinatına koy.
- <SI>@N : İmleci @N koordinatına getir.
- <ESC>P2300<ESC>\> : Diskte bulunan PIC300.jpg resmini @N koordinatına koy.
- <SI>CH : İmleci CH koordinatına getir.
- <TRACK1> : Müşteri isim ve soy ismini karttan okur ve CH koordinatına koyar.
- <SI>KM : İmleci KM koordinatına getir.

Bu ekran ATM'ye yüklendiğinde şu şekilde görüntülenmektedir:

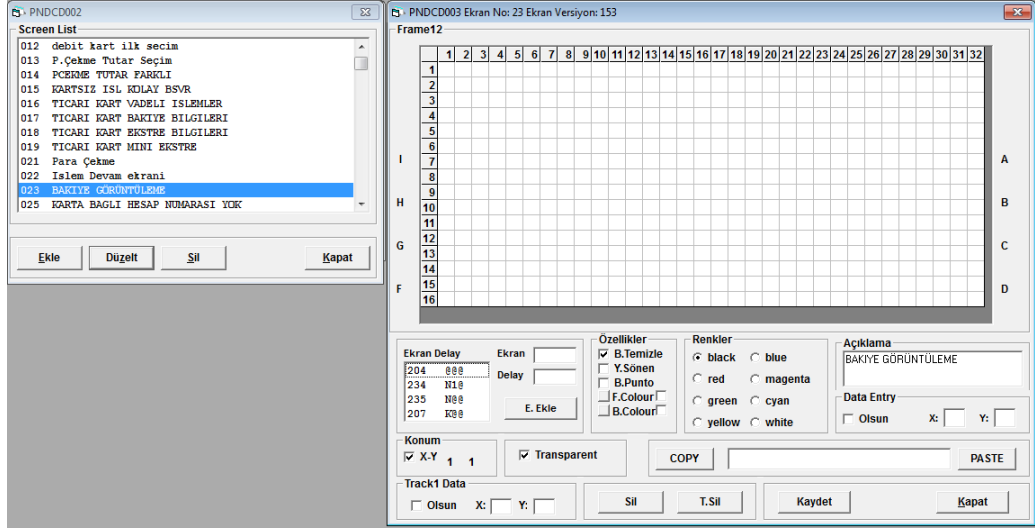


Şekil 5.3 : ATM üzerinde görülen örnek ekran görseli.

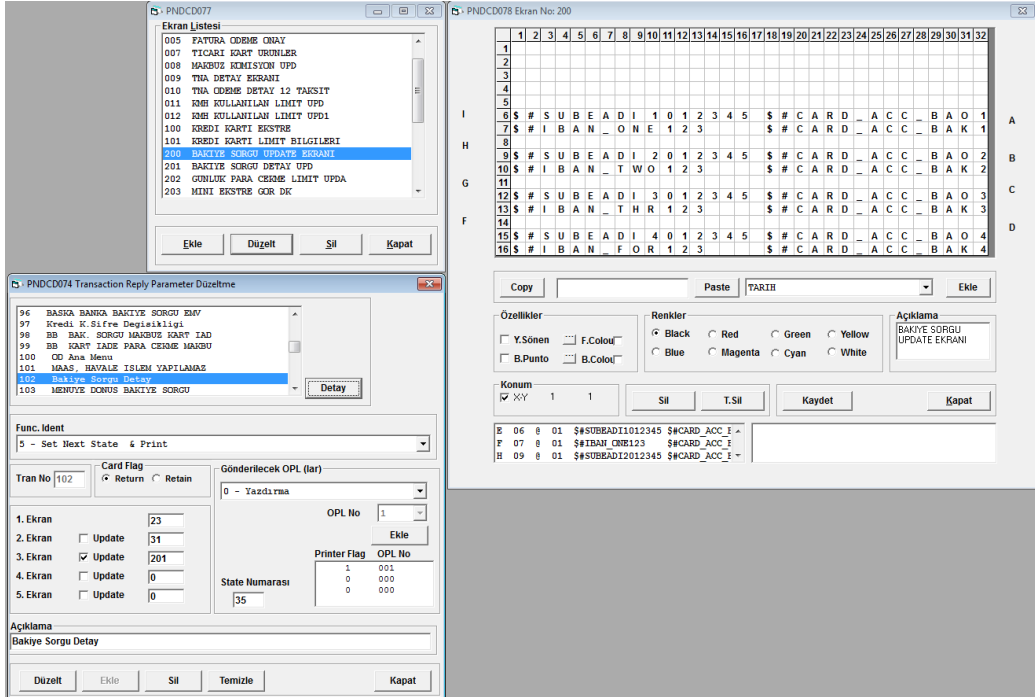
Örnekte verilen ekran için yapılan tasarım düşünüldüğünde, işlem setinin bilinmesi halinde bile bu şekilde görseli hayali olarak canlandırarak tasarım yapmak ve her değişikliği ATM üzerinde görmek zorunda olarak bu tasarımı test etmek, bir ekranın tasarımını uzun saatler alan bir süreç haline getirmektedir. Ayrıca örnekte verilen ekranda parametrik veriler gösterilmemiştir. Bir de sunucu tarafından dolan, her kullanıcı grubu için farklı karakterde olan parametrik yazılar düşünüldüğünde, bu sürecin yönetiminin oldukça zor bir süreç olduğu rahatça söylenebilir.

Çalışmakta olunan bankanın mevcutta kullanmakta olduğu yazılımın ekran tasarımı için sağladığı arayüzler Şekil 5.4 ve Şekil 5.5'de gösterilmiştir. Bu yazılıma göre ekran üzerinde görünecek hiçbir resim daha tasarım aşamasında bilinmemektedir. Kullanılacak resimler sadece resim numaraları olarak ekranda tanımlanmakta ve sadece bu resimlerin konum bilgileri tanımlanmaktadır. Bu şekilde bir tasarım düşünüldüğünde, resimlerin görülmeden tasarlandığı ve ortaya çıkan ekranın tahmin yoluyla yaratıldığı görülmektedir. Ayrıca bir ekran için statik resimlerin olduğu bir tanım ekranı ile parametrik yazıların olduğu ayrı bir tanım ekranının olması, parametrik yazıların ilk denemede doğru yerde olmasını sağlamayı daha da güçleştirmiştir. ATM gibi dış bir sistem için ekran tasarımının resimleri görmeden ve sadece ATM üzerinde test edilerek yaratılacak olması tasarımın ne denli zor bir süreç olduğunu ve artık ihtiyaçlara hizmet edemediğini göstermektedir.





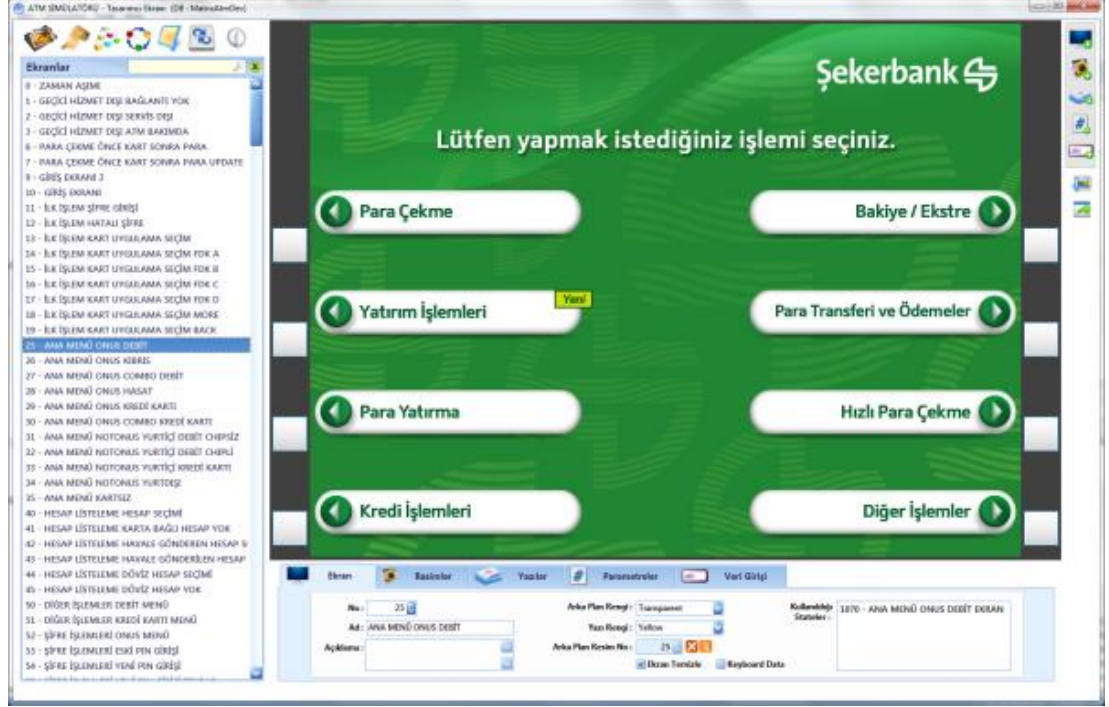
Şekil 5.4 : Matrix statik ekran tasarımı arayüzü.



Şekil 5.5 : Matrix parametrik ekran tasarımı arayüzü.

Bu sebeple ATM Arayüz Yönetim Sistem’inde Şekil 5.6’deki gibi NDC protokolüne uygun ekran tasarımlarının yapılabileceği bir araç sağlanmıştır. Bu sayede sistem modellerinden “Tasarımcı” modeli; NDC protokolünü öğrenmek zorunda kalmadan, resim kütüphanesinde olan resimler, sabit yazılar, parametrik değerler ve veri girişi alanlarını arayüzden sürükleyip bırak yöntemi ile ekrana bırakabilmekte ve ATM’de bire bir aynısını göreceği ekranı kısa sürede

tasarlayabilmektedir. Bu araç sayesinde ekran tasarımı; protokolden izole, nesneye dayalı yaklaşımla modüler ve yeniden kullanılabilir, verimli ve kullanıcı dostu olabilmektedir. Ayrıca ekranlar, bir sonraki bölümde anlatılan durum tasarımında kullanılacak durumlara da birer bileşen olacaklardır.



Şekil 5.6 : Ekran tasarım aracı arayüzü.

Sistem modellerinden “Operasyon Yöneticisi” modeli ise bu tasarımları daha tasarım aşamasında görebildiğinden, yazılımsal süreçlerde değişiklik gerektirmeden arayüz üzerine basit ve hızlı bir biçimde değişiklikler yansıtabilir. Bu da, ekranlar üzerine yönetilebilirliği ve geliştirilebilirliği arttıran temel faktörlerden biridir. Ekran tasarım aracının arayüze kazandırdığı bir başka avantaj ise arayüzde standart oluşturmasıdır. Protokol, tasarımcıya esneklik sağlarken, ekranlar arasında tutarsız tasarımlara da sebep olabilmektedir. Fakat kullanıcılar için ekran tasarımındaki tutarsızlıkların kullanılabilirliği oldukça düşürdüğü bilinmektedir. Örneğin, bir ekranda Evet tuşu sağda, Hayır tuşu solda iken; diğer bir ekranda tam tersi bir görüntüde olması, işlemi hızla yapan bir kullanıcı için istenmeyen sonuçlar doğurabilmektedir.

### 5.3. Durum Tasarımı

Tez kapsamında yapılan çalışmada kullanılan NDC protokolüne göre durumlar (state) Tablo 5.1 ve 5.2’de görüldüğü gibi 52 farklı tipten oluşmaktadır [18]. Her bir durumun ATM’deki ünite ve kullanıcı seçimlerine göre karşılık gelen bir davranışı olduğu gibi, kendi içinde de sabit bir özellikleri yoktur. Yani durumları birer nesne olarak düşünecek olursak; bir durumun 8 farklı nesne elemanı (data members) olurken, bir başka durumun 24 farklı nesne elemanı olabilir. Kimisi kullanıcıya bir ekran göstermek için kullanılırken, kimisi para verme ünitesinin açılması veya kartın okunması gibi aksiyonları almada kullanılır.

Tablo 5.1 : NDC protokolü durum tipleri (A-Z).

Tip	Açıklama	Tip	Açıklama
A	Card Read	M	Enhanced PIN Entry
B	PIN Entry	m	PIN & Language Select State
b	Customer Selectable PIN State	N	Camera Control
C	Envelope Dispenser	O	Cardholder Verification
D	Pre-Set Operation Code Buffer	R	Enhanced Amount Entry
d	Available as identifiers for Exit States	S	Language Code Switch
E	Four FDK Selection Function	T	Card Read - PIN Entry Initiation
F	Amount Entry	V	Language Select From Card
G	Amount Check	W	FDK Switch
H	Information Entry	w	Cheque Accept State
I	Transaction Request	X	FDK Information Entry
J	Close	Y	Eight FDK Selection Function
K	FIT Switch	Z	Extension State
k	Smart FIT Check State	z	Master Expansion State
L	Card Write		

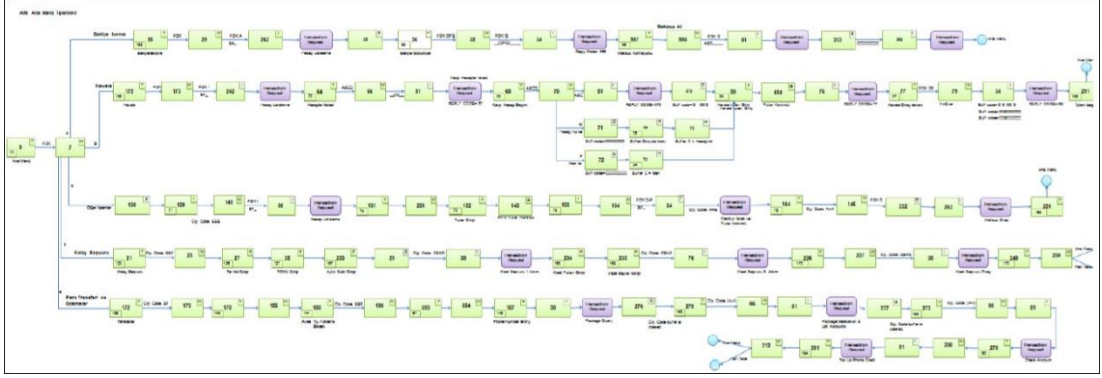
Tablo 5.2 : NDC protokolü durum tipleri (Diğer).

Tip	Açıklama	Tip	Açıklama
z001	Bunch Cheque Accept State	z900	Available as identifiers for Exit States
z002	Bunch Cheque Handling State	-	Automatic Language Selection
z003	Cheque Detail Display State	&	Barcode Reader State
z004	Display Cheque Summary State	,	Complete ICC Initialisation
z007	EMV Smart DIP Take Card State	.	Begin ICC Application Selection & Initialisation
z008	Insert Card State	/	Complete ICC Application Selection & Initialisation
z009	Buffer Validation State	;	ICC Re-initialise
z010	Local Cash Dispense State	?	Set ICC Transaction Data
z012	EMV ICC Application Switch State	_	Expanded FIT Switch
z013	Data Copy and Compare State	+	Begin ICC Initialisation
z014	Cash Deposit Handling State	>	Cash Accept State
z017	Note Mix Selection State		

NDC protokolünde 52 durumun ortalama 16 adet farklı alanı olduğunu düşündüğümüzde, 832 tane farklı alanı temsil eden bir yapıda, her bir alanın özelliği düşünülerek tasarım yapılmaya çalışılması çok da mümkün görülmemektedir. Ayrıca bir durumun içerdiği özelliklerin yanında satranç gibi birkaç durum sonrasının veya geri dönüş durumlarının da düşünülmesi gerekliliği, tasarımı gittikçe zorlaştırmaktadır.

Günümüzde durum tasarımı bankaların yazılım geliştiricileri tarafından parçalı olarak "Elle" veya çeşitli görselleştirilmiş tasarım araçları ile (Power Point gibi) yapılmaktadır. Peki, parçalı durum tasarımı ne demektir? Parçalı durum tasarımı; bir işlemin durum tasarımını ele aldığımızda sadece o işlemi içeren durumları düşünmek, işlemin devamında veya öncesinde hangi durumlardan geçişin olacağını düşünmemektir. Ayrıca durum tasarlarken sadece başarılı senaryoların göz önünde bulundurulması ve hata olabilecek senaryolarda durumların nasıl davranacağını elle

tasarımda hesaba katılmayacak kadar büyük olması, günümüz durum tasarımlarının bir diğer eksik yönüdür. Bu şekilde yapılan tasarımlarda akış büyüdükçe, hata yapma oranı yükselmekte ve bir birini çağırın durumlarda döngü yaratan akışlara çok sık rastlanmaktadır. Ayrıca bu tarz tasarımlar ekran görseli içermediğinden, hangi durumda kullanıcının hangi ekranı göreceği kâğıt üzerinde bilinmemekte ve akışı anlamak için her seferinde durumda kullanılan ekran görsellerine bakılmasını gerektirmektedir.



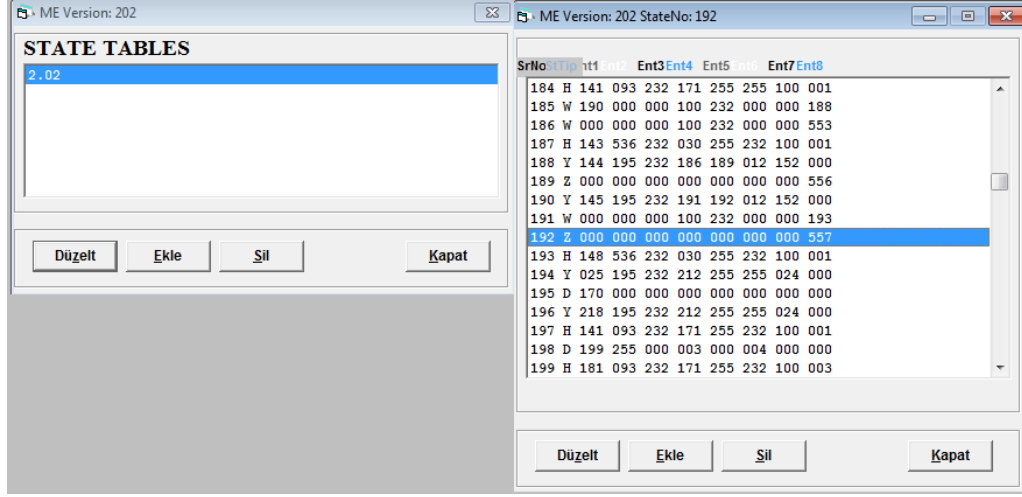
Şekil 5.7 : Günümüzde örnek bir durum tasarımı (Power Point).

Elle veya çeşitli görselleştirilmiş tasarım araçları ile durum tasarımının bir diğer dezavantajının da testinin yine ATM üzerinde yapılmak zorunda olmasıdır. Buna göre tasarım öncelikle Şekil 5.8’de görüldüğü gibi bir durum matrisine dönüştürülmeli ve bu matris verisi ATM’ye yüklenmelidir. Ardından test edilecek parçalı akışına kadar ATM üzerinde adımlar geçilmeli ve ara akış test edilmelidir. Burada en ufak bir akış değişikliğinin testinin bile yapılabilmesi için, ATM’ye kartın takılmasından itibaren tüm ana akışın geçilmesi ve test edilmesi gereken akışa gelmesi gerekmektedir. Bu da uzun zaman alan bir test süreci olarak karşımıza çıkmaktadır.

	ST_ORG	ST_VERSION	ST_STATE_NO	ST_STATE_TYPE	ST_ENTRY1	ST_ENTRY2	ST_ENTRY3	ST_ENTRY4	ST_ENTRY5	ST_ENTRY6	ST_ENTRY7	ST_ENTRY8	ST_ISN
1	100	202	587	O	588	0	0	0	0	0	0	0	587
2	100	202	588	Y	110	195	232	589	590	0	24	0	588
3	100	202	589	W	0	0	0	232	76	0	0	0	589
4	100	202	590	Z	0	0	0	0	456	0	0	0	590
5	100	202	591	D	592	0	0	0	0	0	0	0	591
6	100	202	592	Y	231	195	232	589	590	0	24	0	592
7	100	202	593	J	232	0	232	232	232	0	0	0	593
8	100	202	609	Y	206	195	232	212	255	12	24	0	676
9	100	202	610	W	0	0	0	100	232	605	0	0	677
10	100	202	599	Y	234	195	232	600	601	12	24	0	655
11	100	202	600	W	0	0	0	81	232	0	0	0	660
12	100	202	601	Z	0	0	0	111	0	0	0	0	661
13	100	202	602	Y	234	195	232	603	604	12	24	0	662
14	100	202	603	W	0	0	0	34	232	0	0	0	663
15	100	202	604	Z	0	0	0	5	0	0	0	0	664
16	100	202	605	Y	234	195	232	603	37	2	24	0	667
17	100	202	606	Y	234	195	232	600	390	12	24	0	668
18	100	202	607	Y	234	195	232	608	321	12	24	0	669
19	100	202	608	W	0	0	0	86	232	0	0	0	670
20	100	202	413	Y	16	195	232	417	479	12	24	0	1
21	100	202	414	Y	17	195	232	417	479	12	24	0	2
22	100	202	415	Y	18	195	232	417	479	12	24	0	3
23	100	202	417	W	0	0	0	81	268	0	0	0	4
24	100	202	171	I	22	93	0	0	1	0	0	3	5
25	100	202	26	Y	26	195	232	212	255	12	24	0	6
26	100	202	194	Y	25	195	232	212	255	12	24	0	7
27	100	202	195	D	170	0	0	0	0	0	0	0	8
28	100	202	446	D	447	0	0	0	0	0	0	0	9
29	100	202	447	Y	112	195	232	62	61	12	231	0	10
30	100	202	448	D	449	0	0	0	0	0	0	0	11
31	100	202	449	Y	112	195	232	62	61	12	195	0	12

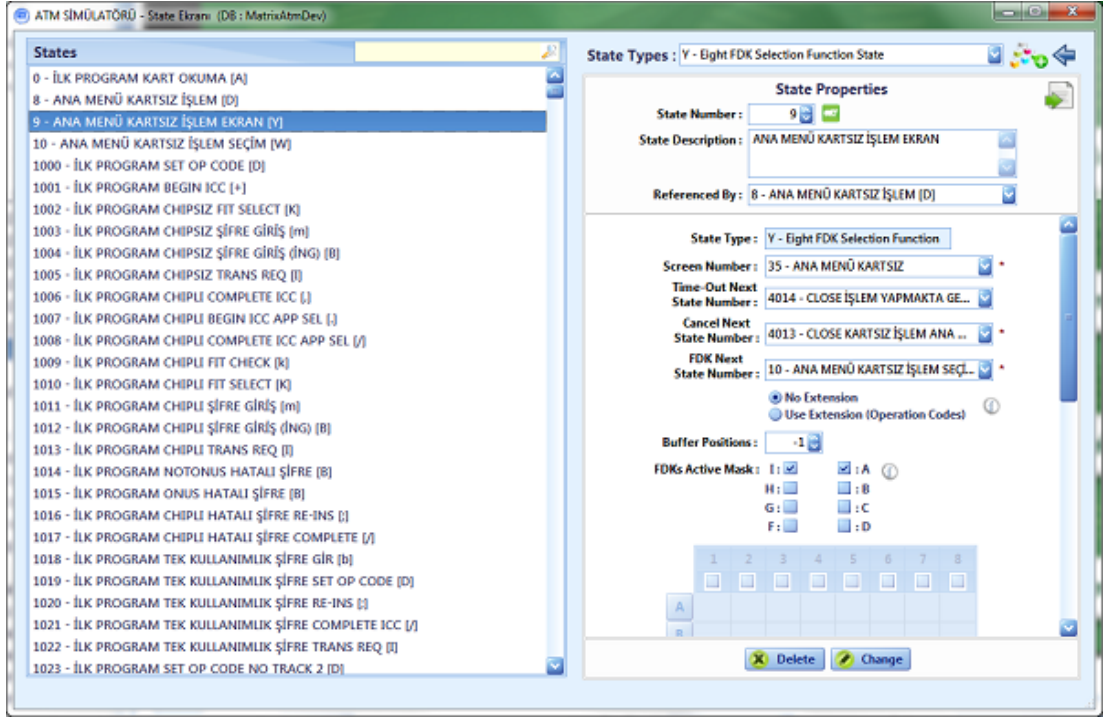
Şekil 5.8 : Örnek bir durum matrisi.

