

Yıl:2023

Cilt:7

Sayı:1

Year:2023

Vol:7

No:1

**UYBİSBBD**

ULUSLARARASI YÖNETİM BİLİŞİM SİSTEMLERİ  
VE  
BİLGİSAYAR BİLİMLERİ DERGİSİ

ULUSLARARASI INTERNATIONAL JOURNAL OF  
YÖNETİM MANAGEMENT  
BİLİŞİM SİSTEMLERİ INFORMATION SYSTEMS  
VE AND  
BİLGİSAYAR BİLİMLERİ DERGİSİ COMPUTER SCIENCE

Cilt: 7 • Sayı: 1 • Haziran 2023

Vol: 7 • No: 1 • June 2023

e-ISSN: 2618 - 5954

**ULUSLARARASI YÖNETİM BİLİŞİM SİSTEMLERİ  
VE  
BİLGİSAYAR BİLİMLERİ DERGİSİ**

**INTERNATIONAL JOURNAL OF MANAGEMENT INFORMATION SYSTEMS  
AND  
COMPUTER SCIENCE**

---

Cilt: 7 • Sayı: 1 • Haziran 2023  
Vol: 7 • No: 1 • June 2023

e-ISSN: 2618-5954

**E-mail :** [ybsbb.info@gmail.com](mailto:ybsbb.info@gmail.com)

**Web :** [dergipark.gov.tr/uybisbbd](http://dergipark.gov.tr/uybisbbd)

UYBİSBBD, uluslararası hakemli, uluslararası indeksli, açık erişimli bilimsel bir dergidir



---

**ULUSLARARASI YÖNETİM BİLİŞİM SİSTEMLERİ VE BİLGİSAYAR BİLİMLERİ  
DERGİSİ**  
**INTERNATIONAL JOURNAL OF MANAGEMENT INFORMATION SYSTEMS  
AND COMPUTER SCIENCE**

---

**Kurucu (Founder)**

Dr. Öğr. Üyesi Adem KORKMAZ

**Baş Editör (Editor-in-Chief)**

Dr. Öğr. Üyesi Selma BULUT

**Editörler (Editors)**

Prof. Dr. Aysun COŞKUN

Doç. Dr. Tarık TALAN

Dr. Öğr. Üyesi Mustafa Mikail ÖZÇİLOĞLU

Dr. Öğr. Üyesi Adem KORKMAZ

Dr. Öğr. Üyesi Ayşe ÇİÇEK KORKMAZ

**Yayın Kurulu (Editorial Board)**

Prof. Dr. Florentin SMARANDACHE	(University of New Mexico, USA)
Prof. Dr. Aysun COŞKUN	(Gazi Üniversitesi)
Dr. Öğr. Üyesi Mustafa Mikail ÖZÇİLOĞLU	(Kilis 7 Aralık Üniversitesi)
Dr. Öğr. Üyesi Ayşe ÇİÇEK KORKMAZ	(Bandırma Onyediy Eylül Üniversitesi)
Dr. Öğr. Üyesi Tarık TALAN	(Gaziantep İslam Bilim ve Teknoloji Üniversitesi)
Dr. Öğr. Üyesi Adem KORKMAZ	(Bandırma Onyediy Eylül Üniversitesi)
Dr. Bogdan PATRUT	(Alexandru Ioan Cuza University of Iasi, Romania)
Dr. Iulian FURDU	(Vasile Alecsandri University of Bacau, Romania)
Dr. Sadiq HUSSAIN	(Dibrugarh University, India)
Dr. Svitlana ILNYTSKA	(National Aviation University, Ukraine)

**Danışma Kurulu (Advisory Board)**

Prof. Dr. Abdulkadir YILDIZ	(Kahramanmaraş Sütçü İmam Üniversitesi)
Prof. Dr. Aysun COŞKUN	(Gazi Üniversitesi)
Prof. Dr. Erdem UÇAR	(Trakya Üniversitesi)
Prof. Dr. Florentin SMARANDACHE	(University of New Mexico)
Prof. Dr. H. Mustafa PAKSOY	(Gaziantep Üniversitesi)
Prof. Dr. İsmail Rakıp KARAS	(Karabük Üniversitesi)
Prof. Dr. Sadettin PAKSOY	(Gaziantep Üniversitesi)
Prof. Dr. Sevinç GÜLSEÇEN	(İstanbul Üniversitesi)
Prof. Dr. Ülkü BAYKAL	(İstanbul Üniversitesi)
Prof. Dr. Yılmaz KILIÇASLAN	(Adnan Menderes Üniversitesi)
Prof. Dr. Mustafa ŞEKELİ	(Kahramanmaraş Sütçü İmam Üniversitesi)
Prof. Dr. Yusuf Ekrem AKBAŞ	(Adıyaman Üniversitesi)
Doç. Dr. Ercan BULUŞ	(Tekirdağ Namık Kemal Üniversitesi)
Doç. Dr. Erdinç UZUN	(Tekirdağ Namık Kemal Üniversitesi)
Doç. Dr. İlhan UMUT	(Trakya Üniversitesi)

**Adres (Address)**

Bandırma Onyediy Eylül Üniversitesi, Gönen Meslek Yüksekokulu  
10900 Balıkesir / TÜRKİYE

**E-mail : [ybsbb.info@gmail.com](mailto:ybsbb.info@gmail.com)**

**Web : <https://dergipark.org.tr/tr/pub/uybisbbd>**

### Bu Sayının Hakemleri

**Prof. Dr. ERSİN KARAMAN**  
(Ankara Hacı Bayram Veli Üniversitesi)

**Dr. Öğr. Üyesi Selma BULUT**  
(Kırklareli Üniversitesi)

**Doç. Dr. Selahattin KOŞUNALP**  
(Bandırma Onyedli Eylül Üniversitesi)

**Dr. Öğr. Üyesi Durmuş KOÇ**  
(Isparta Uygulamalı Bilimler Üniversitesi)

**Dr. Öğr. Üyesi Mehmet KARANFİLOĞLU**  
(İbn Haldun Üniversitesi)

**Dr. Öğr. Üyesi Ali ŞENOL**  
(Tarsus Üniversitesi)

**Dr. Öğr. Üyesi Hamza POLAT**  
(Atatürk Üniversitesi)

**Dr. Öğr. Üyesi Cemal AKTÜRK**  
(Gaziantep İslam Bilim ve Teknoloji Üniversitesi)

**Dr. Öğr. Üyesi Nihal DULKADİR YAMAN**  
(Muş Alparslan Üniversitesi)

**Dr. Armağan MACİT**  
(Ege Üniversitesi)

**Dr. İbrahim Tolga COŞKUN**  
(Çukurova Üniversitesi)

**Dr. Gökhan SATILMIŞ**

**Dr. Gözde KANDEMİR**  
(Marmara Üniversitesi)

## YAYIN POLİTİKASI

Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi yılda iki kez Haziran ve Aralık aylarında yayınlanan uluslararası hakemli bir dergidir. Dergide yer alan yazılar kaynak gösterilmeksizin kısmen ya da tamamen iktibas edilemez. Bu dergide yayınlanan çalışmaların bilim ve dil sorumluluğu yazarlarına aittir.

Dergimize gönderilen çalışmalar, alanında uzman iki ayrı hakem tarafından incelendikten sonra uygun görülenler yayınlanmaktadır. Yazım kurallarına ilişkin bilgilere dergimizin web adresinde yer verilmiştir. Bu derginin tüm hakları saklıdır. Önceden yazılı izin almaksızın hiçbir iletişim ve kopyalama sistemi kullanılarak yeniden kopyalanamaz, çoğaltılamaz ve satılamaz.

---

International Journal of Management Information Systems and Computer Science is an international peer-reviewed journal which is published two times a year in June and December. The articles cannot be cited partly or entirely without showing resources. The responsibility about scientific and grammatical issues is belong to authors.

The papers sent to the journal are reviewed by two referees and after their approval, they will be sent to edit before being published. Writing & Publishing Policies can be found in the journal's website. All rights reserved. No part of this publication may be reproduced, stored or introduced into a retrieval system without prior written permission.

## Makaleler / Articles

### **Çok Kriterli Karar Verme Problemlerinin Çözümüne Yönelik Otomasyon Geliştirme**

*Software Development for Solving Multi Criteria Decision Problems*

Makale Türü: Araştırma Makalesi / Paper Type: Research Paper

**Ahmet Turan ÖZYALÇIN & Hüdaverdi BİRCAN**

**1-32**

### **Yükseköğretimde Yapay Zekânın Rolü: Anatomi Dersi için ChatGPT Değerlendirmesi**

*The Role of Artificial Intelligence in Higher Education: ChatGPT Assessment for Anatomy Course*

Makale Türü: Araştırma Makalesi / Paper Type: Research Paper

**Tarık TALAN & Yusuf KALINKARA**

**33-40**

### **Yazılım Dağıtım Sürecinin Otomatikleştirilmesine İlişkin Uygulamalı Bir Çalışma**

*A Practical Study on Automatizing the Software Deployment Process*

Makale Türü: Araştırma Makalesi / Paper Type: Research Paper

**Ender ŞAHİNASLAN & Nusret ARPACIOĞLU & Önder ŞAHİNASLAN**

**41-67**

### **Bulanık Çok Kriterli Karar Verme Yöntemleri Kullanılarak Karar Destek Sistemiyle Çalışan Ergonomik Kiosk Cihazı Tasarımı ve Yer Seçimi**

*Working with Decision Support System Ergonomic Kiosk Device Design and Location Selection Using Fuzzy Multi-Criteria Decision-Making Methods*

Makale Türü: Derleme Makalesi / Paper Type: Review Paper

**Yusuf YAZAR & Buse BAYRAM & Şeyma Gül COŞKUN & Tamer EREN**

**68-84**

# Çok Kriterli Karar Verme Problemlerinin Çözümüne Yönelik Otomasyon Geliştirme<sup>1</sup>

\*\*\*

## Software Development for Solving Multi Criteria Decision Problems

Ahmet Turan ÖZYALÇIN<sup>2</sup> 

Hüdaverdi BİRCAN<sup>3</sup> 

DOI:10.33461/uybisbbd.1194495

### Makale Bilgileri

**Makale Türü:**  
Araştırma Makalesi

**Geliş Tarihi:**  
26.10.2022

**Kabul Tarihi:**  
24.02.2023

©2023 UYBİSBBD  
Tüm hakları saklıdır.



### Öz

*Bu çalışmada çok kriterli karar verme yöntemleri kullanılarak problemlerin hatasız ve hızlı çözülmesini sağlayan, PHP internet tabanlı dinamik programlama dili ve MySQL veri tabanı kullanılarak, çevrimiçi çalışabilen bir otomasyon geliştirilmiştir. ÇKKV yöntemleri ile ilgili bilgi sahibi olan veya olmayan tüm karar vericilerin problemlerini çözebilmeleri amaçlanmıştır. Oluşturulan otomasyon kullanıcıların 100 farklı alternatifi, 10 farklı kriter ile analiz edebilmelerine olanak sağlamıştır. Analiz yaparken 4 farklı ağırlıklandırma yaklaşımı ve 4 farklı problem çözme yöntemi kullanılabilmektedir. Ayrıca sonuçları birleştirerek tek bir sıralama yapmak imkânı da sağlanmıştır. Ağırlıklandırma yaklaşımlarından AHP, ENTROPY, DEMATEL yöntemlerinin kullanılması ve karar vericilerin istedikleri kriter ağırlıklarını girebilmesine olanak sağlayan bu otomasyon, problemi EDAS, ELECTRE I, TOPSIS ve VIKOR yöntemleri kullanarak analiz edebilmektedir. Otomasyona girilen problemlerin çözümlerinde kullanılan yöntemlerin tüm basamakları tablo şeklinde ekrana yazdırılarak kullanıcıların tüm adımları gözlemleyebilmesi sağlanmıştır. Otomasyon COPELAND bütünlük sıralama yöntemini kullanarak kullanıcılara tek bir sıralama yapma imkânı sağlamıştır.*

**Anahtar Kelimeler:** Çok Kriterli Karar Verme Yöntemleri, Ağırlıklandırma Yöntemleri, Çözüm Yöntemleri, PHP, MySQL.

### Abstract

*In this study, an online automation was developed by using PHP web dynamic programming language and MySQL database, which provides error-free and fast solving of problems by using multi-criteria decision making methods. It is possible for all decision makers, with or without knowledge of MCDM methods, to solve their problems. that allows automation users to analyze with 100 different alternatives and 10 different criteria While analyzing, 4 different weighting approaches and 4 different problem solving methods can be used. It is also possible to combine the results into a single order. The use of AHP, ENTROPY, DEMATEL methods from weight determination approaches and this automation, which allows decision makers to enter the criterion weights they want, can be analyzed using EDAS, ELECTRE I, TOPSIS and VIKOR methods. All steps of the methods used in the solutions of the problems entered into automation were printed on the screen in the form of a table, allowing the users to observe all the steps. Automation COPELAND provides the ability to perform a single sequencing using onboard sequencing data.*

**Keywords:** Multi Criteria Decision Making Methods, Weighting Methods, Solution Methods, PHP, MySQL.

### Article Info

**Paper Type:**  
Research Paper

**Received:**  
26.10.2022

**Accepted:**  
24.02.2023

©2023 UYBİSBBD  
All rights reserved.