Çalışmakta olunan bankanın mevcutta kullanmakta olduğu yazılımın durum tasarımı için sağladığı arayüz Şekil 5.9'da gösterilmiştir. Bu yazılıma göre her hangi bir tipte tasarlanacak durum için bu durumun özellikleri NDC dokümanından bakılmalı ve her bir tipin ilgili alanına istenilen ekran veya durum yazılmalıdır. Şekildeki tasarım mantığını kısaca özetleyecek olursak; 184 numaralı durumun H tipinde bir durum olduğu ve birinci alanının kullanıcıya gösterilecek bir ekran numarasını temsil ettiği, ikinci alanının bir zaman aşımı durumunu temsil ettiği, üçüncü alanının iptal tuşu durumunu temsil ettiği vb. şekilde tüm alanların anlamını önce dokümandan öğrenmekle başlarız. Daha sonra bu alanlara eğer ekransa önceden yapacağımız ekran numarasını, durum ise yine önceden bu ekran ile tasarladığımız durumun numarasını yazarız. Bu şekilde bir biri ardına gelen durumlar şekilde görülen ekran aracılığıyla oradan oraya takip edilerek tasarlanır. Ayrıca her bir durum tipi için alanın sabit ve 8 olmaması, buna gelen ek durumların (extension state) ayrı tasarlanacak olması durumu daha da karmaşık bir hale getirmektedir. Böyle bir tasarımın ortaya doğru bir akış çıkarmasının ne denli zor olduğu, bunun yanında bu akışa eklenecek bir başka akışın ise bu şekilde eklemenin çok uzun süreler aldığı görülmektedir. Bu sebeple bu tarz bir tasarımın yeni ihtiyaçlara hizmet edemediği ve mevcut durumlar için bile sürekli sorunlar çıkardığı gözlemlenmiştir.



Şekil 5.9 : Matrix durum tasarımı arayüzü.

Bu zorluklar düşünüldüğünde, NDC protokolünün durum bilgisinden izole, nesneye dayalı bir tasarım ile kolay kullanılabilir bir ATM durum tasarım aracının gerekliliğinin ne denli büyük olduğu görülmektedir. Bu sebeple de ATM Arayüz Yönetim Sistem’inde Şekil 5.10’daki gibi, NDC protokolüne uygun durum tasarımlarının yapılabileceği bir araç sağlanmıştır. Bu sayede sistem modellerindeki “Tasarımcı” modeli; NDC protokolünü öğrenmek zorunda olmadan, her bir durum için yaratılan nesnelere üzerinden seçeceği birkaç özellik ile durum tanımını yapabilmekte ve bunları bir birine bağlayarak hızlı bir biçimde akışı tasarlayabilmektedir. Ayrıca bu araç sayesinde ek state (extension state) kavramının karmaşası tasarımcıdan soyutlanmış ve her bir durum kendi nesnesi içinde 8’den 24’e kadar alanla temsil edilebilir olmuştur. Bu sayede durum tasarımları; protokolden izole, nesneye dayalı yaklaşımla modüler ve yeniden kullanılabilir, minimum seviyede hata yapma oranı ile verimli ve yönetilebilir olmuştur. Ayrıca durumlar, bir sonraki bölümde anlatılan ve çalışma kapsamında senaryo adı verilen akış tasarımlarına da birer bileşen olacaklardır.



Şekil 5.10: Durum tasarım aracı arayüzü.

Durum tasarım aracının sağladığı arayüz, aynı zamanda tasarımcıyı işin içinden çıkılmaz bir hal alabilen durum matrisi karmaşasından da kurtarmaktadır. Tasarımcı tasarımlarında, NDC protokolü bilgisine müdahil olmadan tasarıma odaklanırken, sistem modellerinden “Operasyon Yöneticisi” modeli ise bu tasarımları daha tasarım aşamasında görebildiğinden, yazılımsal süreçlerde değişiklik gerektirmeden akışta basit ve hızlı bir biçimde değişiklikleri tasarımcıdan isteyebilmektedir. ATM üzerinde yapılması gereken test süreci de burada daha tasarım aşamasında olacağından, yanlışlardan geri dönüşler de hızlı olacaktır. Ayrıca test sürecinde geçirilecek zamanın kısalması, yeni talepleri hayata geçirme zamanını öne çekecek ve hizmet sağlayan servislerde hızlı artışı da sağlayacaktır.

Durum tasarım aracının getirdiği yönetilebilir yapı sayesinde son kullanıcı ise, arayüzde karşısına tasarımda hatanın minimum olduğu bir akışla karşılaşacaktır. Yaptığı aksiyonlara karşı aldığı bildirimlerin ve geri dönüşlerin basit ve anlaşılır olduğu bir akış düşünüldüğünde, sistemin kullanılabilirliği de artacak ve kullanıcılar tarafından sistem memnuniyeti oluşmuş olacaktır.



## 5.4. Çıktı Tasarımı

Tez kapsamında yapılan çalışmada kullanılan NDC protokolüne göre çıktılar iki tiptir. Bunlar ATM kullanıcısına işlem sonunda verilen “makbuz” ve yapılan işlem sonunda detayların tutulduğu ve ATM içindeki yazıcı tarafından çıkarılan “jurnal” çıktılarıdır. Protokole göre çıktılar 40x24’lük bir çıktı matrisi üzerine tasarlanabilmektedir [18]. Bu çıktılar üzerine gelebilecek her bir bileşen, bu matrisin herhangi bir noktasından başlamak üzere soldan sağa ve yukarıdan aşağı olmak üzere ATM tarafından konumlandırılır.

Çıktılarda gösterilebilecek iki tip bileşen vardır. Bunlar sabit yazı ve parametrik yazı alanlarıdır. Çıktı üzerinde; bu bileşenlerin her biri için bir karakter setinin yanında, alt satıra geçme, çıktıyı sonlandırma ve imleci verilen koordinata çekme gibi durumların her biri için de oluşturulmuş karakter seti protokolda belirtilmiştir. Bir çıktı tasarlamak için bu karakter setinin tamamının bilinmesi ve aynı zamanda yazı ve parametrelerin bir biri üzerine gelmeyecek biçimde çıktıda yer alması için koordinatlamının doğru formatta yapılması gerekmektedir. ATM üreticisi firmalar, ekran tasarımında olduğu gibi çıktı tasarımında da firmalara bir tasarım aracı sağlamamakla birlikte, tasarımın doğruluğu da yine ATM üzerinde kâğıt çıktı alınarak test edilmeden bilinmemektedir. Örnek bir çıktı tasarımı verisini inceleyecek olursak;

```
"<LF><SO>3TARİH:[#DATE:%-10s#]<SO>5ATM-NO:[#ATM:%-3s#]<LF><SO>3
SAAT:[#TIME:%-8s#]<SO>7İSLEM:[#TRAN_NO:%-5s#]<LF><LF><LF><SO>?PARA
ÇEKME<LF><LF><LF><SO>3KARTNO:[#CARD_NO:%-25s#]<LF><SO>3SUBE:
[#BRANCH_NAME:%-25s#]<LF><SO>3[#IBAN:%-34s#]<LF><LF><SO>3ÇEKİLEN
MİKTAR:<SO>1[#GIVE_AMOUNT:%15m#]<SO>1TL<LF><LF><SO>3BAKIYE:<SO>1
[#BALANCE:%15m#]<SO>1TL<LF><LF><SO>3KULL.BAKIYE:<SO>1[#USE_BALANCE:%1
5m#]<SO>1TL<LF><LF><SO>3KOMİSYON:<SO>1[#WITHDRAW_TOTAL_COMMISSIO
N:%15m#]<SO>1TL<LF><FF>"
```

Şeklindeki bir çıktı tasarımından bazı alanlar şu anlama gelmektedir;

- <LF> : Çıktıda imlecin bulunduğu yerden imleci bir alt satıra geçirir.
- <SO>3: İmlecin bulunduğu yerden sağa doğru 3 karakter boşluk bırakır.
- TARİH: İmlecin bulunduğu yere statik olarak tarih yazar.
- [#DATE:%-10s#] : İmlecin bulunduğu yere sunucu tarafından doldurulacak 10 karakterli bir tarih alanı yazar.
- <FF> : Çıktının bittiğini belirtir.

Bu çıktı, ATM'den kâğıt olarak alındığında ise şu şekilde görüntülenmektedir:

```

TARİH : #DATE#####          ATM-NO : #A#
SAAT  : #TIME###             İŞLEM  : #TRA#

                                     PARA ÇEKME

KART NO : #CARD_NO#####
ŞUBE    : #BRANCH_NAME#####
#IBAN#####

ÇEKİLEN MİKTAR: ###GIVE_AMOUNT# #T#

BAKİYE          : #####BALANCE# #T#

KULL. BAKİYE    : ###USE_BALANCE# #T#

KOMİSYON        : #WITHDRAW_TOTA# #T#

MAKBUZ BEDELİ OLARAK #COM# TL ÜCRET,
+ #BSM# TL BSMV TAHSİL EDİLMEKTEDİR.

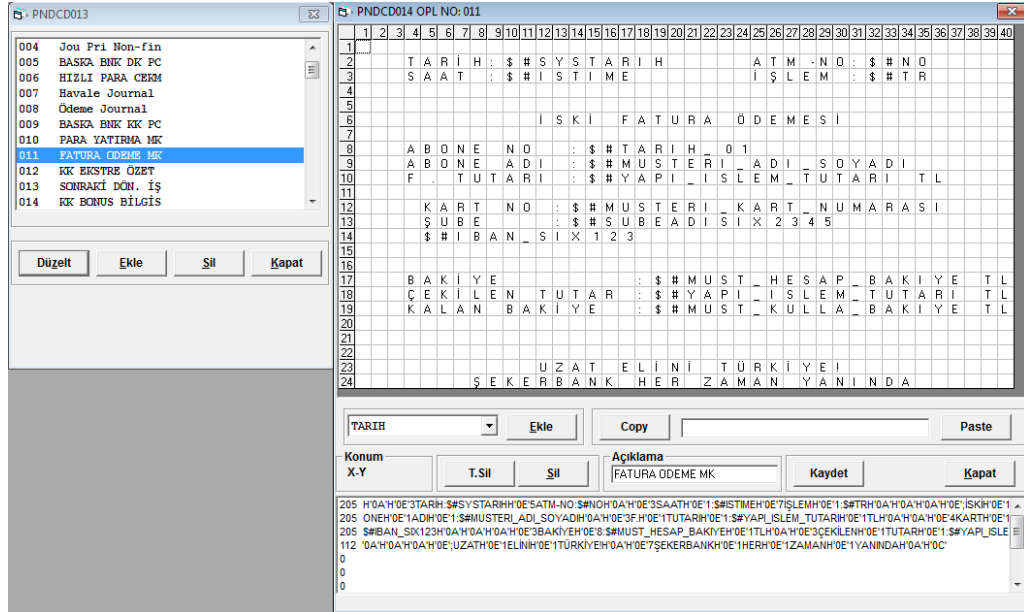
```

Şekil 5.11: ATM'den alınan örnek çıktı görüntüsü.

Örnekte verilen çıktı için yapılan tasarım düşünüldüğünde, işlem setinin bilinmesi halinde bile bu şekilde bir çıktıyı hayali olarak canlandırarak tasarım yapmak ve her değişikliği ATM'den alınacak çıktıda görmek zorunda olarak bu tasarımı test etmek, bir çıktının tasarımını bile uzun bir süreçte yapmayı gerektirmektedir. Ayrıca örnekte verilen çıktıda parametrik veriler gösterilmemiştir. Bir de sunucu tarafından dolan, her kullanıcı grubu için farklı karakterde olan parametrik yazılar düşünüldüğünde bu sürecin yönetiminin oldukça zor bir süreç olduğu rahatça

söylenbilir. Sürecin zorluğunun yanında testinde oluşacak kâğıt israfının da ne kadar fazla olacağı açıkça görülebilmektedir.

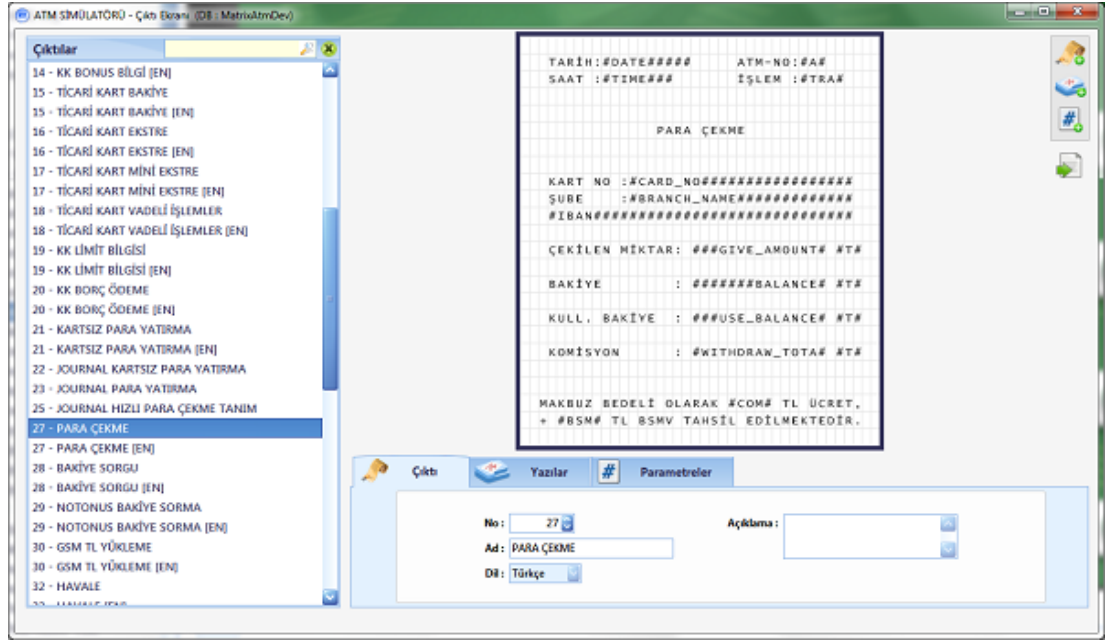
Çalışmakta olunan bankanın mevcutta kullanmakta olduğu yazılımın çıktı tasarımı için sağladığı arayüz Şekil 5.12’de gösterilmiştir. Bu yazılıma göre her hangi bir tipte tasarlanacak çıktı için ekrandan görülen matris üzerine yazılar yazılmakta, fakat bu yazıların kâğıt üzerinde nasıl duracağı bilinmemektedir. Bu sebeple tasarımı yapan kullanıcı her bir tasarım değişikliğinin ardından kâğıt çıktı almak zorunda olmakta ve bu şekilde tasarımın süresi oldukça uzamaktadır. Ayrıca bir biri üzerine gelen yazılar, bu ekran tarafından kontrol edilmemekte ve bu durumlarda tasarımı zorlaştırmaktadır. Bu sebeple bu tarz bir tasarımın yeni ihtiyaçlara hizmet edemediği ve hatta farklı dillerdeki makbuzlar bu arayüz ile tasarlanamadığından sadece tek bir dilde makbuz vermek zorunda kalındığı gözlemlenmiştir.



Şekil 5.12 : Matrix çıktı tasarımı arayüzü.

Bu sebeple ATM arayüz yönetim sisteminde Şekil 5.13’deki gibi NDC protokolüne uygun çıktı tasarımlarının yapılabileceği bir araç sağlanmıştır. Bu sayede sistem modellerindeki “Tasarımcı” modeli; NDC protokolünü öğrenmek zorunda olmadan, sabit yazılar ve parametrik yazıları arayüzden sürükleyip bırak yöntemi ile çıktıya bırakabilmekte ve ATM’den aldığı birer birer aynısını göreceği çıktıyı çok kısa sürede tasarlayabilmektedir. Bu araç sayesinde çıktı tasarımı da; protokolden izole,

nesneye dayalı yaklaşımla modüler ve yeniden kullanılabilir, verimli, çevreci ve kullanıcı dostu olabilmektedir.



Şekil 5.13 : Çıktı tasarımı arayüzü.

Sistem modellerinden “Operasyon Yöneticisi” modeli ise bu tasarımları daha tasarım aşamasında görebildiğinden, yazılımsal süreçlerde değişiklik gerektirmeden çıktı arayüzü üzerine basit ve hızlı bir biçimde değişiklikler yansıtılabilir. Bu da çıktılar üzerine yönetilebilirliği ve geliştirilebilirliği arttıran temel faktörlerden biridir. Çıktı tasarımı aracının, çıktı arayüzüne kazandırdığı bir başka avantaj ise çıktılarda standart oluşturmaktır. Protokol tasarımcıya esneklik sağlarken, çıktılar arasında tutarsız tasarımlara da sebep olmaktadır. Fakat kullanıcılar için çıktılardaki koordinat tutarlılığı ve görsel standartlar sistemin kullanılabilirliği doğrudan etkileyen yönlerdir.

## 5.5. Senaryo Tasarımı

Tez kapsamında kullanılan NDC protokolüne uygun tasarım ekranlarını beşinci bölümde başlıklar halinde belirttim. Tasarlanan ekranlar durumların birer bileşeni olurken, tasarlanan durumlar ise çalışma kapsamında “senaryo” adını verdiğimiz akış

diyagramlarının birer bileşeni olacaklardır. Peki, senaryo nedir, ne amaçla tasarlanır? Senaryoyu şu şekilde tanımlayacağız:

Senaryo; belirli bir amaca hizmet etmek amacıyla tasarlanmış, en az bir başlangıç ve bir bitiş düğümü (node) olan durumlar bütünüdür.

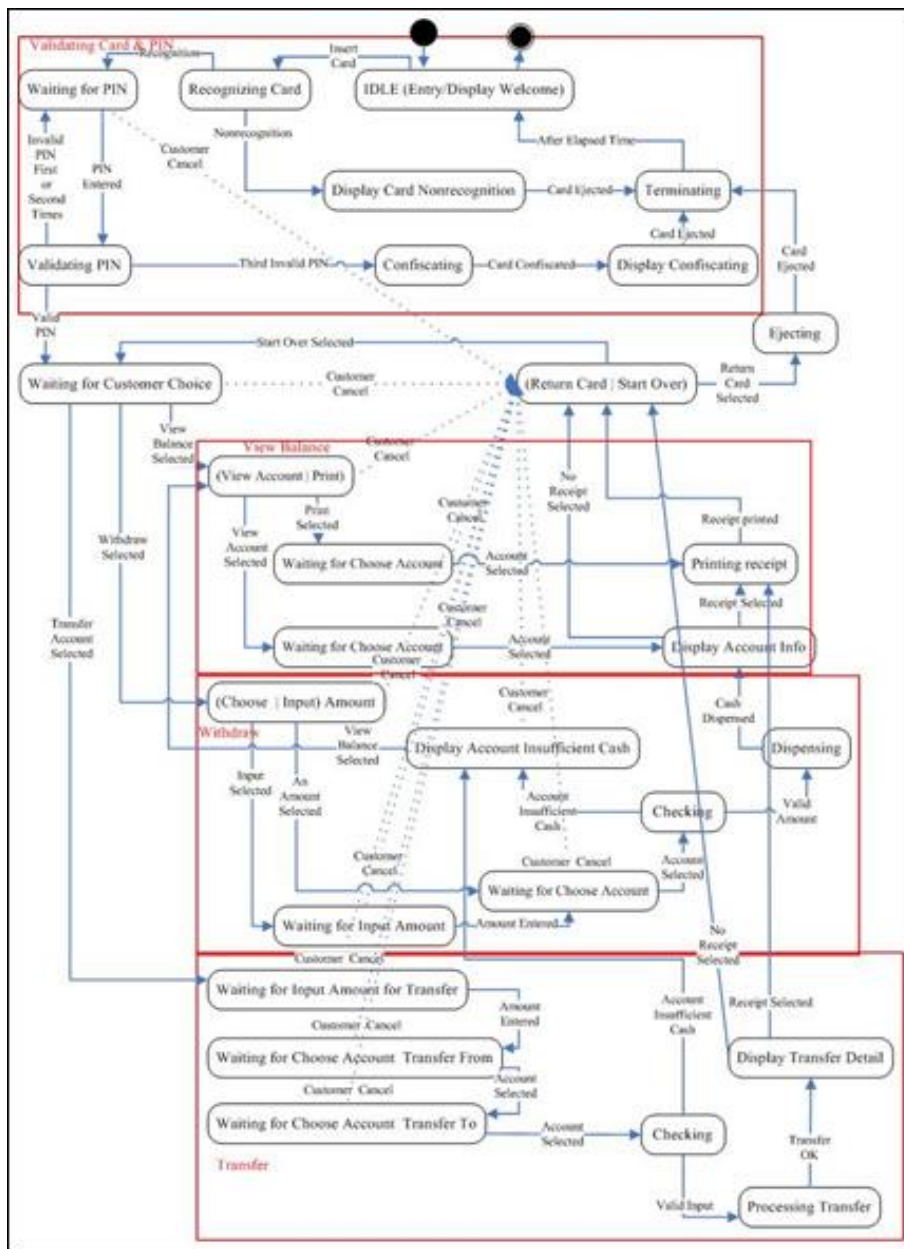
Yapılmış olan tanımdan da anlaşılacağı gibi birden fazla durumun birleşmesi ile oluşan, bir amaç için tasarlanmış ATM akışlarıdır. Örnek; Para Çekme senaryosu, Bakiye Sorma senaryosu, Şifre Değiştirme senaryosu vb. Senaryolar başlangıç ve bitiş olarak belli bir amacı tamamen içereceği gibi, belli bir senaryonun bileşeni olacak alt senaryolarda olabilirler. Bu şekilde düşünüldüğünde bir birine bağlı tüm senaryolar aslında ATM'nin tamamını içinde barındıran bir durum makinasının (state machine) akış diyagramını (senaryolar bütünü) oluşturacaktır. ATM senaryoları için iki çeşit senaryo tipinden bahsedebiliriz:

- Başarılı Senaryo: Başlangıç düğümünden, bitiş düğümüne kadar; amaca hizmet eden olası durumları içeren senaryo çeşididir. (Örneğin; hesapların doğru bir şekilde listelenmesi, para çekmenin olası akışı vb.)
- Başarısız Senaryo: Yaratılan senaryoda kullanılan durumların, beklenmedik aksiyonlarında oluşabilecek durumları içeren senaryo çeşididir. (Örneğin; zaman aşımı, para sıkışması, kartın alınamaması vb.)

Durum tasarımında yaşanan problemlerden 5.3. bölümünde bahsetmiştim. Bu problemlerin yanında, sadece duruma odaklanarak yapılan tasarımların aslında akışı tam olarak çıkarmadığı ve akışta oluşabilecek tüm durumların hesaba katılmadan hayata geçmesine sebep olmaktadır. Bu sebeple de çok fazla işlem yükü olan ATM gibi bir sistemin, yeni gelecek isteklere cevap veremediği ve büyüyen akışla birlikte yönetilebilir olmaktan çıktığı görülmektedir. Ayrıca günümüz literatüründe kullanılan akış diyagramlarının ATM'ye uygulanması durumunda, sayfalarca bir birine bağlı bir diyagram ortaya çıkmakta ve anlaşılması imkânsız bir akış görüntüsü oluşmaktadır.

Literatürde, işlem gereksinimlerinden oluşan senaryolardan bir kullanıcı arayüz ilk örnekleri çıkarma çalışmasının yapıldığı görülmüştür [44]. Buna göre gereksinimlerden önce kullanım durumu diyagramları (use case diagrams) , bu diyagramlardan çıkan fonksiyonlara göre de işbirliği diyagramları (collaboration

diagrams) ve durum diyagramları (statechart diagrams) çıkarılmıştır. Bu diyagramlar senaryolara göre çıkan gereksinimlerin analiz edilip tanımlanmasını sağlamıştır. Her bir senaryonun ilişkisi bir çizelge (graph) ile belirlenmiş ve arayüz örnekleri oluşturma algoritmasının temelini oluşturmuştur. Burada çizelgenin bir birine olan geçişleri aslında arayüzünde kullanıcılar için etkileşimini ortaya çıkarmıştır. Geçişlerin olmadığı düğümler ise kullanıcının yapamayacağı işlemleri göstermektedir. Araştırılan bu çalışmadaki algoritmaların ATM gibi durum tasarımı içeren sistemler için, durum diyagramları oluşturmada kullanılabilir bir örnek olduğu gözlemlenmiştir.



Şekil 5.14 : Örnek bir ATM akış diyagramı.

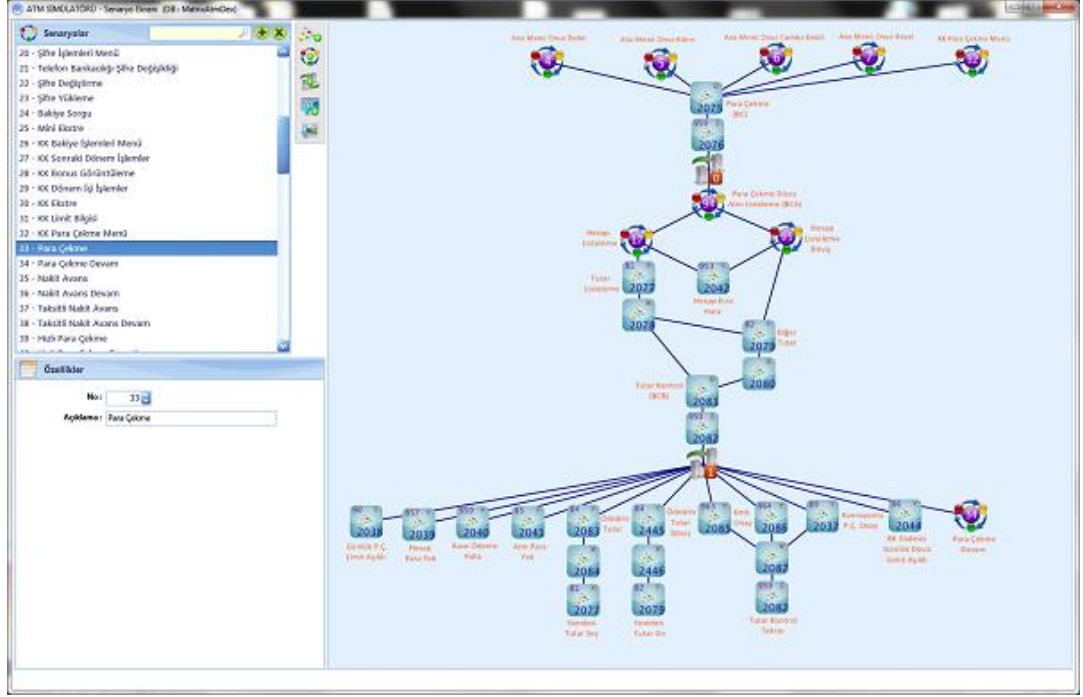
Çalışmakta olunan bankanın mevcutta kullanmakta olduğu yazılımın senaryo tasarımını sağlayacağı bir arayüzü bulunmamaktaydı. Fakat ATM'den gelen her bir işlem koduna karşılık gideceği durum ve ekranın tanımlandığı arayüz 5.15'de gösterilmektedir. Bu tanımlama ekranına göre hangi işlemlerin hangi ekranları ve durumları içerdiği, hangi durumlarda sunucuya gelip hangi durumlarda ATM üzerinde ilerleyeceği ve başarısız durumlarda hangi akışın geçerli olacağı gibi sorular cevapsız kalmaktadır. Bu sebeplerle mevcut akışlara yeni akışların eklenmesi için uzun uğraşlar gerekmekte ve değişen ihtiyaçlara karşın, eski akışlarda değişiklik yapmanın, değişikliğin etkilediği yerler tespit edilemediğinden mümkün olmadığı görülmektedir.

Op. Code	Program Adı	Açıklama	İşlem Kodu
EEZZZZEE	PNDCN025	SOLICITUD REPLY	KL
CBF88888	PNDCN209	KREDİ KARTI LİMİT BİLGİLERİ	GS
BEF88888	PNDCN203	GSM TL SORGU/YUKLEME	GS
EEF88888	PNDCN030	HIZ PARA ÇEK TAHLİLANA	ET
CCCCCCCC	PNDCN02C	ANA MENU	IP
CBAD8888	PNDCN204	KREDİ BASVURU GIRIS	AP
BEF88888	PNDCN204	KREDİ BASVURU SORGULAMA	AI

ID	Code	Sıra
Data_01	0001	100 DD Ana Menu
Data_02	0002	108 CS KART ANAMENU
Data_03	0003	139 BASKA BANKA DD ANA MENU
Data_04	0004	158 BASKA KART MENU
Data_05	0005	167 KARTSIZ ISLEM ANA MENU
Data_06	0006	185 BASKA BANKA DD ANA MENU
Data_07	0007	186 BASKA BANKA YURT DISI ANA MENU
Data_08	0008	078 DD Chipli Menu
Data_09	0009	224 MIBRIS KREDİ ANA MENU
Data_10	0000	000
Data_11	0000	000
Data_12	0000	000
Data_13	0000	000
Data_14	0000	000
Data_15	0000	000
Data_16	0000	000
Data_17	0000	000
Data_18	0000	000
Data_19	0000	000
Data_20	0000	000

Şekil 5.15 : Matrix işlem tanımlama arayüzü.

Günümüzde kullanılan elle veya görselleştirilmiş araçlarla yapılan parçalı tasarımların yetersiz kalması sebebiyle, ATM Arayüz Yönetim Sistem'inde Şekil 5.16'da görüldüğü gibi senaryo tasarımlarının yapılabileceği bir araç sağlanmıştır.



Şekil 5.16 : Senaryo tasarım aracı arayüzü.

Senaryo Tasarımı aracının hangi problemlere ne tarz çözümler getirdiğini maddeler halinde sıralarsak;

- NDC Protokolü karmaşasından uzaklaşarak sadece akışa odaklanarak tasarım yapmayı sağlar

Protokolün getirdiği durum karmaşası problemine karşın, durumlara odaklanmak yerine akışın amacına ve akışta oluşabilecek tüm durumlara yönelmek, daha doğru ve eksiksiz akışlar çıkardığı görülmüştür.

- Tüm akışı aynı anda düşünmek yerine, her bir akışı kendi içinde tüm durumları ile düşünerek, senaryoların tekrar tekrar kullanılabilmesini sağlar

Günümüzde durum tasarımlarının yönetilebilir olmaması sebebiyle, tekrar tekrar kullanılmaya da olanak sağlamaması en büyük problemlerden biridir. Senaryo tasarımıyla birlikte her bir senaryoya kendi içinde bütünlük sağlanmış, bir senaryo başka bir senaryonun içinde tekrar tekrar kullanılabilir olmuştur.

- Senaryo temelli test yapılabilmeye olanak sağlar



ATM'de oluşacak durumların testi; akışın en başından başlanmak üzere, test edilmesi istenen yere kadar gelindikten sonra başlar. Bu en içerdeki bir akış için, çok uzun ve zorlu bir süreçtir. Fakat senaryo tasarımı ile her bir senaryo tek bir amaca hizmet edeceğinden, senaryonun testi sadece o senaryonun başından başlatılarak hızlı ve verimli testler yapılabilecektir. Senaryodaki başarısız durumların atlanması ise tüm senaryoya odaklanma olmayacağı için daha aza indirilmiş olacak ve olası hatalarda ATM davranışının nasıl olması gerektiği kolayca tespit edilebilecektir. Ayrıca tek bir senaryonun test edilmesi, diğer senaryolardan etkilenen durumlar oluşturmayacağından, testlerdeki başarı oranı da yükseltecektir.

- Arayüzün performansı “ölçülebilir” olacaktır

Günümüzde çoğu sistemin arayüz performansı ölçülebilir olmamaktadır. Performansın ölçülebilir olmaması, arayüzdeki bir karışıklığında fark edilemiyor olmasına, belki de bu karışıklıktan ötürü fazladan işlem yapılıyor olmasına sebep olabilir. Bu probleme çözüm olarak kullanılacak senaryo tasarımları sayesinde, senaryoların hangi işlemlerde ne kadar kullanıldığının analizi yapılacak ve arayüz bu analiz sonuçlarına göre iyileştirilecektir. Arayüz performansının ölçülmesi ve yapılan analizlerin sonuçlarına altıncı bölümde detaylı yer verilmiştir.

- “Kişiyeye özgü senaryo” (kişiyeye özel arayüz) tasarımına olanak sunar

Ölçülebilir olmayan arayüz performansı, üzerinde arayüz iyileştirmeleri de yapılamaması anlamına gelir. Fakat ölçülebilir performanslar üzerinde yapılacak analizler ile kullanıcı merkezli işlem tipi ve adetleri hesaplanarak, hangi kullanıcıların hangi senaryoları daha fazla gerçekleştirdiği sonucunu ortaya çıkaracaktır. Bu analizler sonucu ortaya çıkacak profil verilerine göre de yeni senaryo akışları tasarlanacak ve arayüzlerde kişiselleştirme yapılarak, sistemin kullanılabilirliğinin artması hedeflenecektir. Çıkarılacak kullanıcı profillerinin tespitine ve bu kullanıcı tiplerine yapılan arayüzde kişiselleştirmenin nasıl sağlandığına yedinci bölümde detaylı yer verilmiştir.

ATM Arayüz Yönetim Sistemi'nde yaratılan “Senaryo Tasarımı” aracında üç farklı bileşene yer verilmiştir;

- Durum (State)

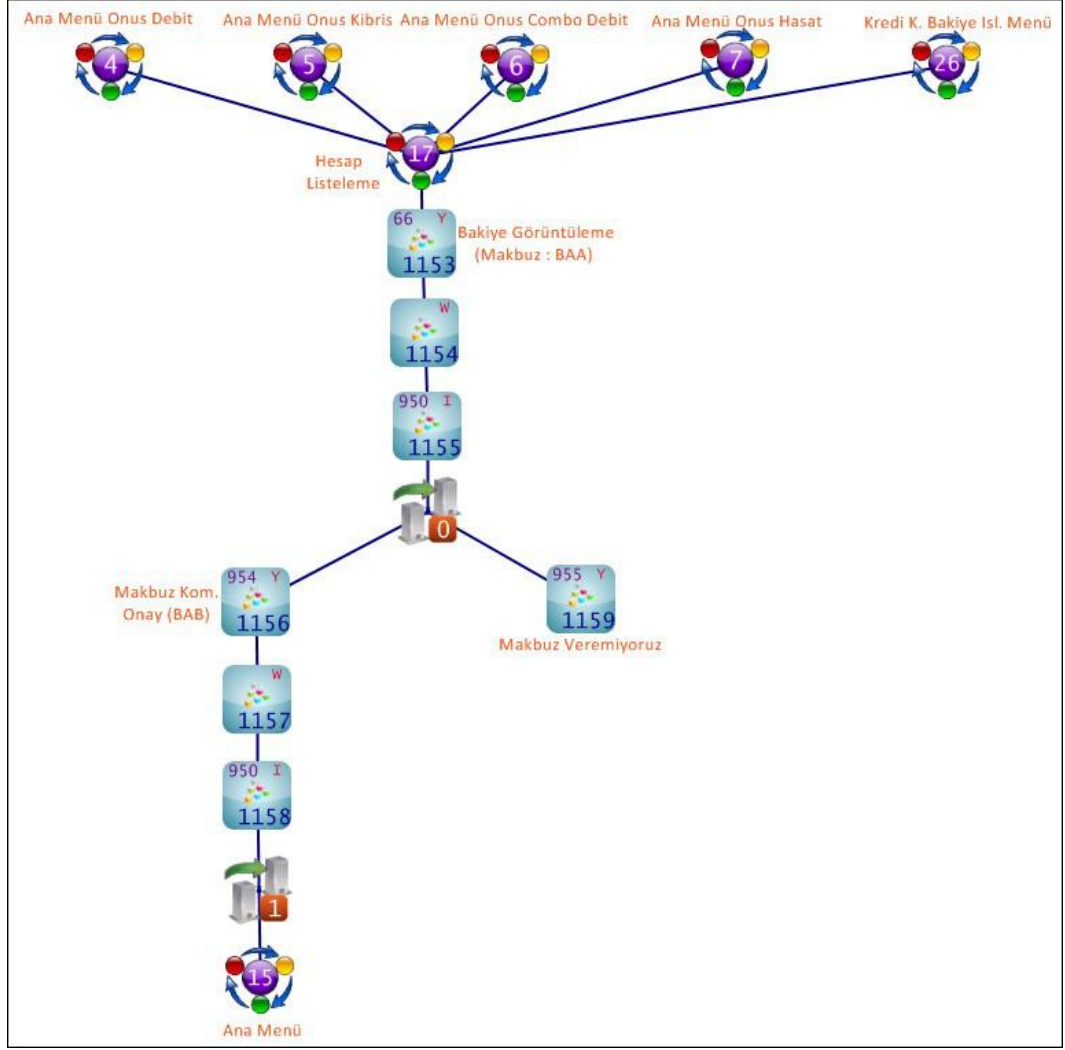
NDC protokolüne göre durum tasarım ekranında tanımlanan 52 farklı durumdan her hangi birini içeren ve senaryo içinde sadece bir durumu temsil eden bileşendir. Senaryo içinde durumlar; bir durumdan başka bir duruma geçerken bir durum bileşenine, sunucuya bir durumu bildirecekken işlem mesajı (transaction request) bileşenine ve bir başka senaryo akışına geçecekken de senaryo bileşenine bağlanabilir.

- İşlem Mesajı (Transaction Request)

ATM'de yapılan herhangi bir işlemdeki kullanıcı bilgilerinin ATM üzerinde tutulmadığını daha önceki bölümlerde belirtmişim. Kullanıcı aksiyonlarına göre bir takım akışından geçtikten sonra bilgi gösterilmesi gereken durumlarda, sunucuya gidileceğini temsil eden bileşen de işlem mesajı bileşenidir. Senaryo tasarım aracında bu bileşen, aynı zamanda sunucu tarafından yönlendirilecek durumların daha tasarım aşamasında belli olması amacını da taşır. Tasarımda bu bileşene, bir durum bileşeni veya bir başka senaryo akışı bileşeni bağlanabilir.

- Senaryo

Senaryoların tekrar tekrar kullanılabilmesi amacını temsil eden bu bileşen sayesinde, tasarımda bir akış bir başka akışa kolayca bağlanabilir. Senaryolar; senaryo bileşeni sayesinde bir başka durum bileşenine veya bir işlem mesajı bileşenine bağlanabilir.



Şekil 5.17: Örnek bakiye sorma senaryo tasarımı.

Senaryo Tasarımı aracı; sistem modellerinden “Tasarımcı” modelinin kullandığı bir ekran olup, “Operasyon Yöneticisi” modeline de hizmet eder. Buna göre istenilen geliştirme akışı doğruluğunun kontrolü, daha yazılıma başlanmadan tasarım aşamasında yapılır. Bu kontrol, akış üzerindeki hataların tespit edilmesini ve geri dönüşü kolaylaştırırken, bazen test ortamlarında gerçekleştirilemeyecek durumlarda da sistemin nasıl davranacağını göstereceğinden sistemi hatasız ve verimli kılar.

Arayüzde kullanılabilirliğin en yüksek seviyede olması, akış tasarımının en doğru ve hatasız bir biçimde yapılmasından geçer. Buna göre iyi bir akış tasarımında kullanıcı, yapacağı işlemde gördüğü bir ekrandan sonra neyle karşılaşacağını bilerek işlemini gerçekleştirebilir. Her hangi bir yanlış işlem yaptığında ise göreceği mesajın anlaşılabilir ve net olması, geri dönüşünün kolay ve telafi edilebilir nitelikte olması

gerekmektedir. Akışlar bu şekilde olduğu sürece de sistemin kullanılabilirliği artacak ve kullanıcı, arayüzü kendine ait hissedecektir. Akışların bu denli doğru tasarlanması içinde nesneye dayalı bir tasarım aracı olan ATM arayüz yönetim sistemi senaryo tasarımı aracınının, ne kadar gerekli olduğu görülmektedir.

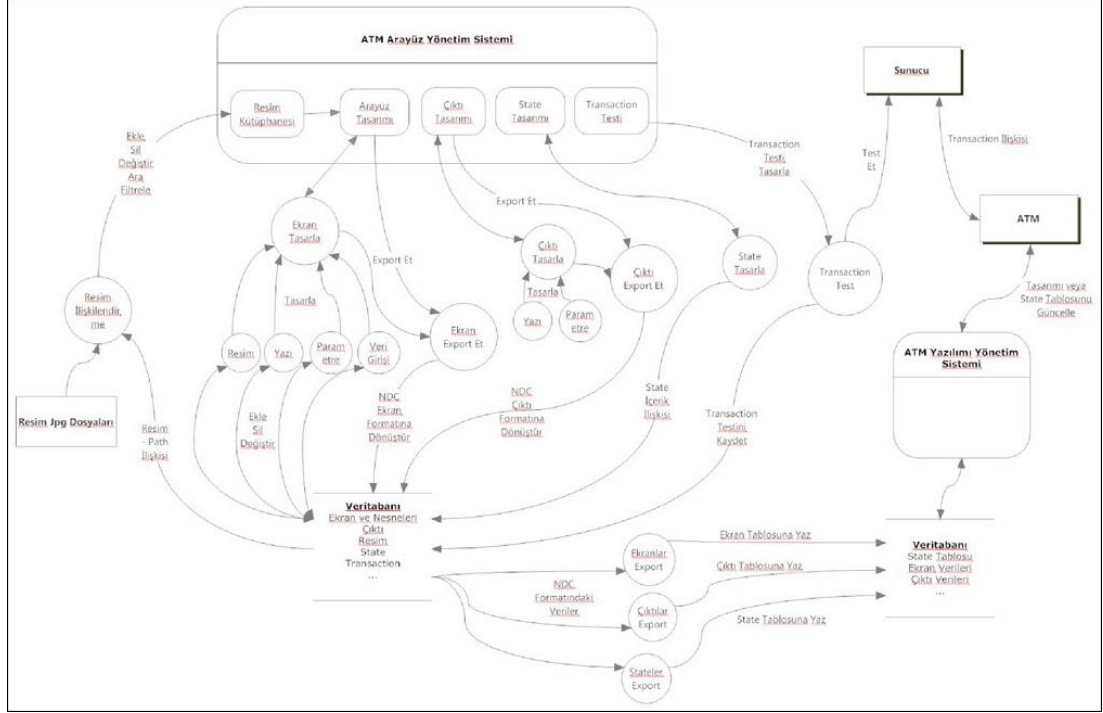
Senaryo tasarımınının getirdiği avantajlardan olan ölçülebilir arayüz performansı ve arayüzde kişiselleştirme, tez kapsamında altıncı ve yedinci bölümlerde detaylı bir şekilde anlatılmıştır.