**Atıf/ to Cite (APA):** Özyalçın, A.T., & Bircan, H., (2023). Çok Kriterli Karar Verme Problemlerinin Çözümüne Yönelik Otomasyon Geliştirme. International Journal of Management Information Systems and Computer Science, 7(1), 1-32.

<sup>1</sup> Bu makale Prof. Dr. Hüdaverdi BİRCAN danışmanlığında yazılan Çok Kriterli Karar Verme Problemlerinin Çözümüne Yönelik Otomasyon Geliştirme adlı yüksek lisans tezinden elde edilmiştir.

<sup>2</sup> Yüksek Lisans, Cumhuriyet Üniversitesi, Sosyal Bilimler Enstitüsü turanozyalcin@hotmail.com, Sivas, Türkiye.

<sup>3</sup> Prof. Dr., Cumhuriyet Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, İşletme Bölümü, hbircan@cumhuriyet.edu.tr, Sivas, Türkiye

## 1. GİRİŞ

İnsan, hayatta olduğu sürece gerek şahsi gerekse sosyal gereksinimlerini karşılamak için karşısına çıkan seçeneklerden birini ya da birkaçını tercih etmek zorunda kalır. Seçenekler çoğaldıkça en uygun seçeneği seçmek zor olabilir. Seçenekler arasından yapılması istenen bu seçme işi “karar verme” olarak adlandırılır (Manisalı 1981: 6). Karar vermede deneyim, sezgi ve akıl çok önemli bir yer tutmakla beraber bilimsel yöntemlerin kullanılması da bir o kadar önemlidir.

“Çok kriterli karar verme problemleri, birden fazla kriterin optimize edildiği mümkün çözüm setleri içerisinde en iyi alternatifin seçildiği problemler olarak tanımlanabilir” (Turan, 2018).

Çok kriterli karar verme yöntemleri alternatiflerin belirlenen kriterlere göre sıralanması veya seçimi için bize en iyi sonucu verir ama bu kriterlere göre alternatifleri değerlendirmek, analiz ederek seçim yapmak uzun hesaplamalar ve uzun zaman gerektirir. Çok kriterli karar verme yöntemleriyle verilen kararlar çok aşamalı olup hesaplamalar fazla zaman almakta ve hata yapılma olasılığı artmaktadır. Çevrimiçi çalışabilen, farklı yöntemlerle analiz yapabilen, ağırlıklandırma yaklaşımlarını kullanabilen ve bunları birleştirerek tek bir sıralama yapabilen yazılım bulunamamıştır. Bu boşluğu doldurmak için çevrimiçi bir otomasyon geliştirilmiştir.

Geliştirilmiş olan otomasyon sayesinde sonuca saniyeler içinde hatasız şekilde ulaşılır. Karmaşık işlemler gerektiren problemlerin çözümlerinin kolaylıkla yapılabilmesi sağlanmıştır. Geliştirilen yazılımı diğer yazılımlardan ayıran en büyük fark; 4 farklı ağırlıklandırma yöntemi ile 4 farklı problem çözme yaklaşımını bir arada sunabilmesi ve bunların kombinasyonları ile işlem yapabilmesidir. Ayrıca sıralamaları birleştirerek tek bir sıralama elde edilmesini sağlamaktadır. Yazılımın web tabanlı olması veri güvenliği, erişilebilirlik ve kullanım kolaylığı bakımından büyük avantaj sağlamaktadır.

Yazılım PHP (Hypertext Preprocessor) programlama dili ve MYSQL(My Structured Query Language) veri tabanı kullanılarak geliştirilmiştir. 12500 satır kod yazılmıştır. Bu kodlar 90 farklı dosyada ve 9 farklı veri tabanı tablosunda çalışmaktadır. Güvenlik için md5 şifreleme algoritması ve diğer güvenlik algoritmaları kullanılarak saldırılara karşı önlem alınmıştır. Kullanılan programlama dili ve veri tabanının yapısı gereği ücretsiz olması programın maliyetinin düşük olmasını ve ücretsiz olarak kullanılabilmesini sağlamıştır.

Yazılımın çevrimiçi olması ayrıca kurulumu gerek olmaması kullanım kolaylığı sağlamakla beraber yazılıma erişimi de kolaylaştırmıştır. Yapılan yazılım internete erişebilen tüm cihazlardan kullanılabilir.

## 2. LİTERATÜR TARAMASI

ÇKKV problemlerinin çözümünü birden farklı yöntem kullanılarak ağırlıklandırma yapan ve bu ağırlıklandırma ile problemin çözümünü gerçekleştiren, sıralama yapan ve sonuçları birleştirebilen çevrimiçi yazılım projesi bulunamamıştır. Geliştirilen yazılımda kullanılan yöntemlerin, farklı sektörlerdeki uygulamaları incelenmiştir.

Karaca(2013) yaptığı çalışmada otomotiv firması için bayi seçiminde ELECTRE I Ve TOPSIS yöntemlerini kullanmıştır.

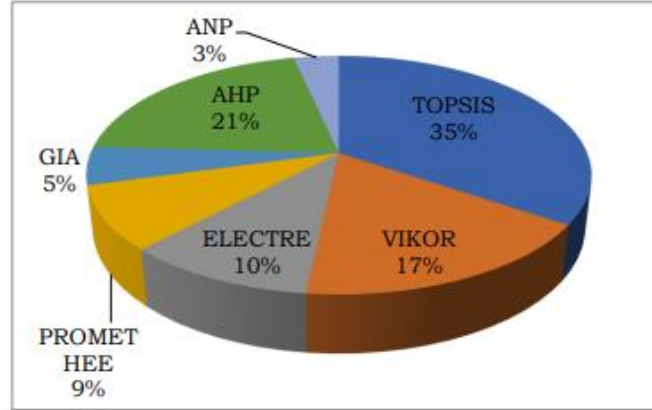
Dursun(2018) çalışmasında bir firma için tedarikçi seçimini TOPSIS ve ELECTRE yöntemlerini ayrı ayrı kullanarak yapmıştır. Çalışmanın sonucunda, TOPSIS ve ELECTRE yaklaşımları ile problem çözülmüş, firma için en uygun olabilecek alternatif seçilmiştir.

Arslan(2018) aynı problemde birden fazla ÇKKV yöntemi kullanılabilen, kullanılan yöntemlerin sonuçları farklılıklar gösterebilmekte olduğu hakkında çalışma yapmıştır. Çalışmasında 4 normalizasyon tekniği verileri ölçeklendirmede kullanılmış, çözüm yöntemi olarak TOPSIS, Gri İlişkisel Analiz, VIKOR, Copras, MOORA, MOORA, Referans Nokta ve ARAS yöntemlerini seçmiştir.



Dalbudak, E., & Rençber (2022), 2006-2020 yılları arasında literatürde bulunan ÇKKV yöntemlerinin kullanıldığı çalışmaları incelemiş ve ÇKKV yöntemlerinin kullanım oranlarını araştırmışlardır. Şekil 2.1’de, incelenen çalışmalarda kullanılan yöntemler gösterilmiştir. TOPSIS yönteminin %35 oranla en fazla kullanılan yöntem olduğu, AHP (Analytic Hierarchy Process) ve VIKOR (Vise Kriterijumska Optimizacija I Kompromisno Resenje) yöntemlerinin sırasıyla %21 ve %17 oranla TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution) yöntemini takip ettikleri sonucuna varmışlardır. En düşük orana sahip olan yöntemin %3 oranla ANP (Analytic Network Process) olduğunu belirtmişlerdir.

Şekil 2.1. İncelenen Çalışmalarda Birden Fazla Kez Kullanılan Yöntemler



Kaynak: Dalbudak, E., & Rençber (2022)

Gökhan A., Birdoğan B., (2016), Sağlık Alanında yapılan çalışmalarda literatürde yer alan ÇKKV yöntemlerini incelemişlerdir. En fazla kullanılan ÇKKV teknikleri AHP, ANP, TOPSIS, VIKOR, ELECTRE (ELimination and Et Choice Translating REality), DEMATEL (The Decision Making Trial and Evaluation Laboratory) ve PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations) yaklaşımları olduğunu belirlemişlerdir.

Çetinbaş M. (2017), yapmış olduğu çalışmada ÇKKV yöntemleri için Microsoft Excel uygulaması geliştirmiştir. Yaptığı çalışmada ağırlıklandırma yöntemi olarak AHP ağırlıklandırma yaklaşımını kullanmıştır. Karar vericilerin 5 farklı yaklaşım ile çözüme ulaşmasını sağlamıştır.

Özbek(2017) yaptığı çalışmada 15 farklı ÇKKV yönteminin Excel ile çözümünü anlatmıştır.

Tek bir yöntem ile çalışan masaüstü yazılımlar veya paket programlara dahil edilmiş yamalar yapılmıştır. Yapılan yazılımların çevrimiçi çalışmaları güncelleme problemlerini öne çıkarmış, yazılıma yeni yöntemlerin entegre edilmesi zorlaşmıştır. Bu sebeple yapılan yazılımların büyük kısmı eski yöntemler kullandığı için geçerliliğini kaybetmiştir. Ayrıca genel olarak bu yazılımlar yaklaşımlara özel olup farklı yaklaşımlarla çalışmamaktadır. Çalışma biçimleri ve algoritmaları geliştirdiğimiz otomasyon ile benzerlik gösteren herhangi bir çalışma yoktur. Ağırlıklandırma yaparak problemin çözümünü sağlayacak ve problemin çözümü için yöntem seçilmesine olanak sağlayacak yazılım bulunamamıştır.

Uluslararası ÇKKV derneğine kayıtlı ÇKKV problemlerin çözümüne yönelik oluşturulan yazılımlar Tablo 2.1. de listelenmiştir (WEB\_1 2022). Bu yazılımlar hem yapı hem de algoritma olarak tamamen farklıdır. İşlev olarak geliştirdiğimiz otomasyon ile ilgileri bulunmamakla beraber yöntemlere özel masaüstü yazılımlarıdır.

Tablo 2.1. ÇKKV Yaklaşımı İçeren Yazılımlar

S.N.	Yazılım Adı	İşlevi
1	Entscheidungsnavi (karar navigasyonu)	ÇKKV (MAUT) dayanan Almanca ve İngilizce olarak sunulan ücretsiz bir karar destek aracı yazılımı.
2	MEM - Multiplex Seçim Metodu	Çok kriterli seçim problemi için bir yöntem sunan yazılım.
3	Diviz	Karmaşık MCDA(Multiple Criteria Decision Analysis) / M algoritmaları ve deneylerini tasarlamaya, yürütmeye ve paylaşmaya yardımcı olan bir yazılımdır.
4	R için MCDA Paketi	Çok ölçütlü karar verme sürecini desteklemek: R ve MCDA paket yazılımı.
5	PriEsT	Öncelikli Tahmin Aracı: Analitik Hiyerarşi Süreci (AHP) için bir karar verme aracı yazılımı.
6	Bubble Chart Pro OPTIMAL	Çeşitli SMART (Simple Multiattribute Rating Technique) önceliği olan ve kullanımı kolay kabarcık grafiği türlerini birleştiren masaüstü yazılımıdır.
7	ChemDecide	Bir karar yapılandırma aracı ve AHP, ELECTRE III ve MARE (Mean Absolute Relative Error) kullanan üç analiz aracı içeren bir yazılım paketi.
8	ElectioVis MCDA	MCDA Problemlerinin çözümüne yönelik bir masaüstü uygulaması
9	IDSS Yazılımı	Akıllı Karar Destek Sistemleri Laboratuvarı'nın MCDM yazılımı (Poznan Üniversitesi, Polonya).
10	MCDA için MACBETH	ÇKKV yardımıyla kategorik tabanlı değerlendirme tekniği ile çekiciliği ölçme yazılımı.
11	MakeItRational	AHP tabanlı karar yazılımı

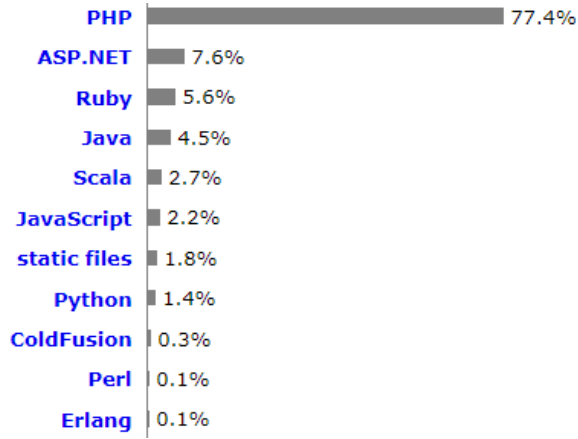
Kaynak:WEB\_1 2022

### 3. ÇOK KRİTERLİ KARAR VERME PROBLEMLERİNİN ÇÖZÜMÜNE YÖNELİK OTOMASYON GELİŞTİRME

#### 3.1. Otomasyon Programlamasına Giriş

Bilgisayara yapılması gereken işleri makine dili ile söylemek gerekir. Programlama dilleri komutları makine diline çeviren birer tercüman gibi çalışırlar. Bir yazılımcı tarafından oluşturulan kod satırları, programlama dilleri yardımıyla makine diline çevrilir. Makine diline çevirme işlemine yorumlama veya derleme denir. Web tabanlı yazılım projelerinde HTML, PHP, PYTHON, JAVA, C#, VB.NET, ASP.NET, XML gibi web tabanlı programlama dilleri kullanılır. Aynı yazılım farklı programlama dilleri ile yazılabilir. Hangi yazılım dilinde yazılacağı tamamen programcının seçimidir. Seçilen programlama dili yazılımın çıktılarını değiştirmeyecektir.

w3techs.com web sitesi dinamik web programlama konusunda araştırma ve anket yapan bir sitedir. 20 Ekim 2022 tarihli sunucu tarafı çalışan programlama dillerinin serverlarda kullanım oranı araştırması Şekil 3.1.'de yer almaktadır (WEB\_2 2022).