## 5.6. Yöntem ve Sistem Mimarisi

ATM Arayüz Yönetim sistemine bir de Şekil 5.18’de gösterilen veri akış diyagramı (data flow diagram) üzerinden bakacak olursak [45]; sistemin “Tasarımcı” modelinin ekran, çıktı ve durum tasarımlarının yapıldığı arayüzü kullandığı bilinmektedir. Bu tasarımlar, her bir bileşen ayrı olacak şekilde veri tabanında tutulmaktadır. Tasarımcı modelinin bir diğer aracı olan resim kütüphanesi ise ATM üzerinde tutulan “.jpeg” dosyaları ile ilişkilendirilmiş resimlerin özelliklerini veri tabanında saklar. Tasarım aşamasındaki bu veriler, sistemin “Operasyon yöneticisi” modeli tarafından tasarım görselleri üzerinden onaylandığında, tasarımcı modeline sağlanan dönüştür (export) fonksiyonu ile NDC protokolünün ekran, durum ve çıktı formatlarına dönüşür. Bu aşamadan sonra arayüzde kullanılacak tüm tasarımlar tamamlanmış ve iş, bu tasarımlara eşlik edecek NDC işlem (transaction) mesajlaşması üzerinden ATM sunucusu ile konuşacak yazılıma bırakılmıştır. Sistemin “Yazılım Geliştiricisi” modeli ise bu aşamadan sonra tasarımcı modelinin oluşturduğu NDC protokolü formatındaki verileri ATM’ye yükleyerek, istenen fonksiyonun senaryo akışını kodlayacaktır. Ayrıca istenen fonksiyonun akışını kodlarken yapılacak testler ise sistemin “işlem testi” ekranı üzerinden test edilebilecek ve ATM’ye gitmeye gerek olmadan senaryonun tamamlanması sağlanabilecektir. Yapılacak testler aynı zamanda veri tabanına kayıt edilerek, farklı durumlar (başarılı veya başarısız senaryolar) için tekrar tekrar kullanılabilir.

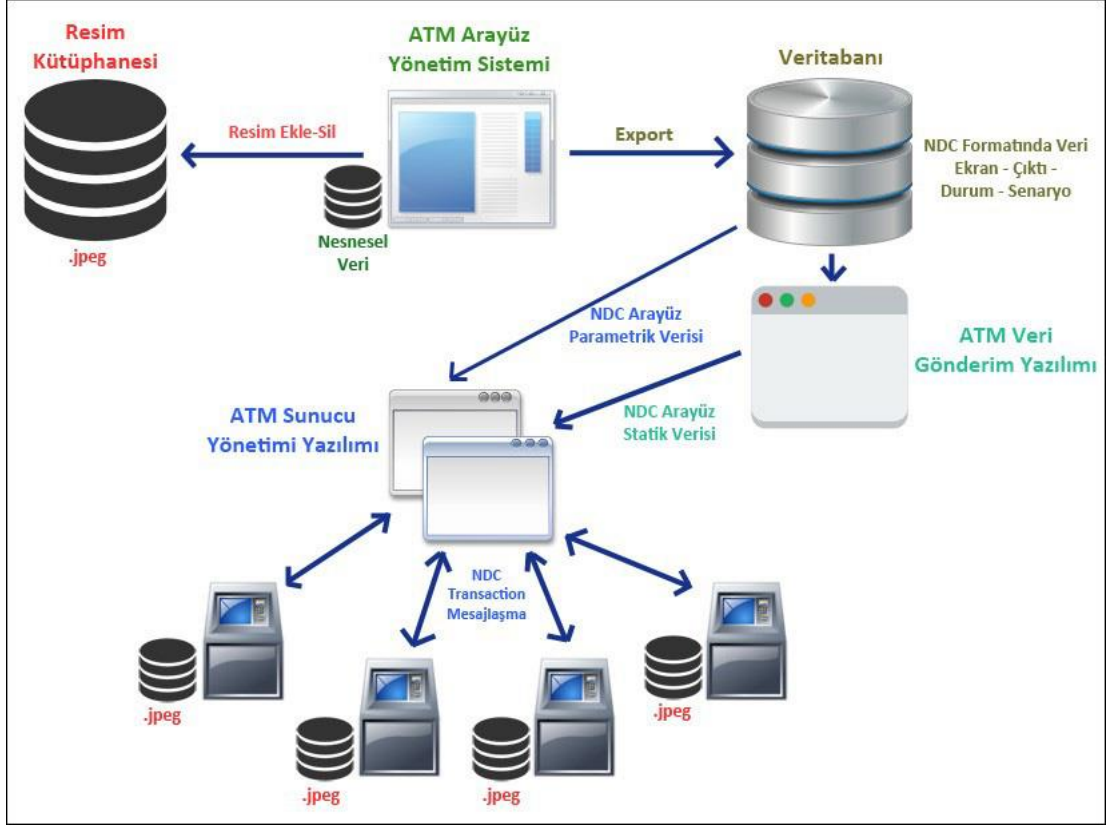
Veri akış diyagramı üzerinden de görüldüğü gibi; sistemin farklı kullanıcı modelleri farklı verileri kullanırken, bir birini takip eden bir iş akışı ile NDC

protokolünü öğrenmeye gerek kalmadan, tasarımlar ve o tasarımlara uygun akış fonksiyonları sağlanabilmektedir. Bu akış aynı zamanda bize; ATM arayüz yönetim sisteminin, ATM yazılım yönetim sistemine kolay bütünleşebilir ve bir birinden izole çalışabilen modüler bir sistem olduğunu da göstermektedir.



Şekil 5.18 : Veri akış diyagramı.

Şekil 5.19’da sistem mimarisinde görüldüğü üzere, ATM arayüz yönetim sistemi; ATM’de tutulan ekran resimlerini kendi diskinde tuttuğu gibi, ayrıca tanımlamalardan oluşan ekran, çıktı, durum ve senaryo arayüzlerini de birer nesne verileri halinde kendi veri tabanında tutar. Bu veriler ATM’ye yüklenecek NDC arayüz formatına yine ATM arayüz yönetim sistemindeki dönüştür (export) fonksiyonu ile dönüştürülür. İstenilen ATM’ye istenilen arayüz verisinin ATM veri gönderim yazılımı tarafından gönderilmesi ile birlikte tasarlanan tüm arayüzler ATM’de kullanılabilir hale gelecektir. Bununla birlikte ATM sunucu yönetimi yazılımı ise ATM’de gösterilen ekranlarda yer alan parametrik verileri, yine ATM arayüz yönetim sisteminin oluşturduğu dönüştürülmüş ekran verisinden alır ve ATM’de çıkacak ekran oluşmuş olur.



Şekil 5.19 : ATM sistem mimarisi.

## 5.7. ATM Arayüz Yönetim Sistemi Değerlendirme

Bu bölümde ATM Arayüz yönetim sistemi ile mevcutta çalışılan bankanın daha önce kullanmakta olduğu “Matrix” arayüz yönetim sistemi arasında kullanılabilirlik deney çalışması yapılmıştır. Buna göre geliştirilen yeni sistemin hipotezde de belirtildiği gibi, “ATM cihazlarında yeni nesil işlemlerin rahatlıkla tanımlanabileceği, banka personelleri tarafından en az hata ile yönetilebilen, arayüz değişimlerinde kullanım sorunlarını en aza indiren ve kullanıcı tarafından kolay kullanılabilen bir sistem” olup olmadığının karşılaştırmalı sonuçlarından bahsedilecektir.

Deney çalışması için farklı cinsiyet, yaş ve unvanlarda 8 kişi seçilmiştir. Ayrıca bu kişilerin eski ve yeni arayüz yönetim sistemleri kullanma tecrübelerinin farklı olmasına dikkat edilmiş ve daha önce hiç kullanmamış kişilere deney çalışması öncesi kısa bir eğitim verilmiştir.

Deney çalışmasına katılan kişilerin bilgileri şu şekildedir:

Tablo 5.3 : Deney çalışmasına katılanların kişi bilgileri.

No	Cinsiyet	Yaş	Görev	ATM Alan Tecrübesi	Arayüz Yönetim Sis. Tecrübesi	Matrix Tecrübesi
1	Bay	40	Yönetici	13 yıl	-	7 yıl
2	Bayan	32	Yazılımcı	6 yıl	3 yıl	1 yıl
3	Bayan	27	Yazılımcı	2,5 yıl	2,5 yıl	0,5 yıl
4	Bay	27	Yazılımcı	1,5 yıl	1 yıl	-
5	Bay	23	Yazılımcı	-	-	-
6	Bayan	39	Analist	10 yıl	1 yıl	-
7	Bay	35	Analist	8 yıl	-	-
8	Bay	24	Tasarımcı	1 yıl	-	-

Deney çalışması için seçilen kişilere iki ayrı sistemde de gerçekleştirmek üzere bir “ekran tasarlama” görevi verilmiştir. Bu tasarlayacakları ekranın ATM üzerindeki görselinin fotoğrafı çekilmiş ve kişilere dağıtılmıştır. Ekran tasarlama görevi için 2 aşamalı bir deney çalışması yapılmıştır. Buna göre;

- Tasarlanması istenilen ekran için 8 adet alt görev verilmiş ve kişilerin bu görevleri tamamlama süreleri tutulmuştur. Tasarım aşaması alt görevleri:
  - Arka plana resim koyma
  - Ekran özelliklerini tanımlama
  - Ekran buton ekleme
  - Ekran statik yazı ekleme
  - Ekran parametrik yazı ekleme
  - Ekran veri girişi ekleme
  - Ekran arama
  - Ekrandaki bir buton ve bir parametrik yazıyı değiştirme
- Tasarım aşaması bittikten sonra, oluşan ekranın ATM’ye yüklenme ve test işlemlerinin yapılması için ise 4 adet alt görev verilmiş ve kişilerin bu görevleri tamamlama süreleri tutulmuştur. Test aşaması alt görevleri:
  - Ekranı dönüştürme ve yükleme (export & load)
  - Ekranı görsel olarak test etme (PC & ATM)
  - Hatalı tasarımların düzeltilmesi ve yeniden test etme

Deneyin birinci aşamasının tamamlanma süresi sonuçları (saniye):

Tablo 5.4 : Deneyin birinci aşama sonuçları – 1.

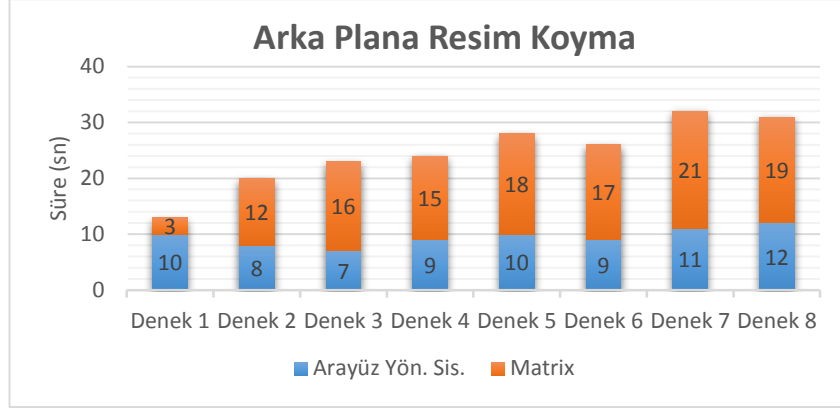
No	Denek 1		Denek 2		Denek 3		Denek 4	
	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix
1	10	3	8	12	7	16	9	15
2	15	10	13	18	11	17	14	20
3	13	12	7	16	6	17	8	19
4	6	7	5	8	4	7	6	8
5	21	29	12	48	10	50	15	62
6	7	5	5	10	4	11	5	13
7	4	12	3	18	3	14	4	21
8	25	39	18	52	15	56	21	65

Tablo 5.5 : Deneyin birinci aşama sonuçları – 2.

No	Denek 5		Denek 6		Denek 7		Denek 8	
	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix
1	10	18	9	17	11	21	12	19
2	13	22	13	18	15	19	14	23
3	10	21	9	19	10	25	12	22
4	8	10	8	9	7	11	10	12
5	21	75	18	72	22	81	21	89
6	5	12	4	14	6	14	6	15
7	4	17	3	22	5	15	5	25
8	29	61	25	63	28	70	29	72

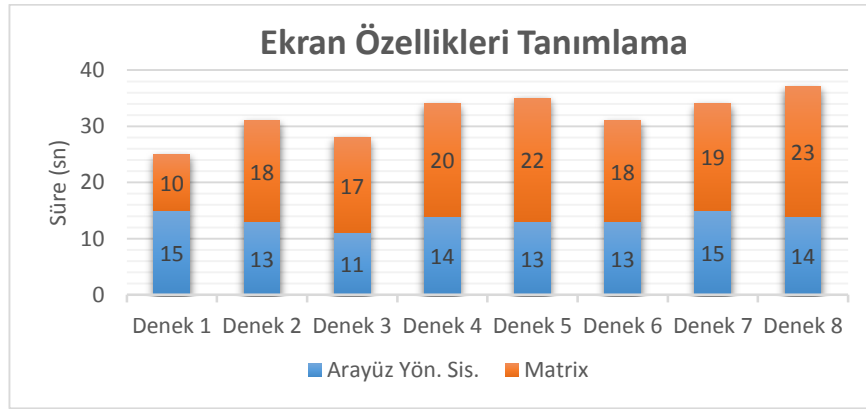
- Deneyin birinci aşamasında arka plana resim koyma görevinin; ATM arayüz yönetim sisteminde ortalama 9,5 saniyede, Matrix sisteminde ortalama 15,12 saniyede tamamlandığını görüyoruz. Yaklaşık 6 saniyelik bir işlem süresinde kısalma var. Ayrıca sistemi ilk defa kullanan bir kullanıcı için ATM arayüz yönetim sisteminde arka plana resim koyma görevi daha önce kullanmış birine göre yaklaşık aynı zamanda tamamlanırken, bu süre Matrix sistemini kullanan ve kullanmayan kullanıcılar arasında büyük fark olduğunu gösteriyor. Bu da bu işlemin öğrenme süresini ve hata yapma riskini arttırmaktadır. (Şekil 5.20)





Şekil 5.20 : Arka plana resim koyma sonuç grafiği.

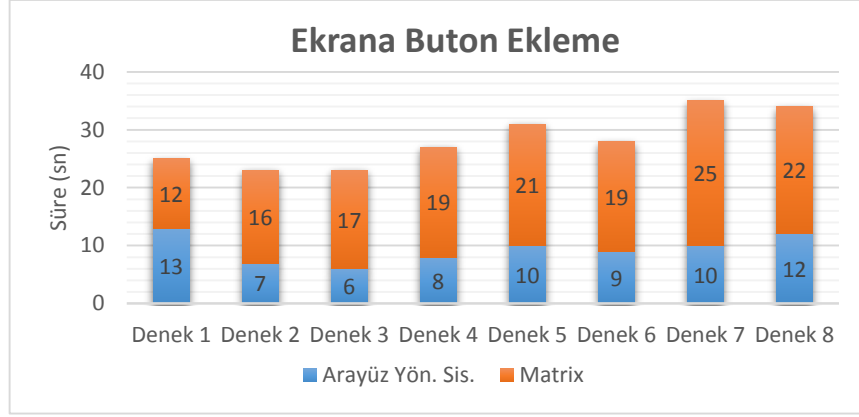
- Deneyin birinci aşamasında ekran özelliklerini tanımlama görevinin; ATM arayüz yönetim sisteminde ortalama 13,5 saniyede, Matrix sisteminde ortalama 18,37 saniyede tamamlandığını görüyoruz. Yaklaşık 5 saniyelik bir işlem süresinde kısalma var. İki sistem içinde bu görev tüm denekler tarafından ortalama bir sürede tamamlanmıştır. (Şekil 5.21)



Şekil 5.21 : Ekran özellikleri tanımlama sonuç grafiği.

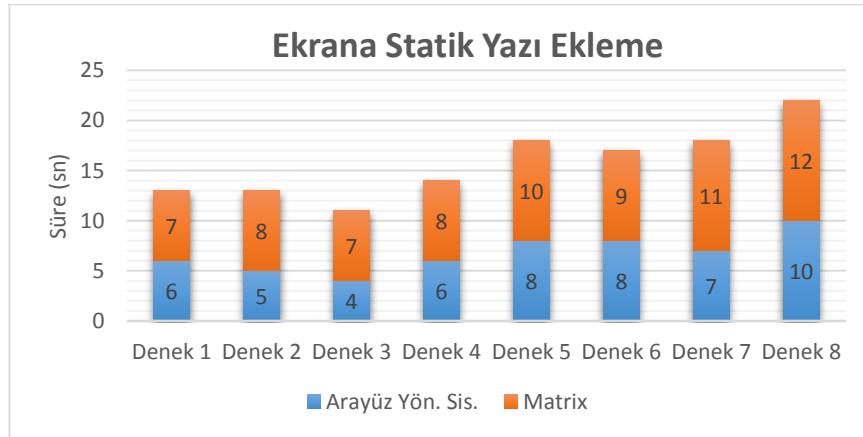
- Deneyin birinci aşamasında ekrana buton ekleme görevinin; ATM arayüz yönetim sisteminde ortalama 9,37 saniyede, Matrix sisteminde ortalama 18,87 saniyede tamamlandığını görüyoruz. Arada yaklaşık iki katlık bir fark var. Bunda ATM arayüz yönetim sisteminde butonların görsel olarak resim kütüphanesinden seçilebilme fonksiyonunun büyük payı olduğunu görüyoruz. Ayrıca Matrix sistemini daha önce kullanmayan kişilerin kullananlara göre buton ekleme görevini

tamamlama süresinde oluşan ciddi farkın, yine öğrenme süresinin uzun olduğunun bir göstergesidir. (Şekil 5.22)



Şekil 5.22 : Ekran buton ekleme sonuç grafiği.

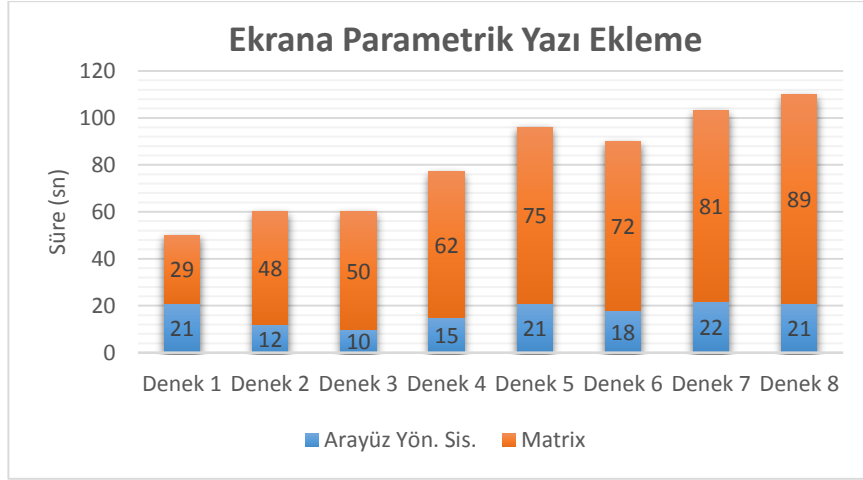
- Deneyin birinci aşamasında ekrana statik yazı ekleme görevinin; ATM arayüz yönetim sisteminde ortalama 6,75 saniyede, Matrix sisteminde ortalama 9 saniyede tamamlandığını görüyoruz. Bu görev her iki sistem içinde ortalama seyirde öğrenilebilir ve tamamlanabilir bir görev olarak göze çarpmaktadır. Bunun sebebi her iki sistem içinde statik yazıların görsel geri bildirimleri ile tasarlanıyor olmasıdır. (Şekil 5.23)



Şekil 5.23 : Ekran statik yazı ekleme sonuç grafiği.