Şekil 3.1. Sunucu Tarafli Programlama Dillerinin Kullanım Oranları (Kaynak: WEB\_2 2022)

Şekil 3.1. de de görüldüğü gibi %77,4 oranla en fazla kullanılan web programlama dili PHP'dir. %7,6 ile ASP.NET takip etmektedir. Görüldüğü gibi en fazla kullanılan programla dili PHP olmasının sebebi ücretsiz ve açık kaynak kodlu olmasıdır. Ücretsiz olması maliyet açısından avantaj olduğu için bu çalışmada web tabanlı programlama dili olarak PHP tercih edilmiştir.

Veri tabanı verilerin satırlar ve sütunlar halinde belirli kriterlere göre depolandığı, filtreleme, ekleme ve silme işlemlerinin yapılabildiği depolama alanıdır. Geliştirilen otomasyonda veri tabanı yönetim yazılımı olarak; MYSQL veri tabanı yönetim yazılımı tercih edilmiştir.

Oracle firması tarafından geliştirilen MySQL ücretsiz, açık kaynak kodlu veri tabanı yönetim sistemidir. 2006 yılında Hurrison tarafından yazılan, MySQL kitabında, David Axmark, Allan Larsson ve Michael "Monty" Widenius tarafından geliştirildiğini belirtmiştir. 1994 yılında ilk sürümü çıkan MySQL'in; Kasım 2019 itibariyle en son sürümü 8.0 geliştirilmiş ve Ekim 2019'da yayınlanmıştır (WEB\_3 2022).

Avusturyalı bilişim şirketi "solid IT" firmasının oluşturduğu DB-Engines.com sitesinde veri tabanlarının kullanım oranları hesaplanmaktadır. Aylık olarak sürekli güncellenen ve popülerlik sıralamasına göre oluşturulan listede, MySQL; 2022 yılı eylül ayının en popüler ikinci veri tabanı yönetim yazılımı olarak yerini almıştır. Aynı listede açık kaynak kodlu veri tabanı yazılımları karşılaştırıldığında 1.sıradadır. 10 farklı veri tabanı sisteminin yer aldığı liste Tablo 3.1. de yer almaktadır (WEB\_4 2022).

Tablo 3.1. Veri Tabanı Yönetim Sistemleri Kullanılma Sıraları (Kaynak: WEB\_4 2022)

Sıra	Veri Tabanı Yönetim Sistemi
1	Oracle
2	MySQL
3	Microsoft SQL Server
4	PostgreSQL
5	MongoDB
6	IBM Db2

7	Elasticsearch
8	Redis
9	Microsoft Access
10	Cassandra

Genel olarak tanımlayacak olursak, istemci-sunucu modelli MySQL açık kaynaklı, ücretsiz dağıtılan bir veri tabanı yönetim sistemi yazılımıdır. Geliştirilen otomasyonda MYSQL veri tabanının tercih edilme sebepleri; ücretsiz olması, PHP ile uyumlu çalışması, hızlı olması ve yaygın olarak kullanılmasıdır.

### 3.2. Geliştirilen Otomasyonda Kullanılan Çok Kriterli Karar Verme Yöntemleri

Yapılan literatür araştırmasında en çok kullanılan ÇKKV yaklaşımları belirlenmiştir. Otomasyonun ilk sürümü için kullanımı artırmak amaçlı temel ÇKKV yöntemlerinden en çok tercih edilen yaklaşımlar seçilmiştir. Otomasyonun yapı itibarıyla esnek olması ileriki sürümlerinde diğer yaklaşımların da eklenmesine olanak sağlamaktadır. Otomasyon geliştirilerek <http://ckkvtez.a0001.net/> adresinde kullanıma açılmıştır.

Geliştirilen otomasyonda 4 farklı ağırlıklandırma yöntemi ve 4 farklı problem çözme yaklaşımı kullanılmıştır. Yöntemlerden elde edilen sıralama sonuçları COPELAND yaklaşımı ile birleştirilip tek bir sıralama yapılması sağlanmıştır.

Kullanılan ağırlıklandırma yaklaşımları AHP ağırlıklandırma yaklaşımı, DEMATEL ağırlıklandırma yaklaşımı ve ENTROPY ağırlıklandırma yaklaşımıdır. Bu yaklaşımlara ilave olarak kullanıcıların ağırlık girişi yapabilmelerine de imkân sağlanmıştır.

Problem çözümü sağlayan yöntemler ise ELECTRE I, EDAS, TOPSIS ve VIKOR yaklaşımlarıdır.

Problemin çözümü istenilen ağırlıklandırma yaklaşımı kullanılarak istenilen yöntemle yapılabilmekte ve sonuçlar COPELAND yöntemi ile birleştirilebilmektedir.

Yazılımda ayrıca otomasyonda fonksiyonları içeren kütüphane oluşturulmuştur. Tekrarlanan matris hesaplamalarında ve bazı işlemlerde yazılan fonksiyonlar kullanılarak çözüme ulaşılması sağlanmıştır.

#### 3.2.1. AHP Ağırlıklandırma Yaklaşımı

Çok kriterli problemleri çözümlenirken kullanılan 1970'li yıllarda geliştirilen Analitik Hiyerarşi Yöntemi, ikili karşılaştırmalar kullanarak değişkenleri analiz eden bir yöntemdir (Saaty 1986: 841-855). İkili karşılaştırmalarda Tablo 3.2. de gösterilen Saaty'nin 1-9 skalası kullanılır. Karar vericiler nicel karşılaştırmalar yapmak zorunda değildir, nitel karşılaştırmalar yapılabilmesine de imkân sağlamaktadır. Yapılan otomasyonda Saaty'nin 1-9 skalası kullanılarak 2li karşılaştırmalar yapılmıştır. 5 adımda ağırlıklandırma yapılır. Bu adımlar sırasıyla aşağıda belirtilmiştir(Saaty 1986: 841-855):

Tablo 3.2. AHP Yönteminde Kullanılan Puanlama Ölçeği (Kaynak: Saaty 1986: 841-855)

Karşılaştırmalarda Kullanılan Önem Derecesi Tablosu	
1	Eşit Derecede Önemli
3	Orta Derecede Önemli
5	Kuvvetli Derecede Önemli
7	Çok Kuvvetli Derecede Önemli
9	Mutlak Derecede Önemli
2,4,6,8	Ara Değerleri Temsil Etmektedir.

**1.Adım:** İkili karşılaştırma matrisinin genel yapısı Eşitlik 1. de gösterilmiştir. Tüm kriterler birbirleriyle karşılaştırılarak n kriter için nxn boyutlu matris oluşturulur.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (1.)$$

**2.Adım:** İkili karşılaştırma matrisi (Eşitlik 1.) 1-9 skalasına göre oluşturulduktan sonra sütun toplamları hesaplanır. Normalize edilmiş matris her eleman sütun toplamına bölünerek oluşturulur. Eşitlik 2’de gösterilen formül kullanılır.

$$a'_{ij} = \frac{a_{ij}}{\sum_{i=1}^n a_{ij}} \quad (2.)$$

**3.Adım:** Normalize edilmiş matriste Eşitlik 3. kullanılarak kriter ağırlıkları hesaplanır.

$$w_i = \left(\frac{1}{n}\right) \sum_{j=1}^n a'_{ij} \quad i, j = 1, 2, \dots, n \quad (3.)$$

**4.Adım:** Karşılaştırmaların tutarlı olup olmadığı TI (tutarlılık indeksi) Eşitlik 4. kullanılarak hesaplanır (Saaty,2000). TI değerinin hesaplanmasında kullanılan Özdeğer olarak adlandırılan  $\lambda_{max}$  eşitlik 5. yardımıyla bulunur.

$$TI = \frac{\lambda_{max} - n}{n - 1} \quad (4.)$$

$$\lambda_{max} = \frac{1}{n} \cdot \sum_{i=1}^n \left[ \frac{\sum_{j=1}^n a_{ij} \cdot w_j}{w_i} \right] \quad (5.)$$

**5.Adım:** Tutarlılık oranı TO Eşitlik 6 yardımıyla hesaplanır. Rasgele indeks değeri (RI) en fazla boyutu 15 olan matrisler için hesaplanabilmektedir.

$$TO = \frac{TI}{RI} \quad (6.)$$

TO, 0,10’ dan daha düşük bir değer alması durumunda matrisin tutarlı olduğuna karar verilir.

### 3.2.2. DEMATEL Ağırlıklandırma Yaklaşımı

DEMATEL (The Decision Making Trial and Evaluation Laboratory Method) yaklaşımı 1972 'de Cenevre Araştırma Merkezi' nin Battelle Anma Enstitüsü tarafından, araştırmalarda karışık ve karar vermenin zor olduğu problem gruplarının çözümünde kullanılması için geliştirilmiştir (Gabus ve Fontela 1976).

“DEMATEL yaklaşımı kriterleri; sebep ve sonuç kriterleri olarak ikiye ayırmaktadır. Burada sebep kriterleri, diğer kriterler üzerinde etkiye sahip olanlar; sonuç kriterleri ise diğer kriterlerden etkilenen kriterler olarak ayrılmaktadır “(Aksakal ve Dağdeviren 2010:907). Tablo 3.3. de DEMATEL yönteminde kullanılan puanlama ölçeği verilmiştir (Senvar ve diğ., 2014:143-160). Yapılan yazılımda Tablo 3.3. de gösterilen DEMATEL Puanlama Ölçeği kullanılmıştır. 5 adımda ağırlıklandırma yapılır. Bu adımlar aşağıda sırasıyla belirtilmiştir (Tzeng and Huang 2011, Aksakal ve Dağdeviren 2010: 908, Tsai and Chou 2009: 1444-1458, Demir, G & vd. 2021):

Tablo 3.3. DEMATEL Yönteminde Kullanılan Puanlama Ölçeği (Kaynak: Senvar ve diğ., 2014:143-160)

Sayısal Değerler	Dilsel İfade
0	Etkisi Yok
1	Düşük Etkili
2	Normal Etkili
3	Yüksek Etkili
4	Çok Yüksek Etkili

**1.Adım:** Kriterlerin etki düzeylerinin belirlenmesi ve direkt ilişki matrisinin oluşturulması. Uzman gruptan Tablo 3.3.'de belirlenen ölçeğe göre; “Kriterler birbirlerini hangi düzeyde etkiliyorlar?” (Senvar ve diğ., 2014:143-160) sorusuna cevap vermesi istenir.

Tablo 3.3. de belirtilen ölçek kullanılarak uzman veya uzmanlar tarafından ikili karşılaştırmalar yapılır. Eğer değerlendiren uzman sayısının fazla ise, uzmanların verdikleri puanların aritmetik ortalaması alınır. Böylelikle Eşitlik 7’de gösterilen asimetrik matris elde edilir. n tane kritere sahip bu elde ettiğimiz matris direkt ilişki matrisi (X) adı verilir ve nxn boyutundadır.

$$X = \begin{bmatrix} 0 & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & 0 \end{bmatrix} \quad (7.)$$

**2.Adım:** Normalleştirilmiş direkt ilişki matrisinin (C) oluşturulması.

Direkt ilişki matrisinin Eşitlik 7 kullanılarak elde edilmesinden sonra “Eşitlik 8.” kullanılarak her bir satır ve sütun toplamının en büyük olanı bulunur.

$$s = \max (\sum_{j=1}^n x_{ij}, \sum_{i=1}^m x_{ij}) \quad (8.)$$

Ardından direkt ilişki matrisinin her bir elemanı “s” değerine (Eşitlik 9.) bölünerek normalleştirilmiş direkt ilişki matrisi “C” hesaplanır.

$$c = \frac{x}{s} \quad (9.)$$

**3.Adım:** Toplam ilişki matrisinin (F) oluşturulması.

Toplam ilişki matrisini (F) elde etmek için, C matrisi birim matrsten çıkarılıp tersi alınır ve tekrar C matrisi ile çarpılır (Eşitlik 10.).

$$F=C*(I-C)^{-1} \quad (10.)$$

**4.Adım:** Etkileyen ve etkilenen faktör grupları matrisinin hesaplanması.

F matrisinde satır ve sütun toplamaları bulunur (Çınar, 2013).

Di: (Eşitlik 11.) Tüm satırların ayrı ayrı toplamıdır. Kriterin diğer kriterleri doğrudan veya dolaylı etkilemesini belirtir.

Ri: (Eşitlik 12.) Tüm sütunların ayrı ayrı toplamıdır. Kriterin diğer kriterlerden doğrudan veya dolaylı etkilenmesini belirtir.

Di+ Ri gönderilen ve alınan toplam etki değerini gösterir.

Di- Ri kriterlerin sisteme yaptığı toplam etkiyi gösterir. Pozitif ise etkileyen, negatif ise etkilenen olarak nitelendirilir.

$$D_i = \sum_{j=1}^n f_{ij} \quad (11.)$$

$$R_i = \sum_{j=1}^n f_{ij} \quad (12.)$$

**5.Adım:** Eşitlik 13. kullanılarak kriter ağırlıklar toplamı hesaplanır.

$$w_{ia} = \sqrt{(D_i + R_i)^2 - (D_i - R_i)^2} \quad (13.)$$

Ardından her bir ağırlık, ağırlıkların toplamına bölünerek (Eşitlik 14.) kriter ağırlıkları hesaplanır.

$$w_i = \frac{w_{ia}}{\sum_{i=1}^n w_{ia}} \quad (14.)$$

### 3.2.3. ENTROPY Ağırlıklandırma Yaklaşımı

Çok kriterli karar verme yöntemlerinde iki farklı ağırlıklandırma yaklaşımı yapılmaktadır. Bunlar objektif ağırlıklandırma ve sübjektif ağırlıklandırmadır (Shemshadi vd. 2011: 2161). Sübjektif ağırlıklandırmalar konu uzmanının değerlendirmelerinden oluşurken, objektif ağırlıklandırmalar alternatiflerin nicel değerlerinden oluşur. “Objektif ağırlıklandırma yöntemlerinden biri olan ENTROPY yöntemi, karar matrisi skorlarının bilindiği durumlarda kullanılabilir” (Çatı vd. 2017: 204).

ENTROPY yaklaşımında başlangıç matrisi kullanılarak kriter ağırlıklarının hesaplanması yapılır. Kriterlerin ağırlıklandırılması için ayrıca veri girişine ihtiyaç olmaması nedeniyle oldukça yaygın olarak kullanılır (Özdağoğlu, Yakut ve Bahar 2017: 346).

5 adımda ağırlıklandırma yapılır. Adımlar aşağıda sırasıyla belirtilmiştir (Shemshadi vd., 2011;İslamoğlu vd., 2015; Ömürbek ve Aksoy, 2016, Demir, G & vd. ,2021):

**1.Adım:** ENTROPY yönteminde diğer çok kriterli karar verme problemlerinde olduğu gibi öncelikle karar matrisi (Eşitlik 15.) oluşturulur.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \quad (15.)$$

**2.Adım:** Eşitlik 16. kullanılarak normalizasyon yapılır.

$$r_{ij} = \frac{x_{ij}}{\sum_1^j x_{ij}} \quad (16.)$$

Eşitlik 16. yardımıyla R (Eşitlik 17.) normalize edilmiş karar matrisi oluşturulur.

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1j} \\ r_{21} & r_{22} & \dots & r_{2j} \\ \vdots & \vdots & \dots & \vdots \\ r_{i1} & r_{i2} & \dots & r_{ij} \end{bmatrix} \quad (17.)$$

**3.Adım:** Eşitlik 19. kullanılarak ENTROPY katsayısı bulunur. ENTROPY değerleri Eşitlik 18. yardımıyla hesaplanır.

$$e_j = -k \sum_{j=1}^n r_{ij} * \ln(r_{ij}) \quad (i = 1 \dots n, j = 1 \dots m) \quad (18.)$$

$$k=(\ln(n))^{-1} \text{ ENTROPY Katsayısı} \quad (19.)$$

**4.Adım:** Belirsizlik katsayıları Eşitlik 20. kullanılarak hesaplanır.

$$d_j=1-e_j \quad (j=1 \dots m) \quad (20.)$$

**5.Adım:** Son adımda Eşitlik 21. kullanılarak ENTROPY kriter ağırlıklarının hesaplanması yapılır.

$$w_j = \frac{1-e_j}{\sum_{i=1}^n (1-e_j)} \quad (21.)$$

$$w_1+w_2+\dots+w_j+\dots+w_n=1$$

### 3.2.4. EDAS Yöntemi

EDAS yöntemi 2015’de Ghorabae ve arkadaşları tarafından çok kriterli karar verme yöntemlerinin çözümü için geliştirilmiştir. EDAS (Evaluation based on Distance from Average Solution- Ortalama Çözüm Uzaklığına Göre Değerlendirme) metodu karar verirken ortalamalardan faydalanır (Ghorabae vd. 2015).

EDAS yönteminde yazılım hesaplama yaparken problem çözümü için karar verici tarafından seçilen ağırlıklandırma yaklaşımını kullanır.

6 adımda problem çözülür. Adımlar aşağıda sırasıyla belirtilmiştir (Wang, Y.M., 1998, Demir, G & vd.,2021):

**1.Adım:** Karar matrisi (A) ‘nın oluşturulması

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \quad (22.)$$

**2.Adım:** Eşitlik 22. kullanılarak her bir kriterin tüm alternatiflerdeki ortalaması Eşitlik 24. yardımıyla hesaplanarak ortalama matris AV (Eşitlik 23.) bulunur.

$$AV = [AV_j]_{1xm} \quad (23.)$$



$$AV_j = \frac{\sum_{i=1}^n A_{ij}}{n} \quad (24.)$$

**3.Adım:** PDA (Ortalamadan pozitif uzaklık matrisi) ve NDA (negatif uzaklık matrisi) kriterlerin fayda yönlü ve maliyet yönlü değerlerine göre hesaplanır.

$$PDA = [PDA_{ij}]_{n \times m} \quad (25.)$$

$$NDA = [NDA_{ij}]_{n \times m} \quad (26.)$$

Kriter faydalı ise;

$$PDA_{ij} = \frac{\max(0, (A_{ij} - AV_j))}{AV_j} \quad (27.)$$

$$NDA_{ij} = \frac{\max(0, (AV_j - A_{ij}))}{AV_j} \quad (28.)$$

Kriter maliyet ise;

$$PDA_{ij} = \frac{\max(0, (AV_j - A_{ij}))}{AV_j} \quad (29.)$$

$$NDA_{ij} = \frac{\max(0, (A_{ij} - AV_j))}{AV_j} \quad (30.)$$

**4.Adım:** Alternatiflerin ağırlıklı toplam pozitif değerleri  $SP_i$  Eşitlik 31. ile, ağırlıklı toplam negatif değerleri  $SN_i$  Eşitlik 32. hesaplanır.

$$SP_i = \sum_{j=1}^m w_j x PDA_{ij} \quad (31.)$$

$$SN_i = \sum_{j=1}^m w_j x NDA_{ij} \quad (32.)$$

**5.Adım:** Tüm alternatifler için hesaplanan  $SP_i$  ve  $SN_i$  değerleri eşitlik 33. ve 34. kullanılarak normalize edilir.

$$NSP_i = \frac{SP_i}{\max_i SP_i} \quad (33.)$$

$$NSN_i = 1 - \frac{SN_i}{\max_i SN_i} \quad (34.)$$

**6.Adım:** Tüm alternatifler için değerlendirme skorları hesaplanarak (eşitlik 35.) en büyük değere sahip olan alternatif en iyi alternatif olarak seçilir.

$$AS_i = \frac{(NSP_i + NSN_i)}{2} \quad (35.)$$

### 3.2.5. ELECTRE I Yöntemi

ELECTRE I yöntemi; belirli kriterler ve bu kriterlerin ağırlıklarına göre alternatifleri birbirine göre baskınlık ölçüsüne dayanır. ELECTRE I yönteminde alternatiflerin sıralaması yapılır.

ELECTRE I yönteminde yazılım hesaplama yaparken problem çözümü için karar verici tarafından seçilen ağırlıklandırma yaklaşımını kullanır. Ağırlıklandırma verileri kullanılarak problem çözümü gerçekleştirilir.

ELECTRE I metodu ikili karşılaştırmalarla üstünlüğe dayalı çalışır. Bu metodu Bernard ROY 1968 yılında ortaya koymuştur.

7 adımda problem çözülerek sıralama yapılır. Adımlar aşağıda sırasıyla belirtilmiştir (Roy 1968):

**1.Adım:** A matrisi karar verici veya karar vericiler tarafından oluşturulan başlangıç matrisidir. m alternatif ve n kritere göre düzenlenmiş karar matrisi aşağıdaki gibi gösterilir (Eşitlik 36.).

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad (36.)$$

**2.Adım:** (R) standart matrisi Eşitlik 36. kullanılarak ve (V) ağırlıklı standart karar matrisi Eşitlik 37. kullanılarak hesaplanır.

$$r_{ij} = \frac{a_{ij}}{\sqrt{\sum_{k=1}^m (a_{kj})^2}} \quad (37.)$$

R matrisini elemanlarını hesaplamak için Eşitlik 37 kullanılır. Yeni oluşan matris Eşitlik 38. deki gibidir.

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \dots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix} \quad (38.)$$

Daha sonra Eşitlik 39. kullanılarak V ağırlıklı standart karar matrisi (Eşitlik 40.) elde edilir.

$$v_{ij} = r_{ij} \cdot w_j \quad (39.)$$

$$V = \begin{bmatrix} r_{11}w_1 & r_{12}w_2 & \dots & r_{1n}w_n \\ r_{21}w_1 & r_{22}w_2 & \dots & r_{2n}w_n \\ \vdots & \vdots & \dots & \vdots \\ r_{m1}w_1 & r_{m2}w_2 & \dots & r_{mn}w_n \end{bmatrix} \quad (40.)$$

**3.Adım:** Eşitlik 40 kullanılarak Uyum setleri belirlenir. Eşitlik 41 kullanılarak karar noktaları kıyaslanır.

$$C_{kl} = \{j, v_{kj} \geq v_{lj}\} \quad (41.)$$

Satır elemanlarının birbirlerine göre büyüklükleri karşılaştırılır. Uyum setindeki eleman sayısı en fazla kriter sayısı (n) olacak şekilde, m(m-1) adet uyumluluk seti oluşturulur. k≠l olmalıdır. ELECTRE yönteminde her uyum setine (C<sub>kl</sub>) bir uyumsuzluk seti (D<sub>kl</sub>) karşılık gelir.

$$D_{kl} = \{j, v_{kj} < v_{lj}\} \quad (42.)$$

**Adım 4:** Uyum (C) ve Uyumsuzluk Matrislerinin (D) Oluşturulması

C matrisi mxm boyutludur ve k=l için değer almaz. Uyum matrisinin (Eşitlik 44.) elemanları Eşitlik 43. yardımıyla hesaplanır.

$$c_{kj} = \sum_{j \in C_{kl}} w_j \quad (43.)$$

$$C = \begin{bmatrix} - & c(1,2) & \dots & c(1,m) \\ c(2,1) & - & \dots & c(2,m) \\ \vdots & \vdots & \dots & \vdots \\ c(m,1) & c(m,2) & \dots & - \end{bmatrix} \quad (44.)$$

Uyumsuzluk matrisinin (Eşitlik 46.) elemanları ise Eşitlik 45. yardımıyla hesaplanır.

$$d_{kl} = \frac{\max_{j \in D_{kl}} |v_{kj} - v_{lj}|}{\max_{j \in J} |v_{kj} - v_{lj}|} \quad (45.)$$

Uyumluluk matrisi (Eşitlik 44.) gibi uyumsuzluk (Eşitlik 46.) matrisi de mxm boyutludur ve k=l için değer almaz.

$$D = \begin{bmatrix} - & d(1,2) & \dots & d(1,m) \\ d(2,1) & - & \dots & d(2,m) \\ \vdots & \vdots & \dots & \vdots \\ d(m,1) & d(m,2) & \dots & - \end{bmatrix} \quad (46.)$$

**Adım 5:** Uyumluluk üstünlük matrisi; uyumluluk matrisindeki her eleman için hesaplanan eşik değeri yardımıyla oluşturulur. Eşik değeri Eşitlik 47. ile hesaplanır.

$$\bar{c} = \frac{\sum_{a=1}^m \sum_{b=1}^m c(a,b)}{m(m-1)} \quad (47.)$$

Uyumluluk üstünlük matrisi elemanları eşik değerine (Eşitlik 47.) göre Eşitlik 48. yardımıyla hesaplanır.

$$E = \begin{cases} e(a,b) = 1 & \text{eğer } c(a,b) \geq \bar{c} \\ e(a,b) = 0 & \text{eğer } c(a,b) < \bar{c} \end{cases} \quad (48.)$$

Uyumsuzluk üstünlük matrisi; uyumsuzluk matrisindeki her eleman için hesaplanan uyumsuzluk eşik değeri oluşturulur. Uyumsuzluk eşik değeri Eşitlik 49. ile hesaplanır.

$$\bar{d} = \frac{\sum_{a=1}^m \sum_{b=1}^m d(a,b)}{m(m-1)} \quad (49.)$$

Uyumsuzluk üstünlük matrisi(F) elemanları uyumsuzluk eşik değerine göre aşağıdaki Eşitlik 50. kullanılarak elde edilir.

$$F = \begin{cases} f(a,b) = 1 & \text{eğer } d(a,b) \geq \bar{d} \\ f(a,b) = 0 & \text{eğer } d(a,b) < \bar{d} \end{cases} \quad (50.)$$

**Adım 6:** Toplam baskınlık matrisi G matrisi olarak adlandırılır. m x m boyutludur ve 1 yada 0 değerlerinden oluşur. Eşitlik 51. ile hesaplanır.

$$g_{ij} = e_{ij} \cdot f_{ij} \quad (51.)$$

**Adım 7:** Toplam baskınlık matrisi(G) satırları toplanarak karar tablosu oluşturulur. Sayısal olarak değeri büyük olandan küçük olana doğru sıralama yapılır. Alternatiflerin önem dereceleri değeri büyük olandan küçük olana doğru sıralanarak belirlenir.