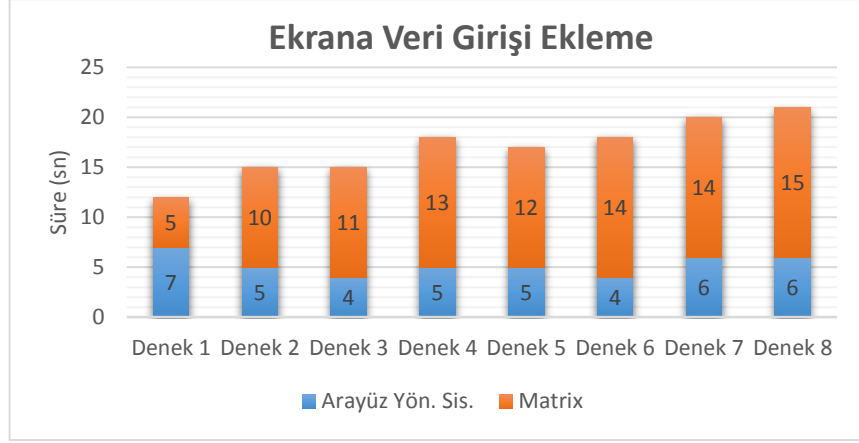
- Deneyin birinci aşamasında ekrana parametrik yazı ekleme görevinin; ATM arayüz yönetim sisteminde ortalama 17,75 saniyede, Matrix sisteminde ortalama

63,25 saniyede tamamlandığını görüyoruz. Bu deneyde aradaki yaklaşık dört kat farka, Matrix sisteminin parametrik yazılar ile normal ekran tasarımını ayrı iki ekranda yapması ve buradaki tanımlamaların bir biri üzerine giydirilmesi sebebiyle daha karmaşık bir tasarım mantığı olması sebep olmaktadır. ATM arayüz yönetim sisteminde bu zorluk ortadan kaldırılarak parametrik yazılarında tek bir tasarım ekranında ve görsel geri bildirimle tasarlanabilmesi çözümüne gidilmiş ve bu yazıların tasarımında büyük zaman kazancı sağlanmıştır. (Şekil 5.24)



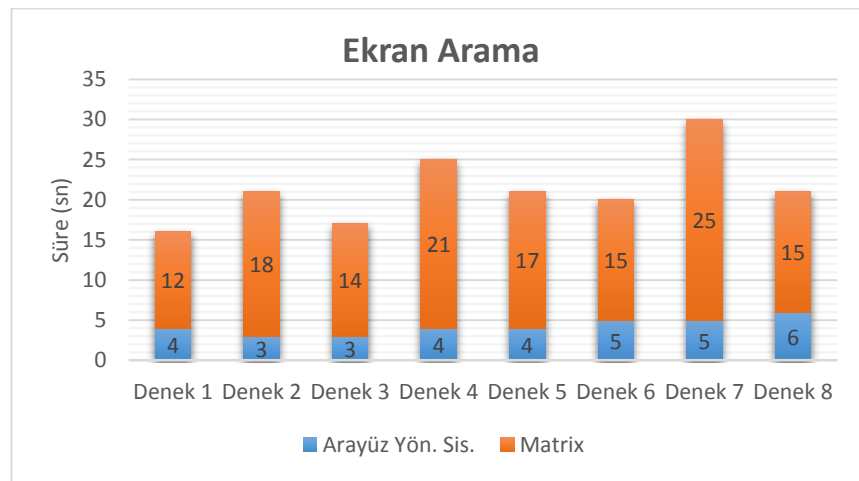
Şekil 5.24 : Ekran parametrik yazı ekleme sonuç grafiği.

- Deneyin birinci aşamasında ekrana veri girişi ekleme görevinin; ATM arayüz yönetim sisteminde ortalama 5,25 saniyede, Matrix sisteminde ortalama 11,75 saniyede tamamlandığını görüyoruz. İki sistem arasındaki fark ATM arayüz yönetim sisteminin veri girişine format tanımını yine ekran üzerinde yapabilecek kabiliyette olması sağlamıştır. Çünkü Matrix sisteminde kullanıcı, veri girişi formatı tanımlamak için NDC dokümanına bakmak zorunda bırakılmıştır. Bu çözüm veri girişi ekleme görevine yeni sistemle birlikte yarı yarıya zaman kazancı sağlamıştır. (Şekil 5.25)



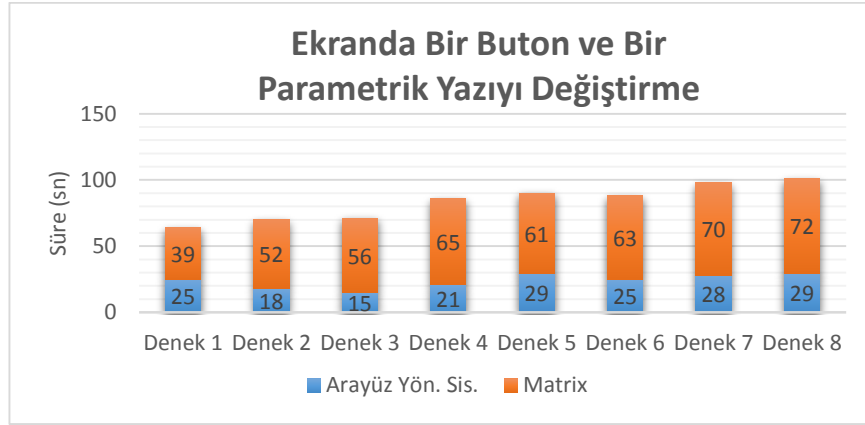
Şekil 5.25 : Ekran veri girişi ekleme sonuç grafiği.

- Deneyin birinci aşamasında ekran arama görevinin; ATM arayüz yönetim sisteminde ortalama 3,87 saniyede, Matrix sisteminde ortalama 18 saniyede tamamlandığını görüyoruz. Matrix sisteminde ekran arama fonksiyonu sadece ekran listesinden gözle aranılan ekranı bulma şeklinde yapılmaktadır. Bu sebeple grafikten de görüleceği gibi kişiler arasında gözle bulma zamanları arasında dalgalanma oluşmuştur. Aradığı ekranı önce bulanlar için süre kısalmış, bulamayanlar için süre epey uzamıştır. Buna karşılık ATM arayüz yönetim sisteminde gelişmiş bir ekran arama fonksiyonu vardır. Buna göre ekran numarasından, adından veya açıklamasından arama çubuğu yardımıyla ekran bulunabilmektedir. Bu fonksiyon sayesinde aranılan ekran çok kısa sürede bulunmuş ve bu görev için süreler çok düşük çıkmıştır. (Şekil 5.26)



Şekil 5.26 : Ekran arama sonuç grafiği.

- Deneyin birinci aşamasında ekrandaki bir buton ve bir parametrik yazıyı değiştirme görevinin; ATM arayüz yönetim sisteminde ortalama 23,75 saniyede, Matrix sisteminde ortalama 59,75 saniyede tamamlandığını görüyoruz. Bu aradaki farkın fazla olmasının sebebi Matrix sisteminin parametrik yazıyı eklemedeki kötü tasarım şekli olarak karşımıza çıkmaktadır. Bu da gösteriyor ki her hangi bir yanlış tasarımda değişiklik yapılacak olması ATM arayüz yönetim sisteminde hızlı ve kolay aksiyon alınabilmesi anlamına gelmektedir. (Şekil 5.27)



Şekil 5.27 : Ekran değişikliği sonuç grafiği.

Deneyin ikinci aşamasının tamamlanma süresi sonuçları (saniye):

Tablo 5.6 : Deneyin ikinci aşama sonuçları – 1.

No	Denek 1		Denek 2		Denek 3		Denek 4	
	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix
1	3 & 5	- & 15	2 & 4	- & 20	2 & 3	- & 23	2 & 4	- & 29
2	0 & 10	- & 8	0 & 12	- & 11	0 & 8	- & 13	0 & 13	- & 15
3	32	182	0	318	0	287	0	486

Tablo 5.7 : Deneyin ikinci aşama sonuçları – 2.

No	Denek 5		Denek 6		Denek 7		Denek 8	
	No	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix	Arayüz Yön. S.	Matrix	Arayüz Yön. S.
1	3 & 4	- & 35	2 & 4	- & 32	5 & 7	- & 40	4 & 6	- & 36
2	0 & 15	- & 17	0 & 9	- & 14	0 & 11	- & 16	0 & 17	- & 21
3	0	513	0	324	0	567	44	465

Deneyin ikinci aşaması denekler için hazırlanan tasarımı kendi bilgisayarlarında ve ATM üzerinde test etme aşamasıydı. Buna göre;

Birinci görev olan ekranı dönüştürme ve yükleme işlemleri için; Matrix sistemi dönüştürme fonksiyonunu sağlamadığından ve işlemin yükleme fonksiyonu ile birlikte karmaşık bir işlem adımıyla yapılması ortalama süreyi 28,75 saniyeye çıkarmıştır. ATM arayüz yönetim sisteminde ise bu işlem, dönüştürme ve yükleme fonksiyonu ayrı şekilde tasarlanmıştır. Buna göre sadece dönüştürme işlemi ile tasarımın NDC formatına dönüşmüş halinin ATM'ye yüklenmeden bile kontrol edilebilmesi sağlanmıştır. İki işlemi ayrı ayrı düşündüğümüzde ise dönüştürme işleminin ortalama 2,87 ve yükleme işleminin ortalama 4,62 gibi kısa bir sürede yapıldığını görürüz. Bu iki işlemin toplam süresinin bile Matrix sisteminin 28,75'lik tek işlem süresinden oldukça kısa olduğu görülür.

İkinci görev olan ekranın görsel olarak hem tasarımcıların bilgisayarlarında hem de ATM'de test etme işlemi için; Matrix sisteminin arayüz ekranlarından da görüldüğü gibi tasarımın resim ve yazı verileri görülmeden yapıldığı bilinmektedir. Bu da kullanıcıların tasarımı kendi bilgisayarlarında görerek tasarlama imkânını ortadan kaldırır. Bu özellik ATM arayüz sisteminde, ATM'de görülen ekran boyutu ile bire bir aynı olacak şekilde sağlandığından, ekran sürükleyip bırak yöntemiyle görerek tasarlanmış olur. Bu da ekranın ATM'ye atıldığında çoğunlukla istenen şekilde olacağını garanti eder. Deneyimizdeki süreler baktığımızda ise; ATM arayüz yönetim sistemiyle tasarlanan bir ekranın, ATM'de test edilme süresi ortalama 11,87 saniye olurken, Matrix sisteminde bu süre ortalama 14,37 saniyedir. Fakat asıl önemli fark üçüncü görevde karşımıza çıkmaktadır. Üçüncü görev olan hatalı tasarımların düzeltilmesi ve yeniden test etme işlemine göre; ATM arayüz yönetim sistemiyle tasarlanan tasarımların 8 denekten sadece 2'si tarafından hatalı tasarlandığı, 6'sı tarafından ise sonucun istenildiği gibi olduğu görülür. 2'sinin de hatalardan ortalama dönüş süresi 38 saniyedir. Buna karşın Matrix sistemi ile tasarlanan ekranlarda ise ilk seferde doğru tasarlayan hiçbir denek bulunmazken, hatalı tasarımların geri dönüş süresi tüm denekler için ortalama 392,75 saniye yani yaklaşık 6,5 dakikadan fazla sürmüştür.

Sonuç olarak sürelerden de anlaşıldığı gibi resim ve yazı verilerini görmeden sadece tanımsal olarak tasarlanan ekranların hiç biri ilk defada doğru tasarıma

ulařmamakta ve dzenlemelerin ardından tekrardan test etme iřlemleri ile bir ekranın tasarımının son halini alması ok uzamaktadır. Fakat ATM arayz ynetim sistemi gibi bir ara sayesinde tasarımlar bire bir grlerek daha etkin, kolay ve kısa srede dođru sonulara ulařmaktadır. Hemen hemen her gn bir ekranın deđiřikliđe uđradıđı veya yeni ekranların tasarlandıđı bilinen ATM gibi bir sistem iin, arayz ynetim sisteminin kullanılabilirliđi ne denli arttırdıđı daha net anlařılmaktadır. Ayrıca bu sistemin tasarımcılara getirdiđi standartlar sayesinde ekranlar daha tutarlı ve kullanıcı dostu olmaktadır.

Deneklere verilen grevler arasında sadece ekran tasarlama grevi yer alırken, aynı durum ıktı, durum ve senaryo tasarımı iinde geerlidir. ATM arayz ynetim sistemi ıktı, durum ve senaryo tasarımında da tasarıma getirdiđi kolaylıklarla, ATM arayzlerinin geliřtirilebilir ve ynetilebilir olmasına katkıda bulunmuřtur.

## 6. ATM ARAYÜZ PERFORMANSI

Bölüm 3.2.'de arayüz tasarımında performans ölçümünün günümüz sistemlerinde genelde tercih edilmediğinin, fakat sistem kullanılabilirliğinin bir parçası olduğundan bahsetmiştik. Büyük sistemlerde arayüz performansının, aslında genel sistem performansını da etkilediği ve kullanıcı memnuniyetsizliğinin yanında sistem problemleri de yarattığı bilinmektedir.

ATM gibi yoğun işlem hacmi içeren büyük bir sistem içinde; tez kapsamın çalışılan banka üzerinde yapılan araştırmalar sonucu arayüz performansının değerlendirilmediği ve genelde performans problemlerine donanımsal desteklerle çözüm getirildiği görülmektedir. Bunda kuşkusuz arayüzün tasarımındaki zorluklar ve net bir performans metriğinin çıkarılamamış olması sebep olmaktadır.

Tasarımdaki zorlukların, çalışma kapsamında "ATM Arayüz Yönetim Sistemi" sayesinde giderildiği düşünüldüğünde, sistemin arayüz performans problemi üzerine de bir çözüm önerisinin olması kuşkusuz sistemi daha da önemli hale getirecektir. Bu sebeple ATM arayüz yönetim sistemi içinde yaratılan ve bölüm 5.5.'de anlatılan "senaryo" kavramı ve tasarımları, performans ölçümlemesinde metrik olarak kullanılmıştır. Buna göre ATM üzerinde para çekme, bakiye sorma, hesap listeleme vb. gibi senaryo veya alt senaryo tasarımları üzerinde yapılan analizler ve bu analizler sonucunda bir performans metriği ortaya çıkarılmıştır.

Senaryoyu belirli bir amaca hizmet etmek amacıyla tasarlanmış, en az bir başlangıç ve bir bitiş düğümü olan durumlar bütünü olarak önceki bölümlerde tanımlamıştık. Bunun yanında senaryo tasarım bileşenleri olan durum, alt senaryo ve işlem mesajı bileşenlerinden de bahsetmiştik. Bu bölümde ise; ATM arayüz yönetim sisteminin senaryo tasarımı verilerini temel alarak, ATM arayüz performansını ölçmede kullandığımız performans metriklerinden bahsedeceğiz. Her bir performans metriği üzerine yapılan ölçümlerin sonuçları ise ATM arayüz tasarımındaki yanlışları bize gösterirken, tasarımda yapılacak değişikliklerle performansın ve kullanılabilirliğin nasıl arttığını göstereceğiz.

Yapılan arayüz performans analizlerde kullanılan veriler; SAN-TEZ kapsamında çalışılmakta olan Cybersoft firmasının müşterisi olan ŞEKERBANK ATM tasarımları



verileridir. Bankanın ATM arayüzlerinin tamamı, beşinci bölümde karşılaştırmalı olarak anlatılan Matrix sisteminden devir alınmış olup, 2013 yılında tez kapsamında yazılan ATM arayüz yönetim sistemine geçirilmiştir. Geçen süre zarfında ise tüm yeni tasarımlar ATM arayüz yönetim sisteminde tasarlanmış olup, eski tasarımlardan oluşan tasarım hataları üzerine yapılan analizlerden çıkan sonuçlara göre de performans problemleri yaratan tasarımlar değiştirilerek gerçek ortamda test edilmiştir.

## **6.1. ATM Arayüz Performans Metriği ve Analizi**

Tez kapsamında ortaya çıkarılan ve sonuçları analiz edilen ATM arayüz performans metrikleri şu şekildedir;

- Ortalama senaryo bileşen sayıları tespiti
- Tekrar eden senaryo tespiti
- Senaryo tamamlanma süreleri tespiti
- Senaryo başarılı - başarısız tamamlanma oranları
- Zaman aşımına uğrayan senaryo tespiti

### **6.1.1. Ortalama Senaryo Bileşen Sayıları Tespiti**

Bu metriğe göre senaryo tasarımındaki durum, işlem mesajı ve alt senaryo bileşenleri sayılmış ve bunun üzerine analizler yapılmıştır. Bir senaryonun bileşen sayısının ortalamanın üzerinde olması, aynı zamanda o senaryo akışında kullanıcıya sorulan soruların fazlalığı veya tekrar eden akışın varlığı anlamını taşır. Bu anlam üzerine yapılacak analizler ise tasarımdaki yanlışlıkları gözler önüne serecektir.

Ortalama senaryo bileşen sayıları metriği analiz edilirken 9 farklı senaryo akışı seçilmiştir. Buna göre seçilen senaryo akışlarının bileşen sayıları hesaplanmış ve bu senaryoların işlem başarılı tamamlanma ve hata oranları çıkarılmıştır. Ardından senaryo bileşen sayılarının ortalama ve standart sapma değerleri bulunup, standart sapmanın uç değerine yakın bileşen sayısına sahip senaryoların tasarımları tekrardan

gözden geçirilmiştir. Bu senaryolar üzerine yapılan tasarım değişiklikleri sırasında ise tekrardan işlemin başarılı tamamlanma ve hata oranları çıkarılmış, tasarım değişikliklerinin hata oranlarına etkisi gözlenmiştir. Standart sapmanın uç değerinin üstünde senaryo bileşen sayısına sahip olan senaryoların, kullanıcının işlemi tamamlama süresini arttıracak gibi bilindiği gibi sistemin kullanılabilirliğini de düşürdüğü gözlenmiştir. Bu sebeple yapılan tasarım değişikliklerinde kullanıcıya sorulacak az ve öz soruyla, kullanıcının senaryoyu en kısa sürede tamamlanması ve kullanılabilirliğin artırılması hedeflenmiştir.

Senaryo tasarımıda değişiklik olmadan önceki senaryo bileşen verileri;

Tablo 6.1 : Senaryo bileşen sayıları (değişiklik öncesi).

Senaryo İsmi	Durum Bileşeni	İşlem Mesajı Bileşeni	Alt Senaryo Bileşeni	Hata Oranı	Başarılı Tamamlanma Oranı
Ana Menü (6 Farklı Kart Türünde)	30	6	0	2%	95%
Bakiye Sorgu (Hesap Listelemeli)	18	3	1	7%	91%
Havale (Hesap Listelemeli)	45	5	0	18%	77%
Hesaba ve Kredi Kartına Para Yatırma (Hesap Listelemeli)	68	9	2	35%	61%
Para Çekme (Hesap Listelemeli)	47	5	1	14%	66%
Şifre Değiştirme	13	2	2	4%	92%
Ortalama	36,83	5,00	1,00	%13,33	80,33%
Standart Sapma	20,55	2,45	0,89	%12,23	14,53%
Uç Değer	57,38	7,45	1,89	%25,56	65,81%

Senaryo tasarımıda değişiklik olmadan önceki bileşen sayılarını inceleyecek olursak, ilk olarak ana menü senaryosunun 6 farklı kart türünde tekrar ettiği ve her biri için durum yönlendirmesinin ayrı ayrı yapıldığı durum sayısının fazlalığından görülmektedir. Hâlbuki ana menü senaryosu bir yönlendirme senaryosudur ve her ana menü bir alt senaryo olarak tasarlanabilir. Bu şekilde her menüden alt

senaryonun başlangıç durumu çağırarak yerine alt senaryo çağrılarak ana menü senaryosundaki durum sayısı azaltılabilir. Bu aynı zamanda para çekme, bakiye sorma gibi alt senaryoların bir kere tasarlanması ve bu tasarımların ana menü operasyonundan alt senaryo olarak çağırılmasına olanak sunar. Bu da senaryo tasarımının tekrar edilme problemine çözüm sunar. Bu durum sadece tasarımsal probleme çözüm getirdiğinden hata oranlarına bir etkisi gözlenmemiştir.

Tasarımda modülerlik ile birlikte, senaryolara ara senaryo kavramının getirildiğinden bahsetmiştik. Senaryo bileşen sayılarından da görüldüğü üzere birçok işlem hesap listeleme akışı sebebiyle durum bileşeni sayısını arttırmaktadır. Bu durum hem senaryonun uzamasına hem de hesap listeleme akışında yapılacak bir değişikliğin tüm senaryo tasarımında yapılması gerekliliğine sebep olmaktadır. Bunun önüne geçmek için hesap listeleme içeren senaryo tasarımlarından hesap listeleme akışı çıkarılmış ve bu akış hesap listeleme adında bir ara akışta toplanmıştır. Bu sayede hesap listeleme akışının tasarımlarda tekrar edilmesinin önüne geçilmiş, diğer akışların senaryo bileşen sayılarının da azalması sağlanmıştır. Bu durum sadece tasarımsal probleme çözüm getirdiğinden hata oranlarına bir etkisi gözlenmemiştir.

Bileşen sayılarında uç değerün üstünde bileşene sahip olan hesaba ve kredi kartına para yatırma senaryosu tasarımı üzerinde de çalışma yapılmıştır. Buna göre ana menüden para yatırma butonu ile başlayan senaryonun önce hesap veya kredi kartı seçimi daha sonra yatırılacak hesap veya kredi kartı numarası seçimi gibi ardı ardına çok fazla soruyla tamamlandığı gözlenmiştir. Bu şekilde yapılan bir tasarımın, aslında iki farklı iş olmasına rağmen tek bir tasarımda toplanması, hem kullanıcının işlemi başarılı tamamlama oranını düşürmekte hem de arayüzde çok fazla soru ile kullanıcıyı yormaktadır. Bu problemler düşünülerek yapılan çalışmayla hesaba para yatırma ve kredi kartı borç ödeme işlemi bir birinden ayrılmış ve izole iki ayrı senaryo tasarımına kavuşturulmuştur. İki ayrı senaryo ile birlikte bileşen sayılarında da gözle görülür bir azalma meydana gelirken, bir işlemdeki probleminden diğer işlemin etkilenmemesi de sağlanmıştır. Ayrıca tasarımdaki bu değişiklik ile akışın işlem mesajı sayısı azalmış ve sunucuya gelen mesaj trafiği de bir nebze olsun rahatlatılmıştır.

ATM'ler de en fazla işlem gören senaryo para çekme senaryosudur. (Günlük işlem adetleri banka gizliliği sebebiyle sayı olarak verilememiştir.) İşlem sayısının çok olması ve senaryo tasarımındaki bileşen sayısının uç değere yakınlığı sebebiyle, para

çekme senaryosunun tasarımı da gözden geçirilmiştir. Buna göre hesap listelemenin ara senaryo olarak tasarlanmasıyla azalan bileşen sayısı dışında akıştan azaltılabilecek bir durum gözlenmemiştir. Fakat para çekme işlemini süre bakımından kısaltacak, işlemini hızlı yapmak zorunda olan kullanıcılar için, sorularla zaman kaybettirmeden işlemi tamamlamasına olanak sunacak yeni bir senaryo tasarımı yapılmıştır. Hızlı para çekme adındaki bu senaryo tasarımına göre, para çekme senaryosunda bulunan hesap ve tutar seçme işlemleri kaldırılmış ve tek tuşla kullanıcının daha önceden belirlediği tutarı çekmesine olanak sunulmuştur. (tanımlamayı yine ATM üzerindeki başka bir operasyonla yapabilir) Bu tarz bir operasyonla bileşen sayısı, para çekme operasyonuna göre yarı yarıya azaltılmış ve işlemin başarılı tamamlanma oranı %17 civarında artmıştır. Başarı oranındaki bu artış ile kullanıcılar için bu operasyonun, sistemin kullanılabilirliğini arttıran memnun edici bir çalışma olduğu görülmektedir.