### 3.2.6. TOPSIS Yöntemi

TOPSIS yöntemi ile alternatifler istenilen kriterler doğrultusunda sıralanır. Bu yöntemde ilk olarak karar matrisi oluşturulur. 2.adımda ise karar matrisi normalize edilir. Normalize karar matrisi kriter ağırlıkları kullanılarak ağırlıklandırılır. İdeal çözüme ve negatif ideal çözüme olan uzaklıklar hesaplanarak tüm alternatifin göreceli puanları elde edilir. Bu puanlarla alternatiflerin sıralaması yapılır (Hwang, Yoon 1981).

TOPSIS yönteminde yazılım hesaplama yaparken problem çözümü için karar verici tarafından seçilen ağırlıklandırma yaklaşımını kullanır. Ağırlıklandırma verileri kullanılarak problem çözümü 6 adımda gerçekleştirilir Adımlar aşağıda sırasıyla belirtilmiştir (Ömürbek ve Kınay, 2013, Demir, G & vd. ,2021).:

**Adım 1:** Karar verici tarafından oluşturulan başlangıç matrisi A Eşitlik 52. de gösterilmiştir.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad (52.)$$

**Adım 2:** Karar matrisi oluşturulduktan sonra normalizasyon işlemi gerçekleştirilerek N normalizasyon matrisi Eşitlik 53. kullanılarak oluşturulur.

$$n_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^m a_{ij}^2}} \quad (i = 1, \dots, m \text{ ve } j = 1, \dots, p) \quad (53.)$$

Normalize edilmiş matris (Eşitlik 54.) aşağıdaki şekilde elde edilir.

$$N = \begin{bmatrix} n_{11} & n_{12} & \dots & n_{1p} \\ n_{21} & n_{22} & \dots & n_{2p} \\ \vdots & \vdots & \dots & \vdots \\ n_{m1} & n_{m2} & \dots & n_{mp} \end{bmatrix} \quad (54.)$$

**Adım 3:** Ağırlıklandırılmış normalize matris Eşitlik 55. ve Eşitlik 56. kullanılarak hesaplanır.

$$v_{ij} = w_j \cdot n_{ij} \quad (55.)$$

$$V = \begin{bmatrix} w_1 n_{11} & w_2 n_{12} & \dots & w_n n_{1p} \\ w_1 n_{21} & w_2 n_{22} & \dots & w_n n_{2p} \\ \vdots & \vdots & \dots & \vdots \\ w_1 n_{m1} & w_2 n_{m2} & \dots & w_n n_{mp} \end{bmatrix} \quad (56.)$$

**Adım 4:** Eşitlik 56'da gösterildiği gibi ağırlıklandırılmış normalize matris elde edildikten sonra kriter yönlerine göre her bir sütuna ait maksimum değerler ideal çözüm değerleridir. Her bir sütuna ait minimum değerler ise negatif ideal çözüm değerlerini verir.

İdeal çözüm değerleri Eşitlik 57. yardımıyla hesaplanır.

$$A^* = \left\{ \max_j v_{ji} \mid j = 1, \dots, p; i = 1, \dots, m \right\} \quad (57.)$$

İdeal negatif çözüm değerleri ise Eşitlik 58. yardımıyla hesaplanır.

$$A^- = \left\{ \min_i v_{ji} \right\} \quad (58.)$$

**Adım 5:** Öklidyen uzaklıklar kullanılarak ideal ve negatif ideal noktalara uzaklık değerleri hesaplanır. Eşitlik 59'da öklidyen uzaklık hesabı gösterilmiştir.

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (59.)$$

Eşitlik 60 ile İdeal çözüme en yakın öklidyen uzaklık hesaplanır. Negatif ideal çözüme en uzak uzaklık ise Eşitlik 61 kullanılarak hesaplanır.

İdeal uzaklık:

$$S_i^* = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^*)^2} \quad (60.)$$

Negatif ideal uzaklık:

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad (61.)$$

**Adım 6:** İdeal ve negatif ideal çözüm uzaklıklardan faydalanılarak ideal çözüme göreli yakınlık değerleri hesaplanır.

$$C_i^* = \frac{S_i^-}{S_i^- + S_i^*} \quad (62.)$$

Eşitlik 62. ile hesaplanan ideal çözüme göreli yakınlık karar vermemizi sağlar.  $C_i^* = 1$  ilgili karar noktasının ideal çözüme mutlak yakınlığını gösterirken,  $C_i^* = 0$  ise ilgili karar noktasının negatif ideal çözüme mutlak yakınlığını gösterir.  $C_i^*$  sayısal değeri 1'e en yakın olan alternatif en iyi alternatif,  $C_i^*$  sayısal değeri 0'a en yakın olan alternatif en kötü alternatif olarak değerlendirilir.

### 3.2.7. VIKOR Yöntemi

1973 senesinde Yu tarafından uzlaşık çözümün temelleri atılmıştır. 1982 yılında Zeleny yaptığı çalışmalarla bu metodu geliştirmiştir. Uzlaşık çözüm, ideal çözüme en yakın uygun çözümü ifade eder. Başka bir deyişle ortak bir karar üzerinde varılan anlaşmayı ifade etmektedir (Opricovic, Tzeng 2007: 515).

VIKOR yöntemi ile problem çözümleri 6 adımda yapılır Adımlar aşağıda sırasıyla belirtilmiştir (Opricovic, Tzeng 2004):

**Adım 1:** Başlangıç matrisi olan A Karar matrisi Eşitlik 63. de gibi gösterilmiştir.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad (63.)$$

Karar matrisine (Eşitlik 63.) göre her bir kriter için kriter ağırlık yönleri de hesaba katılarak en iyi  $f_j^*$  değerleri ( $j=1,2,\dots,n$ ) değerleri Eşitlik 64. kullanılarak hesaplanır.

$$f_j^* = \begin{cases} \max_i a_{ij}, & \text{eğer kriter yönü fayda ise} \\ \min_i a_{ij}, & \text{eğer kriter yönü maliyet ise} \end{cases} \quad (64.)$$

Karar matrisine (Eşitlik 63.) göre her bir kriter için kriter ağırlık yönleri de hesaba katılarak en kötü  $f_j^-$  ( $j=1,2,\dots,n$ ) değerleri Eşitlik 65. kullanılarak hesaplanır.

$$f_j^- = \begin{cases} \min_i a_{ij}, & \text{eğer kriter yönü fayda ise} \\ \max_i a_{ij}, & \text{eğer kriter yönü maliyet ise} \end{cases} \quad (65.)$$

**Adım 2:** Linear normalizasyon işlemi uygulanarak normalizasyon matrisi Eşitlik 18. yardımıyla hesaplanır ve R normalizasyon matrisi (Eşitlik 67.) oluşturulur.

$$r_{ij} = \frac{f_j^* - a_{ij}}{f_j^* - f_j^-} \quad (66.)$$

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \dots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix} \quad (67.)$$

**Adım 3:** Normalize karar matrisinde sütunlarda gösterilen kriterlerin ilgili kriter ağırlıklarıyla çarpılması yani Eşitlik 68. kullanılarak V ağırlıklandırılmış normalize karar matrisi Eşitlik 69. gösterildiği gibi oluşturulur.

$$v_{ij} = r_{ij} * w_j \quad (68.)$$

$$V = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & \vdots & \dots & \vdots \\ v_{m1} & v_{m2} & \dots & v_{mn} \end{bmatrix} \quad (69.)$$

**Adım 4:** Ortalama grup faydası  $S_i$  değerleri Eşitlik 70. kullanılarak hesaplanır.

$$S_i = \sum_{j=1}^n w_j \cdot \frac{f_j^* - a_{ij}}{f_j^* - f_j^-} \quad (70.)$$

Maksimum pişmanlık  $R_i$  değerleri Eşitlik 71. kullanılarak hesaplanır.

$$R_i = \max_j \left( w_j \cdot \frac{f_j^* - a_{ij}}{f_j^* - f_j^-} \right) \quad (71.)$$

**Adım 5:** Sıralama indeksi  $Q_i$  değerleri hesaplanmasında kullanılan  $S^*$  Eşitlik 72. ile,  $S^-$  Eşitlik 73. ile,  $R^*$  Eşitlik 74. ve  $R^-$  Eşitlik 75. ile hesaplanırlar.

$$S^* = \min_i S_i \quad (72.)$$

$$S^- = \max_i S_i \quad (73.)$$

$$R^* = \min_i R_i \quad (74.)$$

$$R^- = \max_i R_i \quad (75.)$$

$Q_i$  değerleri hesaplanırken kullanılan  $q$  parametresi maksimum grup faydasını gösterirken  $(1-q)$  ise karşıt görüştekilerin minimum pişmanlığının ağırlığını ifade eder.  $Q_i$  değerleri Eşitlik 76. kullanılarak hesaplanır.

$$Q_i = \frac{q.(S_i - S^*)}{S^- - S^*} + \frac{(1-q).(R_i - R^*)}{R^- - R^*} \quad (76.)$$

**Adım 6:**  $S_i$ ,  $R_i$  ve  $Q_i$  değerleri hesaplanır ve küçükten büyüğe doğru sıralanır. Üç tane sıralanmış liste oluşturulmuş olur. Sıralanmış listelerin doğruluğunu test etmek için minimum  $Q_i$  değerine sahip alternatifin kabul edilebilir avantaj ve kabul edilebilir istikrar koşulunu sağlayıp sağlamadığı kontrol edilir.

**Koşul 1:**  $Q_i$  değerleri küçükten büyüğe doğru sıralanır. İlk sırada yer alan alternatif  $A^1$  ile ikinci sırada yer alan alternatif  $A^2$  ile gösterilir. Eşitlik 77 ile kabul edilebilir avantaj koşulu hesaplanır.

$$Q(A^2) - Q(A^1) \geq DQ \quad (77.)$$

Eşitlik 78 kullanılarak DQ parametresi hesaplanır.  $m$  ile gösterilen alternatif sayısıdır.

$$DQ = \frac{1}{m-1} \quad (78.)$$

**Koşul 2:**  $Q_i$  (Kabul edilebilir istikrar) değerleri küçükten büyüğe sıralanır. İlk sırada yer alan alternatif  $A^1$ , S veya R değerlerine göre küçükten büyüğe yapılan sıralamada da minimum değere sahip en iyi alternatiftir.

**Sonuç:** İki koşulu birden gerçekleştiren alternatif önerilir. İki koşuldan bir tanesi sağlanmadığı zaman uzlaşık çözüm kümesi aşağıdaki şekilde oluşturulur.

Eğer koşul 2 sağlanmıyorsa  $A^1$  ve  $A^2$  alternatiflerinin her ikisi de uzlaşık çözüm olarak kabul edilir.

Eğer koşul 1 sağlanmıyorsa alternatiflerin tamamı uzlaşık en iyi çözüm kümesinde yer alır. Burada üst sınır değeri olan maksimum M, Eşitlik 79. ya göre belirlenir.

$$Q(A^m) - Q(A^1) < DQ \quad (79.)$$

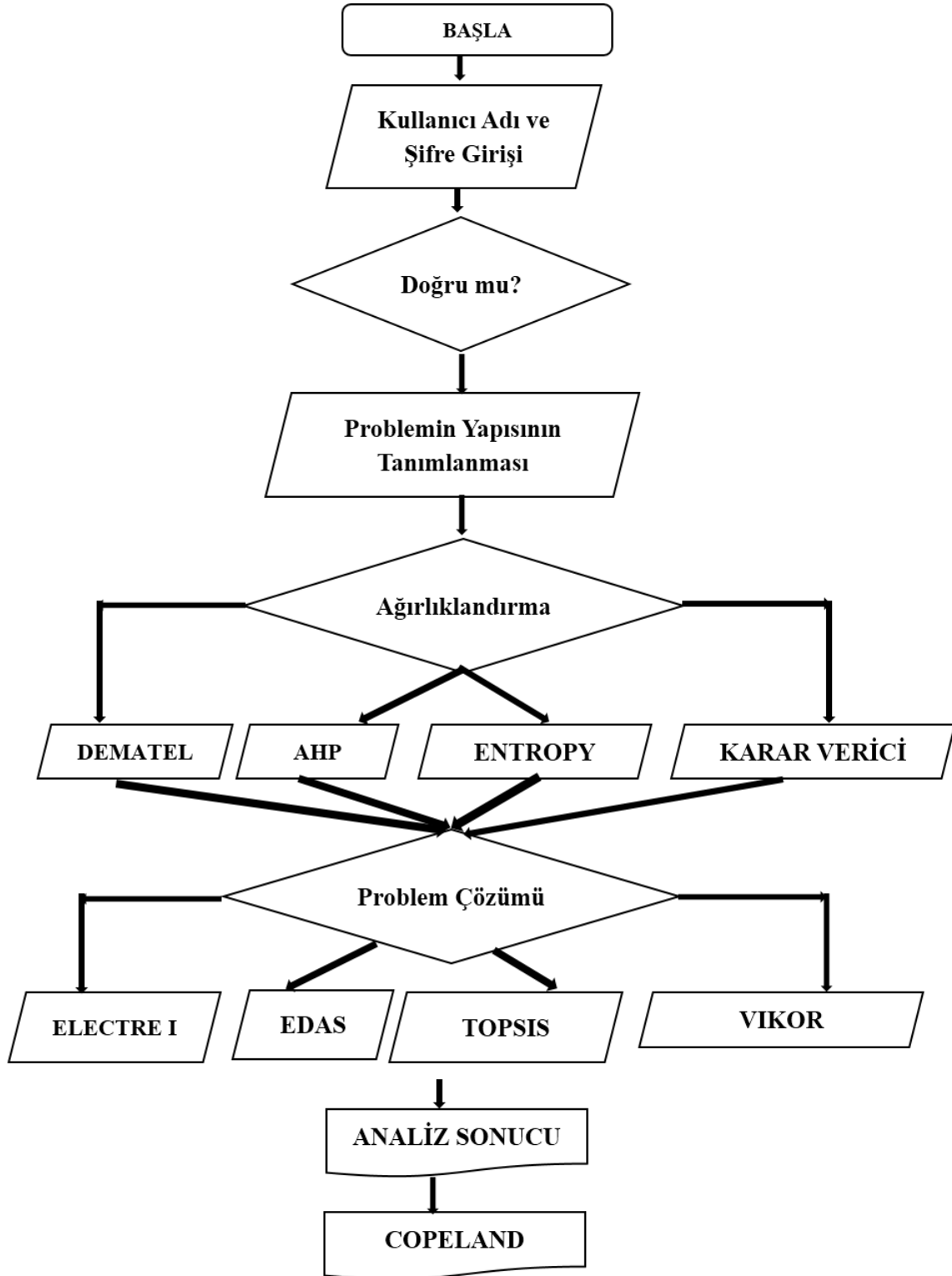
$Q$  değerlerine göre alternatifler sıralanır. En iyi alternatif, minimum  $Q$  değerine sahip alternatiflerden biridir.

### 3.3. Geliştirilen Otomasyonun Algoritması

Otomasyon 3 aşamadan oluşacak şekilde tasarlanmıştır.

1. aşamada problemin yapısı girilir.
2. aşamada probleme ait veriler problem yapısına göre sistem tarafından istenir.
3. aşama sonuç aşamasıdır. Tüm hesaplamalar bu aşamada yapılarak sonuç kullanıcıya detaylı şekilde tüm hesaplama basamaklarıyla gösterilir.

Otomasyonda 2 farklı hesaplama algoritması kullanılmıştır. Ağırlıklandırma yöntemleri hesaplamaları yapılır ve veri tabanına kaydedilir. Çözüm yöntemleri ise bu verileri kullanarak problem çözümünü gerçekleştirir. 2 farklı hesaplama yapılması ağırlıklandırma ve çözüm yöntemlerinin bağımsız kullanılmasına olanak sağlar. İstenilen ağırlıklandırma yaklaşımı ile istenilen çözüm yöntemi birlikte kullanılabilir. Şekil 3.2. de programın akış diyagramı şekil olarak verilmiştir.



Şekil 3.2. Akış Diyagramı

### 3.4.Otomasyonun Veri Tabanı Tasarımı

Geliştirilen otomasyonda MYSQL veri tabanı yönetim sistemi kullanılmıştır. Sistem verilerini kaydetmek için 9 tane tablo tasarlanarak oluşturulmuştur (Şekil 3.3.).





<input type="checkbox"/>	ckkv_agirlik	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır
<input type="checkbox"/>	ckkv_ahp	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır
<input type="checkbox"/>	ckkv_baslik	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır
<input type="checkbox"/>	ckkv_copeland	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır
<input type="checkbox"/>	ckkv_kullanici	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır
<input type="checkbox"/>	ckkv_siralama	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır
<input type="checkbox"/>	ckkv_site_ayar	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır
<input type="checkbox"/>	ckkv_veri	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır
<input type="checkbox"/>	ckkv_veri_dematel	★	Gözet	Yapı	Ara	Ekle	Boşalt	Kaldır

Şekil 3.3. MYSQL Veri tabanı Tablolar

Oluşturulan 9 tablodan 8 tanesi verileri kaydetmek için anahtar dizin tanımlanarak birbirleriyle ilişkilendirilmiştir (Şekil 3.4.). Bir tane tabloda ise sitenin genel ayarları kaydedilmiştir.

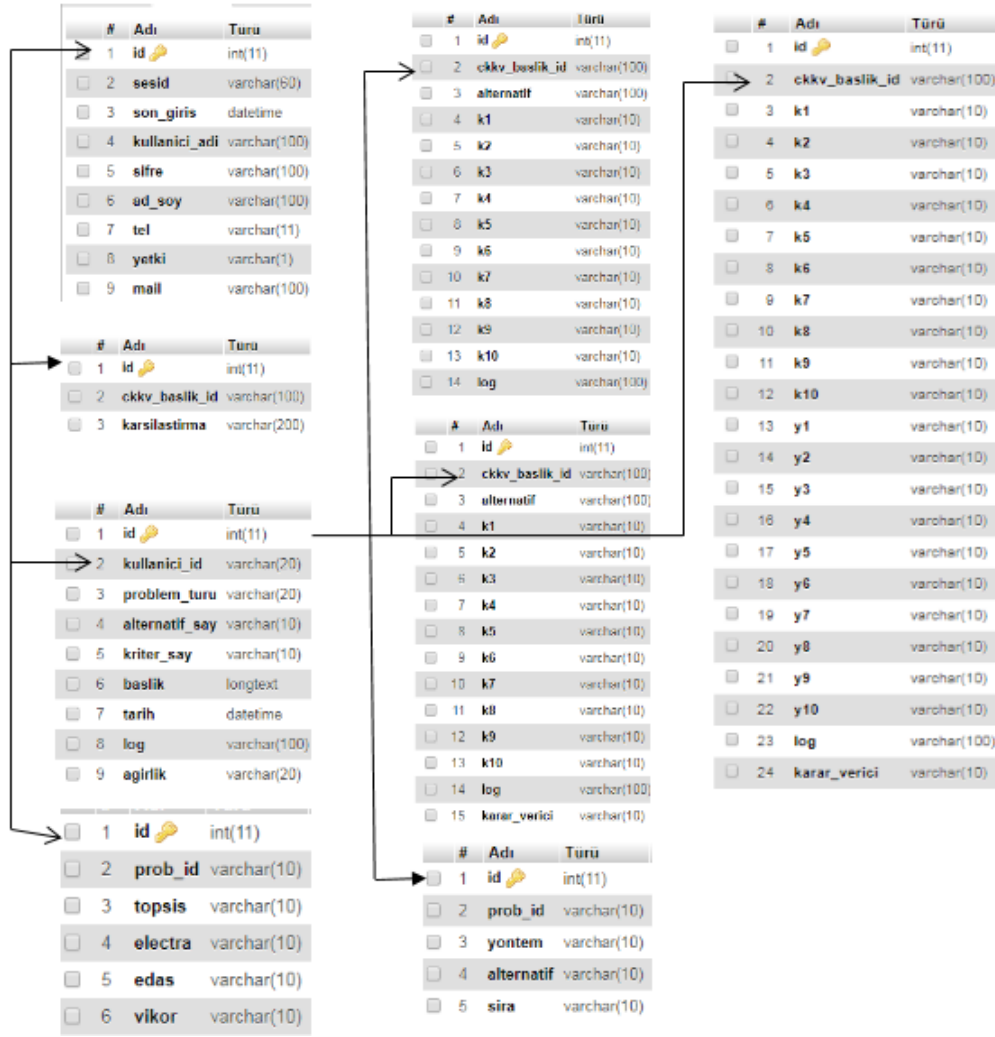
Tablo tasarımları yapılırken; otomasyonun her bölümünün dinamik olarak çalışmasını sağlayacak şekilde oluşturulmuştur. Verilerin korunması amacıyla kişisel bilgiler içeren veriler MD5 sistemine göre şifrelenmiştir.

Tabloların ilişkilendirilmesi aynı probleme farklı yöntem ve ağırlıklandırma yaklaşımları uygulanmasına olanak tanımaktadır. Tüm tablolar birbirleri ile etkileşim içinde olduğu için problem yapısındaki değişiklikler ve verilerdeki değişiklikler tüm tabloları etkilemekte ve güncellenen bilgilere göre işlem yapılmaktadır.

MySQL veri tabanı yönetiminin tablolara doğru ve kararlı bir şekilde işlem yapabilmesi, programın bir problem için kaydedilmiş farklı tablolardaki verilere ulaşabilmesi için indeksleme kısmı çok önemlidir. Bu sebeple tüm tablolar birbirleriyle ilişkilendirilecek şekilde indekslenmiştir.

Program yazılmadan önce veri tabanı tasarımı yapılmış hangi tabloda hangi veri olması gerektiği ve hangi tablonun hangi sütunu diğer tablolarla ilişkilendirilmesi gerektiği hesaplanmış ve otomasyon kodlanırken eksiklikler tamamlanmıştır.

Özenle hazırlanan veri tabanı tabloları sayesinde otomasyonun hızlı çalışması ve verilerin güvenli saklanması sağlanmıştır. Şekil 3.4. de tasarlanan tabloların içerikleri ve birbirleriyle olan ilişkilendirilmeleri verilmiştir.



Şekil 3.4. MYSQL Veri tabanı Tablo Yapıları ve İlişkileri

### 3.5. Çok Kriterli Karar Verme Yöntemleri Otomasyonu

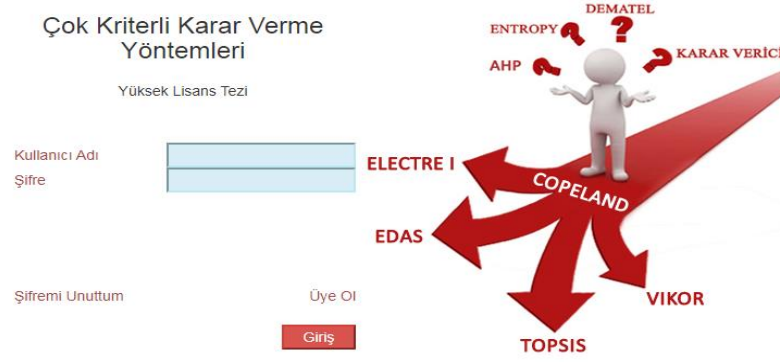
ÇKKV yöntemleri çevrimiçi sisteminde ağırlıklandırma yöntemleri olarak; AHP, Karar Verici, ENTROPY, DEMATEL yaklaşımları seçilmiştir. Problem çözüm için ise EDAS, ELECTRE I, TOPSIS, VIKOR yöntemleri kullanılmıştır.

Aynı problemi 4 farklı yöntemle analiz eden program bu sonuçları COPELAND yöntemi ile birleştirerek tek bir sıralamada verebilmektedir.

Kullanılan her yöntem için dinamik sayfalar oluşturulmuştur. Otomasyonun tamamı 12500 (onikibinbeşyüz) satır koddan ve 9 tane veri tabanı tablosundan oluşturulmuştur.

### 3.6. Geliştirilen Otomasyonun Kullanılması

Web sayfası açıldığında kullanıcı adı ve şifre Şekil 3.5.deki gibi istemektedir. Şifreler güvenlik amaçlı MD5 şifreleme sistemine göre kodlanmıştır. Saldırlara karşı güvenliği artırmak için Google doğrulama kodu eklenmiştir. Üyelik modülünden yönetici onayı ile üye olunmasına olanak sağlanmaktadır. Kullanıcı adı ve şifre girilerek sisteme giriş yapılır.



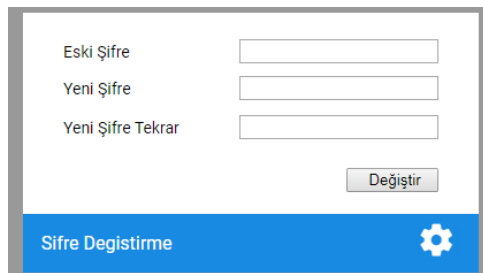
Şekil 3.5. Web Sayfası Giriş Ekranı

Kullanıcı adı ve şifre doğru şekilde girildikten sonra Google doğrulaması güvenlik için yapılarak sistemin ana sayfası Şekil 3.6. deki gibi karşımıza gelecektir.



Şekil 3.6. Otomasyon Ana Sayfası

Ana sayfada orta kısımda program ile ilgili bilgiler, sol tarafta menü ve sağ üst köşede şifre ayar (Şekil 3.7.) kısmı bulunmaktadır. Program kullanım kolaylığı göz önüne alınarak tasarlanmıştır.



Şekil 3.7. Şifre Değiştirme Ekranı

Sol tarafta bulunan menüden problem girişi seçilerek problem giriş ekranı açılır. Problem giriş ekranı Şekil 3.8. gösterilmiştir.