Senaryo tasarımıda değişiklik olduktan sonraki senaryo bileşen verileri;

Tablo 6.2 : Senaryo bileşen sayıları (değişiklik sonrası).

Senaryo İsmi	Durum Bileşeni	İşlem Mesajı Bileşeni	Alt Senaryo Bileşeni	Hata Oranı	Başarılı Tamamlanma Oranı
Ana Menü	5	1	9	2%	95%
Bakiye Sorgu	7	2	1	7%	91%
Havale	45	5	0	18%	77%
Hesap Listeleme (Ara)	11	1	0	3%	94%
Hızlı Para Çekme	16	2	0	8%	83%
Kredi Kartı Borç Ödeme	35	4	1	21%	74%
Para Çekme	36	4	1	14%	66%
Para Yatırma	18	4	1	17%	75%
Şifre Değiştirme	13	2	2	4%	92%
Ortalama	20,67	3,00	1,67	10,44%	83,00%
Standart Sapma	14,34	1,80	2,83	7,16%	10,49%
Uç Değer	35,01	4,80	4,50	17,61%	72,51%

Senaryo bileşen sayısı tespiti çalışmasıyla, tasarımda düzeltilen yanlışların olmasının yanında yanarda düzeltilemeyen durumlarda olmuştur. Buna göre havale

senaryosuna bakacak olursak, 45 durum deęişkeninin uç deęerin üzerinde bir sayıda olduęu görülür. Bu sayıyı azaltmak için, senaryo içindeki hesap listeleme akışının aslında alt senaryoya aktarılması ve tekrar eden durumların yaratılmaması gerekiyordu. Fakat bankanın talepleri doğrultusunda havale ekranlarında kullanıcıya gösterilen ekran mesajının farklılaşması istendiğinden, bu probleme çözüm getirilememiştir.

Sonuç olarak senaryoların ortalama bileşen sayıları metrięi hesaplamaları ve üzerinde yapılan analizler, bize tasarımlardaki yanlışları ve aynı zamanda arayüz kaynaklı performans kayıplarını gösterirken, tasarım hatalarının giderilmesiyle de işlemin başarılı tamamlanma oranların arttığı görülmektedir.

### 6.1.2. Tekrar Eden Senaryo Tespiti

Senaryo tasarımı aracı sayesinde gelen modülerlik ile senaryoların bir başka senaryo içinde alt senaryo olarak kullanılabilidiğinden bahsetmiştim. Yapılan ölçümler sonucunda hesap listeleme, makbuz sorgu ve kurum listeleme akışının diğer senaryolarda aynı şekilde kullanıldığı ve bu senaryoların her birinde tekrar eden tasarımların olduğu görülmüştür. Bir senaryoyu bu şekilde tasarlamak tasarımın bileşen sayısını arttırdığı gibi, bu akışta yapılacak her hangi bir deęişikliği hepsinde ayrı ayrı yapmak zorunda olmayı gerektirmektedir.

Tablo 6.3 : Tekrar eden senaryo sayıları.

Senaryo İsmi	Toplam Bileşen Sayısı	Diğer Senaryolarda Kullanılma Sayısı
Hesap Listeleme	15	12
Makbuz Soru	8	17
Kurum Listeleme	18	2
Şube Listeleme	16	2
Toplam	57	33

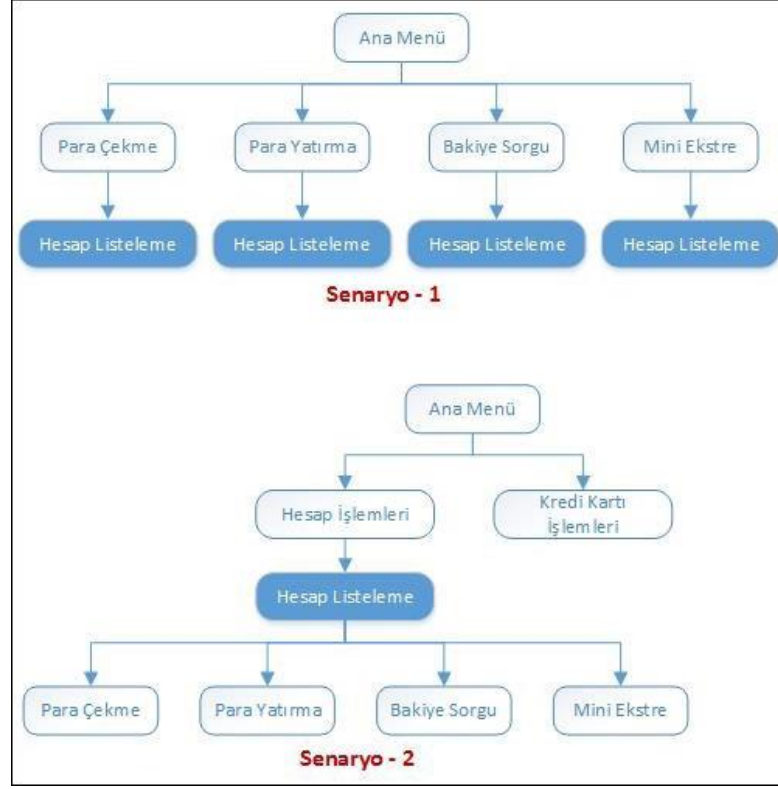
Tablo 6.3'de görüldüğü gibi dört farklı ara senaryonun diğer senaryolarda toplam tekrar etme sayısı 33'dür. Hâlbuki bu senaryolarda bu akışlar ara senaryo olarak tasarlanmış olsaydı,  $33 \times 57 = 1881$  tane bileşen kullanmak yerine sadece 57

tane bileşen kullanılacak ve 33 tanede ara senaryo bileşeni ile toplam 90 bileşen kullanılmış olacaktır. Senaryo tasarımının getirdiği modülerlik ile tekrar eden senaryo tasarımları kaldırılarak, ara senaryoda tasarlanması çözümüne gidilmiştir. Bu sayede hem tekrar eden senaryo tasarımının önüne geçilmiş hem de tüm senaryolarda kullanılan ortak akışlar arasında bir standart sağlanmıştır.

Yapılan bu çalışmanın ATM kullanıcı açısından işlemlerdeki başarı oranını arttırdığı gözlenmemiş olup, tasarımcıya sistemi yönetebilme ve hızlı değiştirebilme imkânı sunmuştur.

Tekrar eden senaryo tespiti metriği üzerinden ayrıca ikinci bir çalışma olarak sistemsel performans yükü çalışması da yapılmıştır. Buna göre ATM gibi işlem hacmi çok büyük bir sistemin, her işlemde hesap listeleme gibi bir işlem için sunucuya gelmesi sisteme oldukça büyük bir yük getirmektedir. Fakat bu yük arayüzde yapılacak birkaç tasarım değişikliğiyle aslında azaltılabilir. Örnek olarak hesap listeleme ara senaryosunu alırsak, bu ara senaryonun para çekme-yatırma gibi işlemler için her seferinde kullanıcıya hesap listelemekte ve seçilen hesap üzerinden işlem yapılmaktadır. Bu aslında iki işlem için iki defa hesap listeleme mesajının sunucuya gelmesi demektir. Hâlbuki kullanıcıya ana menüde işlem seçimi yaptırmadan önce, kredi kartı işlemleri veya hesap işlemleri gibi bir seçenek sunulduğunda, hesap işlemlerini seçen kullanıcı için hesap listeleme bir defa yapılacak ve bundan sonra seçeceği her işlem seçtiği bu hesap üzerinden yapılabilir.

Şekil 6.1 örneğinde görüldüğü gibi Senaryo-1 tasarımına göre hesap listeleme 4 defa yapılıyorken, Senaryo-2 tasarımına göre 1 defa yapılır. Her kullanıcının ATM’de bir gelişte ortalama 3-4 işlem yaptığı düşünülüğünde, bir kullanıcı için bile sunucuya gelen sadece hesap listeleme kaynaklı mesaj sayısı 2-3 azaltacaktır. Bu da yüzlerce kullanıcı ve yüzlerce ATM olduğu düşünülüğünde aslında büyük bir işlem hacmi kazancı demektir.



Şekil 6.1 : Hesap listeleme tasarımı.

Yapılan bu çalışmanın sonucunu gerçek ortamda deneme şansımız olmamıştır. Fakat sayısal verilerle işlem hacmi kazanımını belirtmek gerekirse;

- Kullanıcıların yarısı her bir kart takışında ortalama 2 işlem yapıyor,
- ATM’de yapılan günlük işlem sayısı ortalama 50.000,
- ATM’de hesap listeleme içeren işlem sayısı ortalama %50,

Bu verilerle ortalama günde 6250 işlem sayısının azalacağı sonucu çıkar. Sadece tasarımda yapılacak bir değişiklik ile sunucuya gelen işlem sayısının %12,5 oranında azalacağı bir sistem düşünüldüğünde, böyle bir çalışmanın arayüz performansı metriği sayesinde sisteme ne denli katkı sağladığı açıkça görülmektedir.

### 6.1.3. Senaryo Tamamlanma Süreleri Tespiti

Kullanıcı memnuniyetinin sağlanmasındaki temel faktörlerin başında arayüzün kullanıcı dostu ve kolay kullanılabilir olması gerekliliği bilinmektedir. ATM arayüz

yönetim sistemi senaryo tasarımı aracı sayesinde de artık senaryolar tasarimsal olarak analiz edilebilir duruma gelmiştir. Bu sebeple yapılan çalışma ile senaryoların tamamlanma süreleri metriği ortaya çıkarılmış ve süreler üzerine durum analizleri yapılmıştır. Bir senaryonun kullanıcı tarafından ne kadar sürede tamamlandığı aslında arayüz performansının değerlendirilmesinde en önemli ölçütlerden biridir. Çünkü bir işlemin ortalama bir süreden fazla sürmesi, hem kullanıcının hata yapma olasılığını arttırmakta hem de tasarimsal bir takım yanlışların habercisi olmaktadır.

Senaryo tasarımı bileşenleri temel alınarak işlemlerin ortalama tamamlanma süreleri istatistiği çıkarılmış ve sonuçlar şu şekilde oluşmuştur:

Tablo 6.4 : Senaryo tamamlanma süreleri (değişiklik öncesi).

Senaryo İsmi	Tamamlanma Süresi (saniye)
Ana Menü	3
Bakiye Sorgu	11
Fatura Ödeme	61
Havale	35
Hızlı Para Çekme	7
Kredi Başvuru	52
Kredi Kartı Borç Ödeme	47
Para Çekme	26
Para Yatırma	32
Şifre Değiştirme	18
Ortalama	29,20
Standart Sapma	19,82
Uç Değer	49,02

İşlemlerin tamamlanma sürelerini inceleyecek olursak; fatura ödeme ve kredi başvuru senaryolarının uç değer üstünde bir tamamlanma süresine sahip olduğu görülmektedir. Bu sonuca göre bu iki senaryo tasarımı tekrardan gözden geçirilmiş ve oluşan süre fazlalığının nereden kaynaklanabileceği araştırması yapılmıştır.

Senaryolardan fatura ödemedeki süre fazlalığı işlem istatistiğinden incelenmiş olup, kullanıcının bulunduğu ildeki kurumu listeden bulup kodunu girme ve fatura bilgisini girme işlemlerinde çok zaman harcadığı tespit edilmiştir. Bu tespit tasarımı kullanıcılar kurum kodu sormanın aslında çok da doğru bir tasarım olmadığını



göstermiştir. Bu sonuca göre tasarım değişikliğine gidilmiş ve kullanıcıya önce fatura tipi seçtirilmiş (elektrik, su, doğalgaz vb.), ardından ATM'nin bulunduğu konuma göre o ilçenin seçilen fatura tipindeki kurumu otomatik gösterilmiştir. Kullanıcıların çoğunun evine en yakın ATM'den faturalarını ödeyecekleri düşünüldüğünde, bu çözümün çoğunlukla kullanıcıyı kod girme zahmetinden kurtaracak bir çözüm olduğu söylenebilir. Tablo 6.5'de de görüldüğü gibi tasarım değişikliği çözümünün ardından fatura ödeme işleminin tamamlanma süresi azalarak 37 saniyelere düşmüştür.

Tablo 6.5 : Senaryo tamamlanma süreleri (değişiklik sonrası).

Senaryo İsmi	Tamamlanma Süresi (saniye)
Fatura Ödeme	37
Kredi Başvuru	32

Yine aynı şekilde kredi başvuru senaryosundaki süre fazlalığı işlem istatistiğinden incelenmiş olup, kullanıcının cep telefonu numarası ve kimlik bilgileri girme işlemlerinde zaman kaybettiği tespit edilmiştir. Bu süreyi kısaltmak için de tasarımda kullanıcıya kendi adına kredi başvuru veya başkası adına kredi başvuru seçeneği sunulmuştur. Burada kullanıcının kendi adına kredi başvurusu seçmesi durumunda, bilgilerinin müşteri kayıtlarından alınabileceği varsayılmış ve bu durumda kullanıcıya fazladan soru sorma durumu ortadan kaldırılmıştır. Kredi başvuru işlemlerinin çoğunun kendi adına kredi başvurusu olduğu düşünüldüğünde, sadece basit bir tasarım değişikliğiyle bu işlemin de tamamlanma süresinin azalacağı söylenebilir. Tablo 6.5'de de görüldüğü gibi tasarım değişikliği çözümünün ardından kredi başvuru işleminin tamamlanma süresi azalarak 32 saniyelere düşmüştür.

Sonuç olarak ATM yönetim sisteminin senaryo tasarımı ekranı verileri sayesinde, kullanıcıların hangi işlemleri ne kadar sürede tamamladığı ve hangi sorular yüzünden zaman kaybettikleri kolaylıkla tespit edilebilmiştir. Bu da tasarımdaki yanlışlıkları ortaya koyarken, tasarımda yapılan değişiklikler ile kullanıcıların işlemleri tamamlama sürelerinde ciddi azalmalar meydana gelmiştir. İşlem tamamlanma sürelerindeki düşüş, kullanıcıların hata yapma olasılığının da azaldığının bir göstergesi olurken, sistemin kullanılabilirliğini de arttıran önemli bir faktör olmuştur.

#### 6.1.4. Senaryo Başarılı – Başarısız Tamamlanma Oranları

Günümüzde mevcut ATM sistemlerinde senaryo tasarımı diye bir kavram bulunmazken hangi işlemlerin başarılı, hangi işlemlerin başarısız olduğunu tespit etmekte mümkün olmamaktadır. Sistemler işlem tiplerine özel verilen hata kodlarından sadece hata alan işlemleri tespit edebilirken, kullanıcı davranışları üzerine analizler yapmaya da olanak sağlamamaktadır.

Çalışma kapsamında bu eksiklik ele alınmış olup, senaryo tasarımlarının bitiş düğümlerine başarılı veya başarısız olma özelliği getirilmiştir. Buna göre senaryolar şu şekilde bitebilir:

- Kullanıcı işlemi baştan sona başarılı tamamlamış.

Örneğin, bir para çekme senaryosunda; kullanıcı hesap listelemiş, tutarı girmiş, parasını ve makbuzunu almış, ana menüye dönmüş.

- Kullanıcı işlemi başarılı tamamlamış ama işlem yarıda kalmış. (Bilgilendirme senaryoları)

Örneğin, kullanıcı bakiye sormuş, bakiyesini görmüş ama makbuz almadan işlemi yarıda kesmiş.

- Kullanıcı işlemi tamamlayamamış, istenmeyen bir problemle karşılaşmış.

Örneğin, kullanıcı para çekmek istemiş, para ünitesi arızası veya ATM’de para kalmama gibi bir durumla karşılaşmış.

- Kullanıcı işlemi tamamlayamamış, fakat bu kullanıcının beklediği bir durum.

Örneğin, kullanıcı para çekmek istemiş, fakat hesabında yeterli miktarda para kalmamış.

Senaryolar üzerinde yapılan başarılı-başarısız özellik tasarımları, kullanıcı davranışlarından çıkan istatistiki verilerle birlikte analiz edildiğinde, bazı senaryoların kullanıcı tarafından çoğunlukla yarıda kesildiği, bazı senaryoların ise başarısız sonuçlanma oranlarının yüksek olduğu tespit edilmiştir. Bu sebeple öncelikle yarıda

kesilen işlemler incelenmiştir. Buna göre yarıda kesilen işlemlerin %50'si makbuz ücreti ödemek istemeyen kullanıcıların bıraktığı işlemlerdir. Bu durum 2 numaralı bitiş senaryosuna uyan ve beklenen bir durumdur. Bu durumun dışında yarıda bırakılan işlemlerde ortak bir özellik tespit edilememiş ve kullanıcının ekran mesajlarını anlamaması sebebiyle olabileceği düşünülmüştür. Bu sebeple tespit edilen ekranlarda arayüz metni değişikliğine gidilmiş ve yarıda kesilme oranları tekrardan gözlemlenmiştir. Tekrarlanan gözlemler sonucunda yarıda kesilme oranında %7 gibi azda olsa bir azalma meydana gelmiş ve kullanıcıların ekran mesajlarındaki anlamama gibi durumlarda işlemi yarıda bıraktıkları tahminimiz kısmen doğrulanmıştır.

İkinci olarak ise senaryoların başarısız tamamlanma oranlarının tespiti çalışması yapılmıştır. Buna göre başarısız tamamlanma oranı yüksek olan senaryoların genelde para çekme, para yatırma ve kredi kartı borç ödeme gibi finansal işlemlerde olduğu gözlemlenmiştir. Bu tip işlemlerin başarısız senaryoları incelendiğinde ise, işlemlerin ya kullanıcının aslında bildiği fakat dikkat etmemesi sebebiyle oluşan hatalar (4 numaralı bitiş senaryosu, Örnek: bakiye yetersiz) yüzünden, ya da çoğunlukla anlık donanımsal sebepler (3 numaralı bitiş senaryosu, Örnek: para ünitesi arızası, para kalmaması) yüzünden başarısız sonuçlandığı görülmüştür. Bu analizler bize, tasarimsal bir eksikliğin olmadığını gösterdiği gibi, donanım arızası sıklıkla gelen ATM'leri tespit edebilme imkânı da sunmuştur.

Sonuç olarak başarılı veya başarısız senaryo tasarımları, kullanıcıların davranışları üzerinde analizler yapmamıza ve kullanıcı şikâyetlerine sebep olmadan tasarımdaki olası yanlışlıkların giderilmesine yardımcı olmuştur.

### **6.1.5. Zaman Aşımına Uğrayan Senaryo Tespiti**

Günümüzde ATM'ler, her bir arayüzde 20 saniye gibi bir süre beklemekte ve kullanıcının bu süre zarfında işlem yapmaması durumunda, kullanıcıya işlemi devam ettirmek isteyip istemediği sorusu sorulmaktadır. Senaryo tasarımı verileri sayesinde bu işlemde, bir alt senaryo işlemi olarak istatistiki verisi tespit edilebilir olmuştur. Buna göre zaman aşımı alt senaryosuna giren senaryoların hangileri olduğu gözlemlenmiş ve en çok zaman aşımına uğrayan işlemlerin havale ve EFT gibi IBAN gerektiren senaryolar olduğu görülmüştür. Bu tespite göre yapılan çalışmayla birlikte

IBAN girilen ekranlardan süre sınırı uzatılmış ve gözlemler tekrar yapılmıştır. Değişiklik sonrası sonucuna göre havale ve EFT gibi işlemlerde zaman aşımına uğrama sayısı %85 gibi büyük bir oranda azalmıştır. Bu sonuçta gösteriyor ki, analiz sonucu yapılan çıkarımsama ve tasarımdaki eksiklik doğru tespit edilmiş ve sorun giderilmiştir.

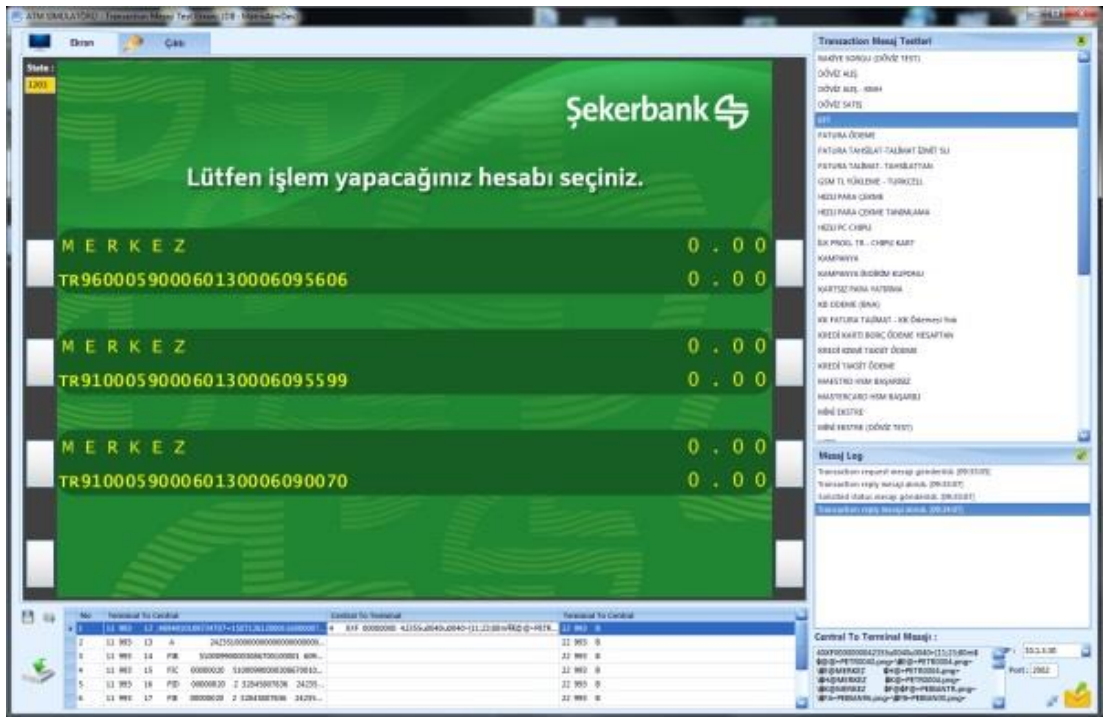
Sonuç olarak senaryo tasarımları üzerine yapılan zaman aşımı metriği analizi, arayüz tasarımlarındaki yanlışlıkları ortaya çıkarmakta ve yanlışların düzeltilmesi, sistemin kullanılabilirliği doğrudan attırmaktadır.

## 6.2. İşlem Testi

ATM arayüz performansı analizlerinin daha verimli yapılabilmesi ve test ortamında ATM'ye gitme zorunluluğunu ortadan kaldırmak için ATM arayüz yönetim sistemi içerisinde "İşlem (Transaction) Test" ekranı yapılmıştır. Buna göre senaryo tasarım ekranlarında tasarlanan senaryolar test ortamlarında bilgisayarlar üzerinde sanki ATM'de deneniyormuşçasına test edilebilmektedir. Bu ekran sisteme bir ATM gibi tanımlanırken, ATM'de oluşan mesajlar bu ekran üzerinden sunuculara sanki gerçek bir mesaj gelmiş gibi gönderilir ve gelen cevaplar yine sanki bir ATM arayüzünde görüldüğü gibi ekranda görülmektedir. Mesajlaşmalar TCP (Transmission Control Protocol) protokolü aracılığıyla sunuculara, NDC standartlarına uygun olarak işlem isteği (transaction request) mesajı olarak gelmekte, ardından sunucularda bu isteklere işlem cevabı (transaction reply) mesajı olarak cevap vermektedir.

İşlem testi ekranı, tasarım aşamasındaki ekran arayüzlerindeki parametrik verileri de ATM ekranında görüldüğü gibi gösterdiğinden, tasarım aşamasında geçirilen süreyi oldukça kısaltmaktadır. Tasarımcılar tasarladığı parametrik ekranın, verileriyle birlikte nasıl görüldüğünü bu ekran sayesinde daha tasarım aşamasında test edebilmektedir. Ayrıca çıktı tasarımları da yine parametrik verilerle bu ekrandan test edilebilmekte ve gereksiz kâğıt israfının önüne geçilmektedir. Tasarım test ortamının bu denli basit ve kullanılabilir olması, tasarım sürecini kısaltmanın yanında tasarımdan oluşabilecek hataları da önceden giderebilme olanağı sunar.

ATM üzerinde bir senaryonun testi için ATM'nin başlangıç düğümünden senaryonun başlangıç düğümüne kadar gelmek zorunda olacağından, aslında test edilen senaryoya gelene kadar geçen süre boşa geçirilen bir zaman olarak karşımıza çıkmaktadır. Fakat bu ekran aynı zamanda senaryo temelli test olanağı da sunduğundan, başta geçirilen test süresi sıfıra inmiş olacaktır. Yapılacak testler sadece denenen senaryo üzerinden olacağından, diğer senaryolardaki çalışmalardan veya hatalardan da etkilenmemiş olacaktır. Bu da test sürecinde geçirilen zamanı azaltacağı gibi, testlerdeki başarı oranını da yükseltecektir. İşlem test ekranı aynı zamanda yapılan testlerin kayıt edilebilme olanağı sunduğundan, başarılı ve başarısız senaryolar tekrar tekrar, farklı kart veya durumlar içinde kolaylıkla test edilebilmektedir. Bu durum aynı zamanda, ATM üzerinde oluşturulması zor veya imkânsız olan durumlarında denenebilmesini sağlarken, tüm olasılıklar gözden geçirilmiş olacaktır.



Şekil 6.2 : İşlem test ekranı.

## 7. KULLANICI MERKEZLİ ATM İŞLEM ÖLÇÜMÜ

Kullanıcı merkezli arayüz tasarımı, ilkeleri ve kullanıcılara sağladığı avantajlarından önceki bölümlerde bahsetmiştik. Bu bölümde ise ATM arayüz yönetim sistemi senaryo tasarımı verilerini kullanarak, kullanıcı tiplerinin tespiti ve bu kullanıcılara özel oluşturulan arayüzlerle, ATM'nin nasıl kullanıcı merkezli bir arayüz tasarımına kavuştuğundan bahsedilecektir.

Kullanıcı davranışlarının tespiti çoğu sistemde çok değerli sonuçlar doğurmasının yanında, üzerinde yapılacak birçok yenilikle de kullanıcıların sisteme bağlılıklarının artırılması hedeflenmektedir. Bu durum kuşkusuz ATM kullanıcıları için de geçerlidir. Kullanıcıların ATM'de sıklıkla yapacakları işlemleri arayüzden kolay bulmaları, çoğunlukla yaptığı işlemlerdeki seçimlerinin kaydedilip işlemi bu değerlerle daha hızlı yapmaları ve kişisel tercihlerinin arayüze yansımalarını görmeleri gibi örnekler sayesinde, ATM kullanıcıları yapacakları işlemlerde arayüzü kendine ait hissedecek ve daha memnun olacaklardır [42]. Bu sebeplerle arayüzde yapılacak kişiselleştirmenin kullanılabilirliği doğrudan etkilediği düşünülmekte ve daha kullanıcı dostu bir arayüz ortaya çıkarmada önemli bir nokta olduğu görülmektedir.

Tez kapsamında çalışılan banka olan Şekerbank'ın kullanıcı verileri temel alınarak yapılan çalışmada, öncelikle kullanıcı grupları tespit edilmiş ve ardından her bir kullanıcı grubu için, ATM arayüz yönetim sisteminde tasarlanıp gerçek ortama geçmiş senaryolardaki davranışları gözlemlenmiştir. Daha sonra, yapılan bu davranış analizlerine göre çıkan sonuçlar temel alınarak farklı kullanıcı gruplarına özgü ATM arayüzleri yaratılmış ve yapılan anket çalışması ile kullanıcıların ATM arayüzlerindeki kişiselleştirmeden oluşan memnuniyeti gözlemlenmiştir. Kullanıcı memnuniyeti ölçmek için doğrudan kullanıcıdan görüş almak etkili bir yöntemdir. Bir kullanıcı için düşünüldüğünde verilen yanıtın öznel olacağı ve sağlıklı sonuç vermeyeceği gerçeği olsa da, birden fazla kullanıcıdan alınan sonuçların ortalaması alındığında kullanıcı memnuniyeti ile ilgili nesnel veri elde edilir [35].

## 7.1. Kullanıcı Grubu Tespiti

ATM arayüz tasarımı senaryo verilerini kullanarak yapılacak kullanıcı davranışları analizi öncesinde çalışılan banka olan Şekerbank'ın kullanıcı türleri incelenmiştir. Buna göre bankanın kullanıcıları gruplanırken, diğer olası sınıflandırmalardan daha homojen bir dağılıma sahip olan hesap türlerine göre yapılacak bir sınıflandırma tercih edilmiştir. Farklı hesap türlerine göre yapılan sınıflandırmaya göre şu kullanıcı grupları oluşturulmuştur:

- Mevduat Hesabı Kullanıcısı
- Yatırım Hesabı Kullanıcısı
- Ticari Kullanıcı
- Kredi Kartı Kullanıcısı
- Maaş Hesabı Kullanıcısı

## 7.2. Kullanıcı Davranışları Analizi ve Arayüzde Kişiselleştirme

Günümüzde; ATM'de kullanılan senaryo tasarım verilerinin olmayışı, kullanıcıların yaptığı işlemlerden bir sonuç çıkarılamamasına ve kullanıcı davranışları üzerine analizler yapılamamasına sebep olmaktadır. Fakat ATM arayüz yönetim sistemi sayesinde oluşturulan senaryo tasarımları, kullanıcıların yaptıkları işlemler üzerine analizler yapabilmeye olanak sunmuştur. Çalışma kapsamında da ATM arayüz yönetim sisteminin senaryo tasarımı verileri kullanılarak, seçilen kullanıcı grupları temel alınarak ATM'de yapılan işlemlerden kullanıcı davranışları analizi yapılmıştır. Analiz sonuçlarına göre kullanıcı davranışlarının ortaklaştığı noktaların tespit edilmesiyle de ATM arayüzlerinde kişiselleştirme yapılabilmektedir.

Kullanıcı davranışları analizinde yapılan çalışmalar şu şekildedir:

- Kullanıcı gruplarına göre, en sık yapılan işlem analizi

Yapılan çalışma kapsamında; ATM arayüz yönetim sisteminde oluşturulan senaryo tasarımı verileri kullanılarak, kullanıcı gruplarına göre ATM'de en sık yapılan

işlemler tespit edilmiştir. Bu tespitte, işlem sayıları hesaplanırken bir aylık bir süreye yer verilmiş ve işlemin başarılı tamamlanması şartı göz önünde bulundurulmuştur. Buna göre; banka gizliliği sebebiyle işlem sayılarına tezde yer verilememiş, fakat kullanıcı gruplarına göre en sık yapılan beş işlem şu şekilde sıralanmıştır;

Tablo 7.1 : Kullanıcı gruplarına göre işlem sırası.

Kullanıcı Grubu	En Çok Yapılan İşlem Sırası (Büyükten Küçüğe)
Mevduat Hesabı Kullanıcısı	Para Çekme, Para Yatırma, Bakiye Sorgu, Mini Ekstre, Havale
Yatırım Hesabı Kullanıcısı	Para Çekme, Vadeli Hesap Açma, Döviz Alış-Satış, Para Yatırma, Bakiye Sorgu
Ticari Kullanıcı	Bakiye Sorgu, Para Çekme, Para Yatırma, EFT, Kredi Sorgu
Kredi Kartı Kullanıcısı	Nakit Avans, Taksitli Nakit Avans, Para Çekme, Kredi Kartı Ekstre Görüntüleme, Kredi Kartı Borç Ödeme
Maaş Hesabı Kullanıcısı	Hızlı Para Çekme, Para Çekme, Bakiye Sorgu, Mini Ekstre, Para Yatırma

Kullanıcı gruplarına göre en sık yapılan işlemlerin farklılaşması, farklı kullanıcı gruplarına özel ATM ana menü arayüzü çıkarma gerekliliğini de doğurmuştur. Çünkü her kullanıcı grubunun en sık yaptığı işlemi ana menüde ve en üst sıralarda görmesi, kullanıcının işlemi kolaylıkla bulmasına sebep olacak ve memnuniyetini de arttıracaktır. Analiz sonucuna göre farklı kullanıcı gruplarına yaratılan arayüzlerin örneği Şekil 7.1’de gösterilmiştir. Arayüzde yaratılan bu farklılaşmanın ardından, kullanıcı memnuniyetini ölçen anket çalışmasına bir sonraki bölümde yer verilmiştir.



Şekil 7.1 : Mevduat hesabı ve kredi kartı hesabı kullanıcısı ana menüleri.



- Kullanıcı gruplarına göre, farklı işlemlerdeki makbuz tercihleri analizi

ATM arayüz yönetim sistemi tasarım verileri incelendiğinde, çoğu mevcut senaryo akışına göre, işlem sonunda makbuz alıp almama bilgisi kullanıcıya bırakılmıştır. Fakat özellikle işlem süresi uzun senaryolarda, bu soruyla birlikte işlem süresi bir kademe daha uzamaktadır. Buna soruna göre yapılan analiz çalışmasında, kullanıcı gruplarına göre makbuz alma tercihleri incelenmiştir. Çıkan sonuçlara göre; mevduat hesabı kullanıcısının para yatırma işlemlerinde %52 oranla makbuz aldığı görülürken, kredi kartı hesabı kullanıcısının %95 oranda makbuz aldığı görülmektedir. Bu oranlar bize, kredi kartı hesabı kullanıcısı için %95 oranda makbuz aldığı bilinen kredi kartı borç ödeme işleminde makbuz sorusunu kaldırıp, yerine otomatik makbuz verme seçeneğini sunabilme olanağı sağlamıştır. Yapılan bu farklılaştırma çalışmasıyla, kredi kartı hesabı kullanıcısının işlem süresi kısaltılmış olacak ve kullanıcı memnuniyeti de artacaktır. Arayüzde yaratılan bu farklılaşmanın ardından, kullanıcı memnuniyetini ölçen anket çalışmasına ve sonuçlarına ise bir sonraki bölümde yer verilmiştir.

Bu çalışma ayrıca, tüm işlemler için kullanıcıların makbuz seçimlerini önceden tanımlama yapmalarını sağlayan bir arayüz ile detaylandırılıp (makbuz tanımlama senaryo tasarımı), kullanıcının hiçbir işlemde makbuz sorusu ile karşılaşmaması çözümüne gidilmesine gerek olduğu sonucunu da göstermiştir.

- Kullanıcı gruplarına göre, para çekme senaryosunda seçilen para miktarı tercihleri analizi

ATM arayüz yönetim sistemi tasarım verileriyle oluşturulan para çekme senaryosu üzerinde mevcut kullanıcı türlerine göre işlem analizi yapılmış ve kullanıcı gruplarına göre çekilen para miktarlarının dağılımı gözlemlenmiştir. Buna göre her hangi bir kullanıcı grubu için her hangi bir para miktarının fazlalığı tespit edilmemiş olup, çoğunluğun TL'deki (Türk Lirası) küpürlerden olan 20, 50, 100 miktarlarının fazlalığı gözlemlenmiştir. Kullanıcı gruplarına göre gözle görülür bir farklılığın olmaması sebebiyle, tüm kullanıcılar için geçerli olacak ve en sık çekilen tutarın en üstte olduğu bir tutar listeleme arayüzü yaratılmıştır ve uygulanmıştır. Bu sıralamayla gruplara göre farklı bir arayüz oluşturulmasa da, kullanıcıların en sık kullanılan tutarları

arayüzde görmesi, aradığı tutarı kolay bulmasına ve memnuniyetlerinin artmasına sebep olacaktır.

Bu çalışma kapsamında ayrıca para çekme senaryosunun mevduat hesabı kullanıcı grubu için oldukça fazla sayıda yapıldığı ve genelde 50 TL çekmek suretiyle işlemler olduğu gözlemlenmiştir. Bu analiz sonucuna göre, mevduat hesabı kullanıcıları için, maaş hesabı kullanıcılarının ana menü arayüzünde olduğu gibi “hızlı para çekme” butonu ana menüye konulmuş ve kullanıcıların tek bir butona basma fonksiyonuyla 50 TL çekebilmesi sağlanmıştır. Bu çözüm mevduat hesabı kullanıcıları için sık yapılan bir işlemin süresini kısaltacak ve kullanıcı memnuniyetinin artmasına sebep olacaktır.

Tasarımda yapılan değişikliklerin kullanıcı üzerinde memnuniyetini ölçen anket çalışmasına ve sonuçlarına ise bir sonraki bölümde yer verilmiştir.

- Kullanıcı gruplarına göre, yapılan başarısız işlem analizi

ATM’de yapılan başarılı işlemlerin yanında başarısız işlemlerdeki kullanıcıların davranışları da bir o kadar önem arz etmektedir. Buna göre başarısız sonuçlanan işlemler üzerine kullanıcı grupları çalışması yapılmış ve grup bazında ortak bir davranışın olup olmadığı incelenmiştir. İncelemelerden çıkan sonuçlara göre; ticari kullanıcı grubunun diğer gruplara kıyasla, kredi başvuru işleminde başarı oranının çok daha düşük olduğu gözlenmiştir. Ardından bu kullanıcı grubunun işlemleri incelenmiş ve çoğu kişinin, senaryodaki maaş bilgisi alma kısmında işlemi yarıda kestikleri tespit edilmiştir. Bu sonuç ticari kullanıcı grubu için, bu sorunun kullanıcıya sorulmasının bir şekilde yanlış sonuçlar doğurduğunu (firma için maaş kavramının olmayacağını düşünüyor veya gelirini vermek istemiyor olabilirler) ve işlemin başarısız sonuçlandığını göstermiştir. Çalışılan firmanın almış olduğu kararlar sebebiyle, tespit edilen bu durum için tasarım değişikliğine gidilememiş ve başarısız olmasına neden olduğu düşünülen tasarım probleminin bu kullanıcı grubu için doğru olup olmadığı gözlemlenememiştir.

Örnekte görüldüğü üzere, normalde tek tip kullanıcı grubu için yapılacak analiz sonucunda, kredi başvuru akış tasarımındaki olası bir yanlışın tespit edilmesi mümkün olmazken; farklı kullanıcı gruplarına göre yapılan analizlerle tasarımdaki hatalar ortaya çıkabilmektedir.

Sonuç olarak kullanıcı grubu temelli işlem ölçümlerinin analiz edilmesiyle ortaya çıkan tespitler, ATM arayüzlerinde farklılaştırmaya gidilebilmesine olanak sunmuştur. Kullanıcı gruplarına göre farklılaşan arayüzler ise kullanıcıların işlemlerini arama yoluna gitmemelerini sağlarken, makinayla aralarındaki mesafeyi azaltarak memnuniyetlerini arttıran önemli bir faktör olmuştur.

Bu çalışma ayrıca; ATM arayüz yönetim sisteminin senaryo tasarımına getirdiği kolaylık sayesinde gelecek yıllarda, kullanıcılara kendi ATM arayüzlerini ATM'den kendilerinin oluşturmalarına olanak sunmaya da ön ayak olmuş bir çalışmadır.

### **7.3. Kullanıcı Memnuniyeti Anketi ve Değerlendirme**

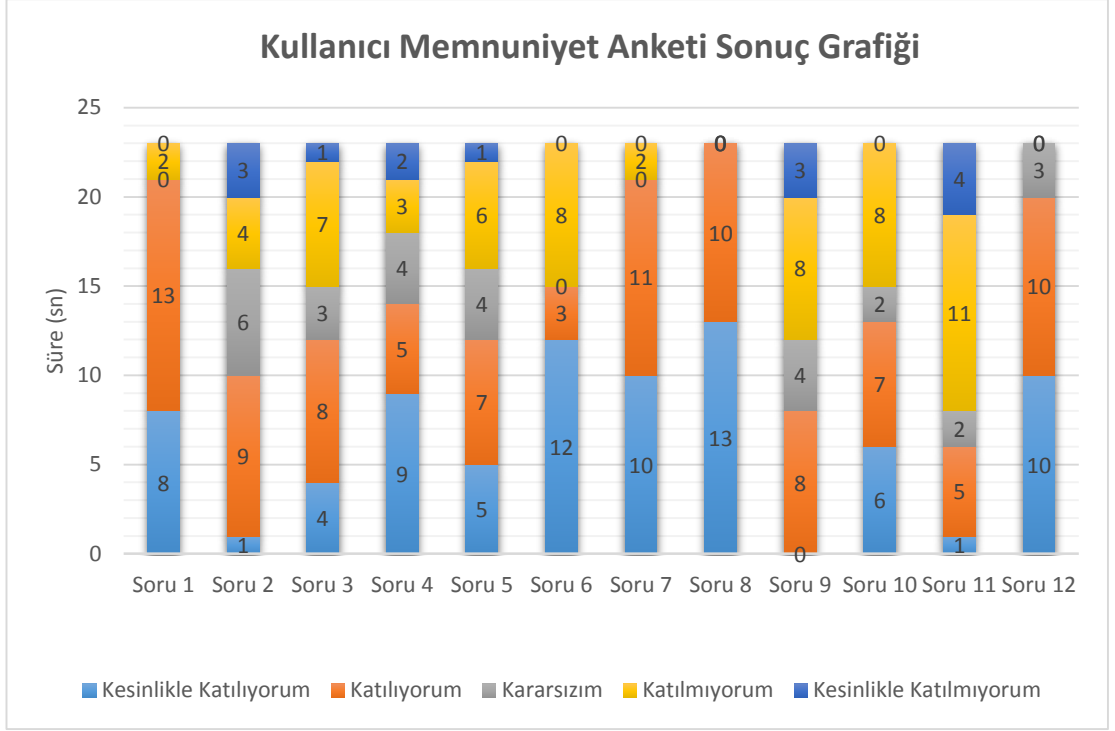
Bölüm 7.2'de anlatıldığı üzere yapılan analiz çalışmaları sonucu, çıkarılan kullanıcı gruplarına göre farklı ATM arayüzleri yaratılmış ve bu arayüzlerin kullanıcı memnuniyetini arttırdığı iddia edilmiştir. Buna göre, bu iddianın doğruluğunu gösterebilmek adına farklı yaş, cinsiyet ve kullanıcı gruplarındaki Şekerbank çalışanları arasında bir memnuniyet anketi çalışması yapılmıştır. Memnuniyet anketi 5'li Likert ölçeğinde (5=Kesinlikle katılıyorum, 4=Katılıyorum, 3=Kararsızım, 2=Katılmıyorum, 1=Kesinlikle katılmıyorum) hazırlanan toplam 12 sorudan oluşmaktadır. Gerçekleştirilen anketteki sorular, yaratılan farklı arayüzlerin eski ve yeni hallerindeki memnuniyeti karşılaştıran veya yaratılabilecek daha farklı arayüzlerdeki memnuniyeti ölçmek amacıyla oluşturulmuştur. Kullanıcı memnuniyet anketi Ek-1'de sunulmaktadır.

Kullanıcı memnuniyet anketinde farklı kullanıcı gruplarındaki 23 ile 34 yaş arasındaki, 11 bayan ve 12 bay toplam 23 kişi seçilmiş ve anketteki soruları cevaplandırmaları istenmiştir.

Sorulara verilen cevaplara göre Likert değerleri hesaplanmış ve değerlerin ortalamaları şu şekilde oluşmuştur;

Tablo 7.2 : Kullanıcı memnuniyet anketi sonuçları.