Sayın: Ahmet Turan ÖZYALÇIN    Sisteme Kayıtlı Problem Sayısı:9

**Problem Giriş Bölümü**

Problem Başlığı  
(Kısa açıklama girebilirsiniz)

Kriter Sayısı    Kriter Sayısını Seçiniz ▼

Alternatif Sayısı    Lütfen Alternatif Sayısını Seçiniz ▼

Alternatif Seçimi    Lütfen Alternatif Seçiniz ▼

Kriter Ağırlıklar    Lütfen Kriter Ağırlıklandırma Yöntemini Seçiniz ▼

Kaydet

Problem Girişi

Şekil 3.8. Problem Giriş Ekranı

Bu ekranda problemin başlığı, kriter sayısı, alternatif sayısı, alternatif seçimi ve kriter ağırlıkları bölümleri eksiksiz doldurularak kaydet butonuna basılır. Kaydedilen problemlerde daha sonra alternatif sayısı ve kriter sayısı bölümlerinde güncelleme yapılamaz. Problem başlığı, alternatif seçimi ve kriter ağırlıkları bölümünü kullanıcı daha sonra değiştirebilir.

Problem başlığı, alternatif seçimi ve kriter ağırlıkları bölümlerinin güncellenebilir olması aynı problemi farklı yöntemlerle sonuçlandırabilmemizi sağlar. Güncelleme yapılan problemler için girilen veriler kaybolmaz.

Sisteme kayıtlı kullanıcının eklediği problemlerin listelendiği kısımdır (Şekil 3.9.). Tanımlanması yapılmış tüm problemler bu alanda listelenir.

Sayın: Ahmet Turan ÖZYALÇIN    Sisteme Kayıtlı Problem Sayısı:3

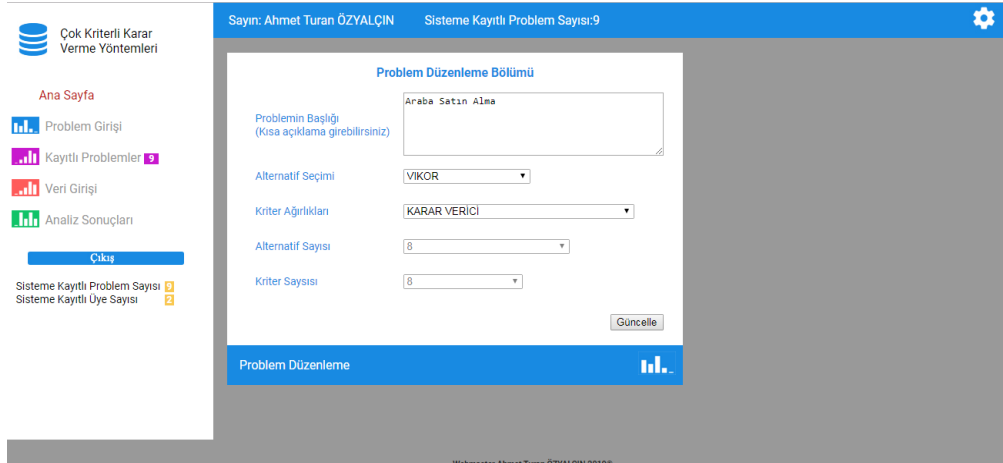
**Kayıtlı Problemler**

Problem Başlığı	Alternatif Seçimi	Kriter Ağırlıkları	Alternatif Sayısı	Kriter Sayısı	Tarih	Sil	Düzenle
Ortaokul 7.Sınıf Matematik Soru Bankası Kitabı Seçimi(DEMATEL)	TOPSIS	DEMATEL	6	6	2019-11-02 20:19:05		
Ortaokul 7.Sınıf Matematik Soru Bankası Kitabı Seçimi(Karar Verici)	VIKOR	Karar Vericiye Göre	6	6	2019-11-02 19:53:25		
Ortaokul 7.Sınıf Matematik Soru Bankası Kitabı Seçimi(ENTROPY)	ELECTRE I	ENTROPY	6	6	2019-10-28 22:40:11		

Kayıtlı Problemleri Listele

Şekil 3.9. Kayıt Listele Bölümü

Bu bölümde kayıt listeleme, silme ve düzenleme işlemleri yapılabilir. Şekil 3.10. da gösterilen problem düzenleme kısmında, problemin alternatif veya kriter ağırlıkları bölümü güncellenerek farklı yollarla çözümü yapılabilir.



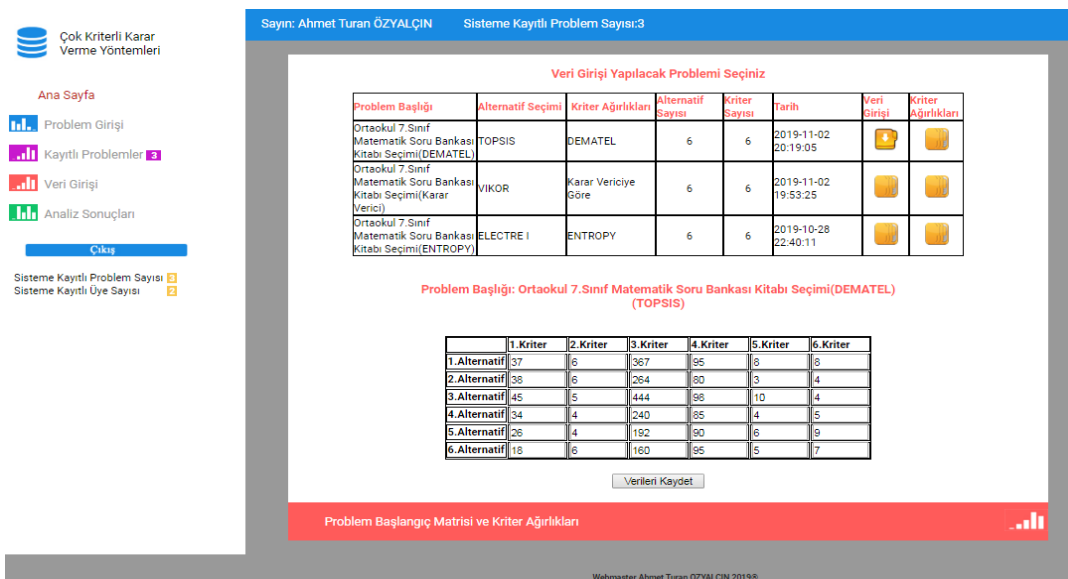
Şekil 3.10. Problem Düzenleme Bölümü

Veri girişi bölümü tıkladığı zaman şekil 3.11. de gösterildiği gibi problemlerin listelendiği sayfa açılır. Kullanıcını sisteme eklediği tüm problemler bu alanda da listelenir.



Şekil 3.11. Veri Girişi Sayfası

Şekil 3.12. de gösterilen sayfada problem verilerinin girişi yapılır.



Şekil 3.12. Problem Veri Girişi

Ağırlıklandırma veri girişi seçilen yöntemle göre farklılık gösterir. Şekil 3.13. de problem ağırlıklandırma yöntemi olarak karar verici girişi seçilmiş olan problemin, ağırlıklandırma veri girişi gösterilmiştir.

**Problem Başlığı: Ortaokul 7.Sınıf Matematik Soru Bankası Kitabı Seçimi(Karar Verici)  
(VIKOR)**

Ağırlık Hesaplama Yöntemi	Karar Verici Girişi					
Karar Verici(Kriter Ağırlıkları)	1. Kriter	2. Kriter	3. Kriter	4. Kriter	5. Kriter	6. Kriter
	0.15	0.10	0.15	0.15	0.25	0.20
Kriter Yönü	1. Kriter	2. Kriter	3. Kriter	4. Kriter	5. Kriter	6. Kriter
	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet

Problem Başlangıç Matrisi ve Kriter Ağırlıkları

Şekil 3.13. Karar Verici Kriter Ağırlık ve Yönleri Girişi

Şekil 3.14. de problem ağırlıklandırma yöntemi olarak AHP ağırlıklandırma yaklaşımı girişi seçilmiş olan problemin, ağırlıklandırma veri girişi gösterilmiştir.

Karşılaştırmalarda Kullanılan Önem Derecesi Tablosu	
1	Eşit Derecede Önemli
3	Orta Derecede Önemli
5	Kuvvetli Derecede Önemli
7	Çok Kuvvetli Derecede Önemli
9	Mutlak Derecede Önemli
2,4,6,8	Ara Değerleri Temsil Etmektedir.

AHP Ağırlıklandırma Yöntemi					
İkili Karşılaştırma Önem Dereceleri	1.kriter -> 2.kriter <input type="text" value="1/1"/>				
	1.kriter -> 3.kriter <input type="text" value="1/1"/>				
	1.kriter -> 4.kriter <input type="text" value="1/1"/>				
	1.kriter -> 5.kriter <input type="text" value="1/3"/>				
	1.kriter -> 6.kriter <input type="text" value="1/3"/>				
	2.kriter -> 3.kriter <input type="text" value="1/1"/>				
	2.kriter -> 4.kriter <input type="text" value="1/1"/>				
	2.kriter -> 5.kriter <input type="text" value="1/1"/>				
	2.kriter -> 6.kriter <input type="text" value="1/1"/>				
	3.kriter -> 4.kriter <input type="text" value="1/1"/>				
	3.kriter -> 5.kriter <input type="text" value="1/3"/>				
	3.kriter -> 6.kriter <input type="text" value="1/3"/>				
	4.kriter -> 5.kriter <input type="text" value="1/3"/>				
	4.kriter -> 6.kriter <input type="text" value="1/3"/>				
	5.kriter -> 6.kriter <input type="text" value="1/2"/>				
1. Kriter	2. Kriter	3. Kriter	4. Kriter	5. Kriter	6. Kriter
<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet	<input type="radio"/> Fayda <input checked="" type="radio"/> Maliyet

AHP Yöntemi Kriterlerin Karşılaştırma Değerleri

Şekil 3.14. AHP Kriter Ağırlık ve Yönleri Girişi

ENTROPY ağırlıklandırma yaklaşımında sadece kriter yönleri Şekil 3.15. de gösterildiği ekrandan seçmek yeterlidir.

Ağırlık Hesaplama Yöntemi	ENTROPY Yöntemi					
Kriter Yönü	1. Kriter	2. Kriter	3. Kriter	4. Kriter	5. Kriter	6. Kriter
	<input type="radio"/> Fayda	<input type="radio"/> Fayda	<input type="radio"/> Fayda	<input type="radio"/> Fayda	<input type="radio"/> Fayda	<input type="radio"/> Fayda
	<input checked="" type="radio"/> Maliyet	<input checked="" type="radio"/> Maliyet	<input checked="" type="radio"/> Maliyet	<input checked="" type="radio"/> Maliyet	<input checked="" type="radio"/> Maliyet	<input checked="" type="radio"/> Maliyet

Verileri Kaydet

Problem Başlangıç Matrisi ve Kriter Ağırlıkları

Şekil 3.15. ENTROPY Kriter Yönleri Girişi

Şekil 3.16. da problem ağırlıklandırma yöntemi olarak DEMATEL ağırlıklandırma yaklaşımı girişi seçilmiş olan problemin, ağırlıklandırma veri girişi gösterilmiştir. DEMATEL yaklaşımında birden fazla karar verici verileri girilebilir.

1. Kriter	2. Kriter	3. Kriter	4. Kriter	5. Kriter	6. Kriter
<input type="radio"/> Fayda	<input checked="" type="radio"/> Fayda	<input type="radio"/> Fayda	<input checked="" type="radio"/> Fayda	<input type="radio"/> Fayda	<input type="radio"/> Fayda
<input checked="" type="radio"/> Maliyet	<input type="radio"/> Maliyet	<input type="radio"/> Maliyet	<input type="radio"/> Maliyet	<input type="radio"/> Maliyet	<input type="radio"/> Maliyet

Karar Verici Sayısı

Karar Verici Sayısı	Karar Verici Verilerini Görüntülemek için <a href="#">TIKLAYINIZ</a>
	2 ▾

Verileri kaydet

Şekil 3.16. DEMATEL Kriter Yönleri ve Karar Verici Sayısı Girişi

Şekil 3.17. de ağırlıklandırma yöntemi olarak DEMATEL yaklaşımı seçilen ve 2 karar verici verisi girilmesi gereken problemin veri giriş sayfası gösterilmiştir. Veri toplama işlemi matris şeklinde tasarlanmıştır. Bu şekilde tasarlanması karar verici sayısının istediğimiz ölçüde artmasını sağlamıştır.