No	Sorular	Ort.	Std. S.
1	ATM ana menüsünde en sık kullandığım işlemlerin yer alması beni memnun eder.	4,17	0,83
2	ATM ana menüsünde herkesin sık kullandığı işlemlerin olması beni memnun eder.	3,04	1,15
3	ATM ana menüsünde sadece finansal işlemlerin(para çekme, para yatırma vb.) yer alması beni memnun eder.	3,30	1,22
4	ATM ana menüsünü kendim belirleyebilmeyi ve düzenleyebilmeyi isterim.	3,70	1,36
5	ATM, her işlem sonunda makbuz isteyip istemediğimi bana sormalı.	3,39	1,23
6	Makbuz alıp almama kararımı belirleyebilmeyi ve ATM'nin bu kararı hatırlamasını isterim.	3,83	1,40
7	ATM, makbuz yerine geçebilecek SMS, E-Posta gibi seçenekleri isteyip istemediğimi bana sormalı.	4,26	0,86
8	ATM'de, makbuz yerine geçebilecek SMS, E-Posta gibi seçenekleri kendim belirleyebilmeyi, düzenleyebilmeyi isterim.	4,57	0,51
9	Para çekme işleminde tutar listelenirken sadece standart kúpürler listelenmeli.	2,74	1,10
10	Para çekme işleminde tutar listelenirken en sık kullandığım tutarlar listelenmeli.	3,68	1,24
11	ATM'de kişiselleştirilmiş sesli arayüz sunulması beni memnun eder.	2,48	1,16
12	ATM'de kampanya sorusuyla karşılaşma durumumu kendim belirleyebilmeyi isterim.	4,30	0,70



Şekil 7.2: Kullanıcı memnuniyet anketi sonuç grafiği.

Anket kapsamında sorulan ana menü arayüzü sorularında; katılıyorum ile kesinlikle katılıyorum arasında çıkan bir oranda (4,17) kullanıcıların ana menüde kendi en sık kullandığı işlemleri görmesinin kendisini memnun edeceğini bildirmiştir. Fakat herkesin sık kullandığı işlemlerini kendi ana menüsünde görmesi kullanıcıları memnun edememiştir (3,04). Bu da bize ana menüde, kullanıcıların kendi en sık kullandığı işlemlere göre bir menü istediğini ve arayüzde kişiselleştirmeden memnun olduklarını göstermektedir. Bunun yanında bir önceki bölümde bahsedilen, kendi ana menüsünü kullanıcının kendinin belirleyebilmesi seçeneği de yine kullanıcıları memnun eden bir seçenek olmuştur (3,70). Çalışma kapsamında yapılanların ön ayak olduğu bir başka çalışmayla ise kullanıcıları gruplanma yapmadan, ana menü arayüzünü kendilerinin belirleyebileceği bir seçenek sunulması da kullanıcıların memnuniyeti arttıracaktır.

Anket kapsamında sorulan makbuz sorularına göre; ATM'nin kendilerine makbuz sorularını her seferinde sormak yerine makbuz seçimlerini hatırlaması ve bunu bir sonraki işlemlerde kullanması seçeneğinden memnun kalmışlardır (3,83). Bu memnuniyet makbuz yerine sunulabilecek SMS, E-Posta gibi seçenekleri kendilerinin belirleyebilmeleri fonksiyonu ile memnuniyeti daha da arttıran bir sonuç

doğurmuştur (4,57). Bu sonuçlar, kullanıcıların makbuz seçimlerindeki kişiselleştirmelerden memnun olduğunu, ayrıca makbuz dışında artan bir arayüzle farklı bildirim seçenekleriyle de, tercihlerin kendilerinde olduğu bir arayüzü istedikleri görülmektedir.

Anket kapsamında sorulan tutar listeleme arayüzünde olmasını istedikleri tutar miktarları sorularına göre; kullanıcıların sadece standart küpürlerin listelendiği bir arayüzden memnun olmadıkları (2,74), buna karşın kendi en sık kullandıkları küpürlerin listelenmesinden memnun oldukları görülmektedir (3,68). Bu sonuç, kullanıcıların tutar listelenirken genel bir tutar listesinden memnun olmadıklarını, fakat kişiselleşmiş bir tutar listesiyle ATM arayüzünün farklılaştırmanın, onlar için daha memnun edici olduğunu göstermektedir.

Anket kapsamında sunulan bazı farklı sorulara göre ise; kullanıcıların ses arayüzünden memnun olmadıkları, fakat kampanya arayüzünün kendi tercihlerine bırakılmasının onları memnun edeceği görülmektedir.

Sonuç olarak anket çalışmasıyla ölçülmeye çalışılan kullanıcı memnuniyetlerine göre, ATM arayüzünde kişiselleştirmenin genel olarak kullanıcıları memnun ettiği görülmektedir.

## 8. SONUÇLAR

Bankacılık sektöründe artan servis ihtiyaçlarına karşın, yaygın kullanım ağına sahip ATM cihazlarının arayüzlerinin sabit kaldığı ve hizmet sağlayan yeni servislerin arayüzlerinin geliştirilmesinin zorlaştığı gözlemlenmiştir. Tez kapsamında da bu sebeple nesne yönelimli ve kullanıcı dostu “ATM Arayüz Yönetim Sistemi” geliştirilmiştir.

Bu çalışmayla; ATM cihazlarda kullanılan NDC protokolü standartlarına uygun, yeni nesil işlemlerin rahatlıkla tanımlanabileceği, banka personelleri ve tasarımcılar tarafından en az hatayla yönetilebilen, arayüz değişimlerinde kullanım sorunlarını en aza indiren ve kullanıcılar tarafından kolay kullanılabilir bir ATM arayüz yönetim sistemi geliştirilmiştir. Geliştirilen sistemin arayüz tasarımına getirdiği modülerlik ve yönetilebilirlik sayesinde arayüz tasarımları, daha hızlı ve tasarım kaynaklı performans sorunu yaratmayacak (yaratırsa da geri bildirim kolay olacak) kalitede gerçekleştirilmiştir. Sistem ayrıca tasarımcı, yazılım geliştiricisi ve operasyon yöneticisi kişilerini aynı ortam üzerinde toplamış, yaratılan tasarımların daha tasarım aşamasında kontrol edilebilmesine ve test zorluklarının ortadan kaldırılarak daha kullanıcı dostu tasarımlar oluşabilmesine olanak sunmuştur.

Çalışma kapsamında senaryo adı verilen modüler akış tasarımı yapılabilmesi için bir araçta geliştirilmiştir. Bu sayede farklı akışlardaki tasarımlar tekrar tekrar kullanılabilir ve daha tasarım aşamasında oluşabilecek tüm durumlar hesaba katılabilir hale gelmiştir. Ayrıca senaryo tasarımıyla oluşan akışlardaki veriler, ATM arayüz tasarımı kaynaklı oluşabilecek performans problemlerini tespit etmek içinde kullanılabilir olmuştur. Tez kapsamında senaryo verileri kullanılarak, işlemlerin tamamlanma süreleri, başarı oranları ve tasarımlardaki bileşen sayıları üzerine analizler yapılmış ve akışlardaki tasarım hataları ortaya çıkarılmıştır. Tasarım hatalarının tespit edilip giderilmesi ise, kullanıcıların işlemi tamamlama sürelerini azaltırken, başarı oranları ve memnuniyetlerini de arttırmıştır.

ATM senaryo tasarımı verileri; tasarım kaynaklı oluşan problemlerin tespit edilmesinde kullanıldığı gibi, kullanıcı temelli ATM işlem ölçümlerinde de kullanılmıştır. Buna göre kullanıcıların yaptığı işlemlerden kullanıcı davranışları analiz

edilmiş ve çıkan sonuçlara göre farklı kullanıcı gruplarına farklı ATM arayüzleri tasarlanmıştır. Farklı ATM arayüzlerine göre kullanıcı memnuniyet anketi yapılmış ve yapılan bu kişiselleştirme çalışmasının kullanıcılarda memnuniyeti arttığı gözlemlenmiştir.

Tez kapsamında yapılan çalışmalarda oluşturulan tasarımlar ve deneysel olarak gözlemlenen verilerin sonuçları, SAN-TEZ kapsamında çalışılan banka ATM'leri üzerinde uygulanmış olup, sisteme kattığı kazanımlar ve kullanıcı memnuniyeti ile tez çalışmasının yararlarını açıkça gözler önüne sermiştir.



## 9. GELECEK ÇALIŞMALAR

ATM arayüz yönetim sistemi; tasarlanan senaryolara özgü NDC protokolü işlem mesajlarının otomatik oluşturulması, bu mesajlara ek chipli-chipsiz kart verilerinin oluşturulması ve bu verilerin sunucuyla mesajlaşmasının sağlanması ile gelecek yıllarda tamamen bir “ATM Simülatörü” olarak kullanılabilir yapıdadır. Tez çalışmasıyla yaratılan bu sistemin, SAN-TEZ kapsamında çalışılan firma ile birlikte bir ATM Simülatörü haline dönüştürülmesi fikri, yakın zaman ek çalışmalarla hayata geçirilecek ve sektöre bu yönde büyük bir katma değer sağlayacaktır.

Tez kapsamında geliştirilen bu sisteme ek olarak gelecekte; ATM yazılımlarını otomatik güncelleme, yazılımların dağıtımı ve “Kioks” cihazları arayüzleri içinde uygulanabilir bir sistem olarak düzenlenebilecektir.

## KAYNAKLAR

- [1] Web 1, (2015), [http://www.bkm.com.tr/istatistik/pos\\_atm\\_kart\\_sayisi.asp](http://www.bkm.com.tr/istatistik/pos_atm_kart_sayisi.asp), (Erişim Tarihi: 10/01/2015).
- [2] Simjian L.G., (1967), "Depository Machine", U.S. Patent 3,358,992.
- [3] Simjian L.G., (1960), "Depository Machine", U.S. Patent 2,960,377.
- [4] Couch W.R., Wylie L., (1987), "Automatic Self Service Machine System and Method", U.S. Patent 4,650,977.
- [5] Gatto J.G., (1996), "Electronic Fund Transfer System", U.S. Patent 5,546,523.
- [6] Oberan S.L., (2008), "Non Cash Media Card and the Process For Dispensing from Automated Teller", U.S. Patent 2008/0116259.
- [7] Magee P.D., Barker D., Ryan P., (2003), "Automated Teller Machine Interface", U.S. Patent D481,518 S.
- [8] NCR, (1998), "PersonaS ATM System Description", 1st Edition, NCR Corp.
- [9] Hayashi F., Sullivan R.C, Weiner S.E., (2006), "A Guide to ATM and Debit Card Industry", 2nd Edition, Federal Reserve Bank of Kansas City.
- [10] Cruz A.M.R, (2007), "Automatic Generation of User Interfaces from Domain and Use Case Models", 2007 Eighth International Conference on the Quality of Information and Communications Technology, 208-212, Lisbon, Portugal, 12-14 September.
- [11] Korman B.R., (2001), "Multi Transactional Architecture", U.S. Patent 6,308,887 B1.
- [12] Curran K., King D., (2008), "Investigating the Human Computer Interaction Problems with Automated Teller Machine (ATM) Navigation Menus", Interactive Technology and Smart Education, 5(1), 59-79.
- [13] Farooq A., (2009), "Haptics in Kiosks and ATMs for the Disabled", Graduate Thesis, University of Tampere.
- [14] Koskimies K., Monnisto T., Systa T., Tuomi J., (1998), "Automated Support for Modeling Object Oriented Software", Software IEEE, 15(1), 87-94.
- [15] Bass L., Clements P., Kazman R., (2003), "Software Architecture in Practice", 2nd Edition, Addison-Wesley Professional.

- [16] Web 2, (2014), [http://en.wikipedia.org/wiki/State\\_transition\\_table](http://en.wikipedia.org/wiki/State_transition_table), (Erişim Tarihi: 22/12/2014).
- [17] Andrei S.M., Lacramioara R., (2010), "Customer Satisfaction Regarding Bank'S Distribution Channels – The Atm Network", University of Oradea - Annals of Faculty of Economics Journal, 1(1), 772-777.
- [18] NCR, (2012), "APTRA Advance NDC Refecence Manuel", 2nd Edition, NCR Crp.
- [19] Ambrose J., Arnalz G., Coker J.L., (2002), "Client Server System With Thin Client Architecture", U.S. Patent 2002/0065879 A1.
- [20] Forsyth G., (2013), "Financial Services Center", U.S. Patent 2013/0238498 A1.
- [21] Comlekoglu F., Gilbert T.A., (2005), "Thin Client End System For Virtual Private Network", U.S. Patent 2005/0198532 A1.
- [22] ISO, (1998), Ergonomic requirements for office work with visual display terminals, ISO/IEC 9241-11, International Organization for Standardization.
- [23] Thatcher A., Mahlangu S., Zimmerman C., (2006), "Accessibility of ATMS for the Functionally Illiterate Through Icon-based Interfaces", Behaviour & Information Technology, 25(1), 65-81.
- [24] Do C., Merman M., (2000), "Automatic Bank Teller Machine For The Blind and Visually Impaired", U.S. Patent 6,061,666.
- [25] Manzke J. M., Egan D. H., Felix D., Krueger H., (1998), "What Makes an Automated Teller Machine Usable by Blind Users?", Ergonomics, 41(7), 982-999.
- [26] Löwgren J., (1998), "History, State and Future of User Interface Management Systems", ACM SIGCHI Bulletin, 20(1), 32-44.
- [27] Web 3, (2014), [http://en.wikipedia.org/wiki/Object-oriented\\_analysis\\_and\\_design](http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design), (Erişim Tarihi: 25/12/2014).
- [28] Blaha M.R., Rumbaugh J.R., (2004), "Object-Oriented Modeling and Design with UML", 2nd Edition, Prentice Hall.
- [29] Sibert J.L., Hurley W.D., Bleser T.W, (1986), "An Object-Oriented User Interface Management System", ACM SIGGRAPH Computer Graphics, 20(4), 259-268.
- [30] Web 4, (2015), [http://en.wikipedia.org/wiki/Software\\_testing](http://en.wikipedia.org/wiki/Software_testing), (Erişim Tarihi: 15/01/2015).
- [31] Galitz W.O., (2007), "The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques", 3rd Edition, Wiley.

- [32] Web 5, (2015), [https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology-media-telecommunications/tr\\_globalmobilesecurirtysurvey\\_infographic.pdf](https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology-media-telecommunications/tr_globalmobilesecurirtysurvey_infographic.pdf), (Eriřim Tarihi: 31/01/2015).
- [33] Ziefle M., Bay S., (2005), "How Older Adults Meet Complexity: Aging Effects on the Usability of Different Mobile Phones", *Behaviour & Information Technology*, 24(5), 375-389.
- [34] Web 6, (2015), <https://infogr.am/turkiyede-vasa-gore-akilli-telefon-kullanimi>, (Eriřim Tarihi: 15/02/2015).
- [35] Nielsen J., (1993), "Usability Engineering", 1st Edition, Morgan Kaufmann.
- [36] Chapman R., Hyland P., (2004), "Complexity and Learning Behaviors in Product Innovation", *Technovation*, 24(7), 553-561.
- [37] Abras C., Maloney-Krichmar D., Preece J., (2005), "User-Centered Design", *Berkshire Encyclopedia of Human-Computer Interaction: When Science Fiction Becomes Science Fact*, 2(1), 763-768.
- [38] Web 7, (2015), [http://www.sapdesignguild.org/editions/edition10/ucd\\_overview.asp](http://www.sapdesignguild.org/editions/edition10/ucd_overview.asp), (Eriřim Tarihi: 17/02/2015).
- [39] Web 8, (2015), [http://en.wikipedia.org/wiki/User-centered\\_design](http://en.wikipedia.org/wiki/User-centered_design), (Eriřim Tarihi: 17/02/2015).
- [40] Chin J.P., Diehl V.A., Norman K.L., (1988), "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface", *CHI '88 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 213-218, New York, USA, 01 May.
- [41] McAleese R., (1990), "Hypertext: State of the Art", 1e, Intellect Books.
- [42] Frøkjær E., Hertzum M., Hornbæk K., (2000), "Measuring Usability: Are Effectiveness, Efficiency and Satisfaction Really Correlated?", *CHI '00 Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 345-352, New York, USA, 01 April.
- [43] Web 9, (2015), [http://www.prosisyazilim.com.tr/matrix\\_atm\\_eng.htm](http://www.prosisyazilim.com.tr/matrix_atm_eng.htm), (Eriřim Tarihi: 02/03/2015).
- [44] Elkoutbi M., Khriş I., Keller R.K., (1999), "Generating User Interface Prototypes from Scenarios", *2013 21st IEEE International Requirements Engineering Conference*, 150-159, Limerick, Ireland, 07-11 June.
- [45] řen O., Gökürk M., (2012), "Simülatör Tabanlı ATM Arayüz Yönetim ve Geliřtirme Sistemi", *Akademik Biliřim'12-XIV*, 187-194, Uřak, Türkiye, 01-03 řubat.

## ÖZGEÇMİŞ

1987 yılında İzmir’de doğdu. 2005 yılında İzmir Gürçeşme Anadolu Lisesini bitirdi. 2011 yılında Gebze Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği bölümünden mezun oldu. Aynı yıl Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda yüksek lisansına başladı. Halen SAN-TEZ projesi işbirliğinde çalışmış olduğu Cybersoft şirketi ŞEKERBANK ATM ekibinde Uzman Yazılım Mühendisi olarak görevini sürdürmektedir.

## EKLER

### Ek-A: Kullanıcı Memnuniyet Anketi

Tablo A1.1 : Kullanıcı memnuniyeti anketi.

Yaş:	Tarih:
Cinsiyet:	Kullanıcı Grubu:
1	ATM ana menüsünde en sık kullandığım işlemlerin yer alması beni memnun eder. <input type="checkbox"/> Kesinlikle katılmıyorum <input type="checkbox"/> Katılmıyorum <input type="checkbox"/> Kararsızım <input type="checkbox"/> Katılıyorum <input type="checkbox"/> Kesinlikle katılıyorum
2	ATM ana menüsünde herkesin sık kullandığı işlemlerin olması beni memnun eder. <input type="checkbox"/> Kesinlikle katılmıyorum <input type="checkbox"/> Katılmıyorum <input type="checkbox"/> Kararsızım <input type="checkbox"/> Katılıyorum <input type="checkbox"/> Kesinlikle katılıyorum
3	ATM ana menüsünde sadece finansal işlemlerin(para çekme, para yatırma vb.) yer alması beni memnun eder. <input type="checkbox"/> Kesinlikle katılmıyorum <input type="checkbox"/> Katılmıyorum <input type="checkbox"/> Kararsızım <input type="checkbox"/> Katılıyorum <input type="checkbox"/> Kesinlikle katılıyorum
4	ATM ana menüsünü kendim belirleyebilmeyi ve düzenleyebilmeyi isterim. <input type="checkbox"/> Kesinlikle katılmıyorum <input type="checkbox"/> Katılmıyorum <input type="checkbox"/> Kararsızım <input type="checkbox"/> Katılıyorum <input type="checkbox"/> Kesinlikle katılıyorum
5	ATM, her işlem sonunda makbuz isteyip istemediğimi bana sormalı. <input type="checkbox"/> Kesinlikle katılmıyorum <input type="checkbox"/> Katılmıyorum <input type="checkbox"/> Kararsızım <input type="checkbox"/> Katılıyorum <input type="checkbox"/> Kesinlikle katılıyorum
6	Makbuz alıp almama kararımı belirleyebilmeyi ve ATM'nin bu kararı hatırlamasını isterim. <input type="checkbox"/> Kesinlikle katılmıyorum <input type="checkbox"/> Katılmıyorum <input type="checkbox"/> Kararsızım <input type="checkbox"/> Katılıyorum <input type="checkbox"/> Kesinlikle katılıyorum
7	ATM, makbuz yerine geçebilecek SMS, E-Posta gibi seçenekleri isteyip istemediğimi bana sormalı. <input type="checkbox"/> Kesinlikle katılmıyorum <input type="checkbox"/> Katılmıyorum <input type="checkbox"/> Kararsızım <input type="checkbox"/> Katılıyorum <input type="checkbox"/> Kesinlikle katılıyorum

Tablo A1.1 : Devam.

8	<p>ATM’de, makbuz yerine geçebilecek SMS, E-Posta gibi seçenekleri kendim belirleyebilmeyi ve düzenleyebilmeyi isterim.</p> <p><input type="checkbox"/>Kesinlikle katılmıyorum   <input type="checkbox"/>Katılmıyorum   <input type="checkbox"/>Kararsızım   <input type="checkbox"/>Katılıyorum</p> <p><input type="checkbox"/>Kesinlikle katılıyorum</p>
9	<p>Para çekme işleminde tutar listelenirken sadece standart küpürler listelenmeli.</p> <p><input type="checkbox"/>Kesinlikle katılmıyorum   <input type="checkbox"/>Katılmıyorum   <input type="checkbox"/>Kararsızım   <input type="checkbox"/>Katılıyorum</p> <p><input type="checkbox"/>Kesinlikle katılıyorum</p>
10	<p>Para çekme işleminde tutar listelenirken en sık kullandığım tutarlar listelenmeli.</p> <p><input type="checkbox"/>Kesinlikle katılmıyorum   <input type="checkbox"/>Katılmıyorum   <input type="checkbox"/>Kararsızım   <input type="checkbox"/>Katılıyorum</p> <p><input type="checkbox"/>Kesinlikle katılıyorum</p>
11	<p>ATM’de kişiselleştirilmiş sesli arayüz sunulması beni memnun eder.</p> <p><input type="checkbox"/>Kesinlikle katılmıyorum   <input type="checkbox"/>Katılmıyorum   <input type="checkbox"/>Kararsızım   <input type="checkbox"/>Katılıyorum</p> <p><input type="checkbox"/>Kesinlikle katılıyorum</p>
12	<p>ATM’de kampanya sorusuyla karşılaşma durumumu kendim belirleyebilmeyi isterim.</p> <p><input type="checkbox"/>Kesinlikle katılmıyorum   <input type="checkbox"/>Katılmıyorum   <input type="checkbox"/>Kararsızım   <input type="checkbox"/>Katılıyorum</p> <p><input type="checkbox"/>Kesinlikle katılıyorum</p>