Karar Verici Verileri

1. Karar Verici						
	1. Kriter	2. Kriter	3. Kriter	4. Kriter	5. Kriter	6. Kriter
1. Kriter	0	2	2	4	3	4
2. Kriter	3	0	3	3	2	3
3. Kriter	4	3	0	3	4	4
4. Kriter	1	1	2	0	4	4
5. Kriter	3	1	1	4	0	2
6. Kriter	2	3	1	2	2	0

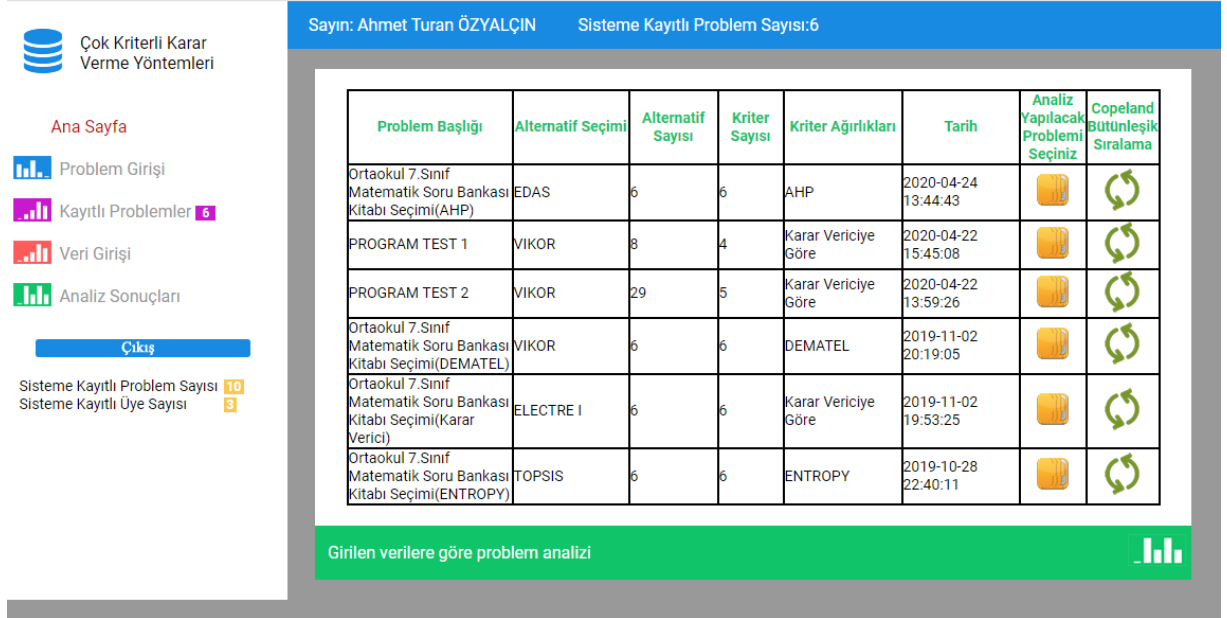
2. Karar Verici						
	1. Kriter	2. Kriter	3. Kriter	4. Kriter	5. Kriter	6. Kriter
1. Kriter	0	2	3	4	3	4
2. Kriter	3	0	3	3	3	2
3. Kriter	3	3	0	2	4	3
4. Kriter	1	2	1	0	3	3
5. Kriter	3	1	1	4	0	3
6. Kriter	2	3	2	2	2	0

Şekil 3.17. DEMATEL Karar Verici Veri Girişi

Sonuçların hesaplandığı bölüm Şekil 3.18. de gösterilmiştir. Bu bölümde 2 farklı yoldan analiz yapılabilir.

1.yol problemin yapısına göre seçilen ağırlıklandırma modeli ve seçilen yöntem kullanılarak problemimiz analiz edilir ve tüm adımlar tablolar halinde ekrana yazdırılır.

2.yol ise problemimizi seçtiğimiz ağırlıklandırma modeline göre sistemde bulunan 4 yöntemle analiz ettirip sonucunu COPELAND yöntemiyle birleştirmemizi sağlar.



Şekil 3.18. Analiz Sonuçları Sayfası

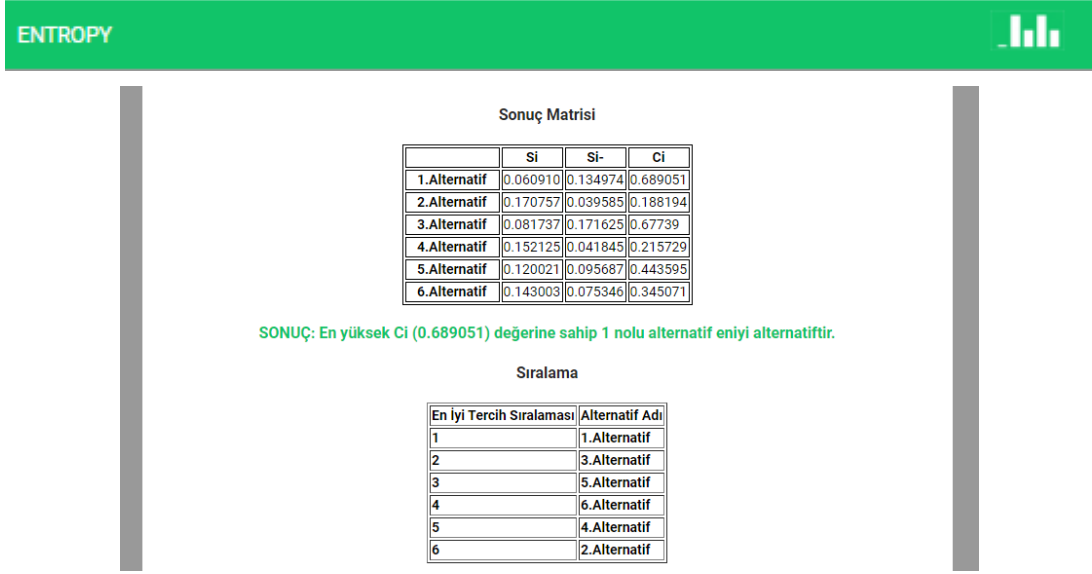
Bu sayfada önce kriter ağırlıklandırma yöntemine göre kriter ağırlıkları hesaplanır bu ağırlıklara göre seçilen yöntemle sonuca ulaşılır. Problem düzenle bölümünden problem yapısı değiştirilerek farklı yöntemlerle sonuçlar hesaplatılabilir. Hesaplamalarda oluşturulan tüm tablolar kullanıcıya gösterilir.

Şekil 3.19. da örnek olarak ENTROPY ağırlıklandırma yaklaşımı ve TOPSIS çözüm yöntemiyle çözülmüş bir problemin ekran çıktısı verilmiştir.



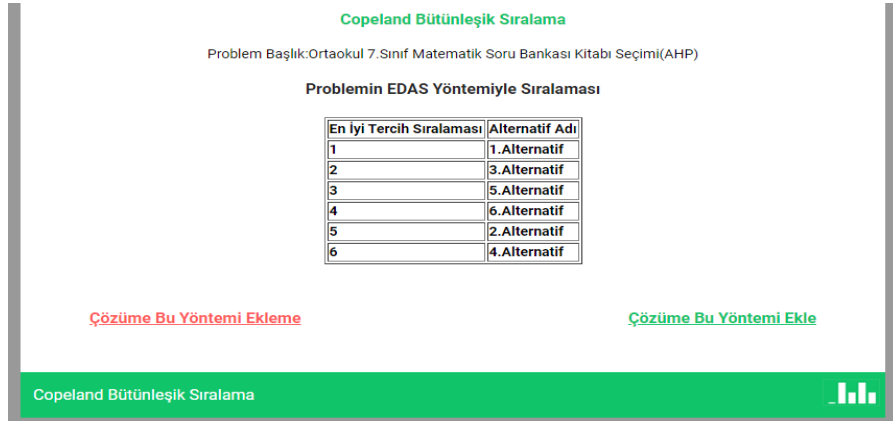
### Kriter Ağırlıkları

w1	w2	w3	w4	w5	w6
0.153705	0.063213	0.250798	0.009987	0.317065	0.205231



Şekil 3.19. ENTROPY ve TOPSIS ile Yapılan Analiz Sonucu

COPELAND bütünleşik sıralama yöntemi seçildiği zaman problemimizin çözümünde hangi yöntemlerin kullanılacağı bilgisi kullanıcıdan tek tek sıralamalar gösterilerek alınır. Şekil 3.20. de gösterilen program görüntüsü COPELAND yöntemi seçildiğinde karşımıza gelen ilk ekrandır.



Şekil 3.20. COPELAND Bütünleşik Sıralama için Yöntem Seçimi

Çözümü yönteme ekle veya çözümü yönteme ekleme butonları kullanılarak ilerlenir. Otomasyonda bulunan 4 çözüm yöntemi bu ekranda kullanıcıya listelenir. Son aşamada seçilen yöntemler ve bu yöntemlerde problemimizin sonuçları gösterilir. Şekil 3.21. de örnek olarak seçilen bir problemin ekran çıktısı verilmiştir.

### Copeland Bütünleşik Sıralama

Problem Başlık:Ortaokul 7.Sınıf Matematik Soru Bankası Kitabı Seçimi(AHP)

EDAS	<b>Eklenmedi</b>
ELECTRA I	<b>Eklendi</b>
TOPSIS	<b>Eklendi</b>
VIKOR	<b>Eklendi(q=0.50)</b>

#### SEÇİLEN YÖNTEMLERDE SIRALAMA SONUÇLARI

	ELECTRA I	TOPSIS	VIKOR
1.Alternatif	1	1	1
2.Alternatif	5	5	6
3.Alternatif	3	2	4
4.Alternatif	6	6	5
5.Alternatif	2	3	3
6.Alternatif	4	4	2

Şekil 3.21. COPELAND Bütünleşik Sıralama için Seçilen Yöntemler

Listelenen sonuçlar COPELAND yöntemi ile birleştirilerek tek bir sıralama elde edilir. Şekil 3.22. de COPELAND yönteminin sonuç ekranı verilmiştir. Bu işlemler istenilen kadar tekrar edilerek farklı yöntemler dâhil edilip çıkarılarak sonuçlara ulaşılabilir.

Alternatiflerin Copeland Puanlarına Göre Sıralanması

En İyi Tercih Sıralaması	Alternatif Adı
1	1.Alternatif
2	5.Alternatif
3	3.Alternatif
4	6.Alternatif
5	2.Alternatif
6	4.Alternatif

SEÇİLEN YÖNTEMLERDE VE COPELAND PUANLARINA GÖRE SIRALAMA SONUÇLARI

	ELECTRA I	TOPSIS	VIKOR	COPELAND
1.Alternatif	1	1	1	1
2.Alternatif	5	5	6	5
3.Alternatif	3	2	4	3
4.Alternatif	6	6	5	6
5.Alternatif	2	3	3	2
6.Alternatif	4	4	2	4

**Tekrar Yöntem Seç**

Şekil 3.22. COPELAND Bütünleşik Sıralama Sonuçları

4 farklı kriter ağırlıklandırma ve 4 farklı ÇKKV yöntemi kullanan sayfada tüm kombinasyonlarla sonuca ulaşılabilir. 16 farklı yaklaşımla sonuç listelenebilir. Otomasyonda kullanılabilecek ağırlıklandırma yaklaşımları ve çözüm yöntemleri Tablo 3.4. de gösterilmiştir. Aynı zamanda problemimizin sonucu COPELAND bütünleşik sıralama yöntemi ile birleştirilebilir.

Tablo 3.4. Problem Çözüm Yöntemleri

S.N.	Kriter Ağırlıklandırma	Problem Çözüm Yöntemi
1	AHP	EDAS
2	DEMATEL	
3	ENTROPY	
4	Karar Verici	
1	AHP	ELECTRE I
2	DEMATEL	
3	ENTROPY	
4	Karar Verici	
1	AHP	TOPSIS
2	DEMATEL	
3	ENTROPY	
4	Karar Verici	
1	AHP	VIKOR
2	DEMATEL	
3	ENTROPY	
4	Karar Verici	

#### 4. SONUÇ VE ÖNERİLER

Geliştirilen otomasyon <http://ckkvtez.a0001.net/> adresinde ücretsiz kullanıma açılmıştır. Otomasyonun ücretsiz hizmet sağlayan sunucularda barındırılması sayesinde maliyetsiz olarak kullanıma sunulabilme imkânı sağlanmıştır. Otomasyon 4 ağırlıklandırma yaklaşımı ve 4 çözüm yöntemi kullanarak kullanıcıya bir problemin çözümünü 16 farklı şekilde çözüme imkânı sağlamıştır. Otomasyon sadece sonucu göstermemiştir. Seçilen ağırlıklandırma yaklaşımı ve çözüm yönteminin tüm adımlarını kullanıcıya sunmuştur. Çok uzun zaman alan bu işlemleri saniyeler içinde hesaplamıştır. Bu sayede daha fazla problemin farklı yollarla çözülmesi veya çözülen problemlerin test edilmesi imkânı sağlanmıştır. Oluşturulan otomasyon bilişim araçlarının etkili ve verimli kullanılmasına örnek teşkil etmiştir. Daha çok çalışma yapılabilmesine olanak sağlamıştır.

Kullanıcının problemine istedikleri zaman erişebilmesine olanak sağlamak amacıyla veri tabanına sonuçlar ve tüm adımlar kayıt edilmiştir. Kullanıcının internet erişimi olan tüm cihazlardan istedikleri zaman problemini ve çözümünü görebilmesi sağlanmıştır.

Geliştirilen otomasyonun MYSQL veri tabanında 9 adet tablo kullanılmıştır. Ağırlıklandırma yaklaşımlarında elde edilen veriler veri tabanında ayrılan tabloya kaydedilerek ağırlıklandırma yaklaşımlarıyla çözüm yöntemlerinin bağımsız çalışabilmesi sağlanmıştır. Her yöntem için farklı

dinamik sayfalar oluşturularak farklı verilerle çalışabilmesi, farklı hesaplanmaların yapılabilmesinin sağlanması otomasyonun güçlü yönüdür.

Dinamik sayfalar yaklaşık olarak 12500 (onikibinbeşyüz) satır özgün kod içermektedir. Bu kodlar ile kullanılan çok kriterli karar verme yöntemleri için yeni algoritmalar geliştirilmiştir. Geliştirilen otomasyonun üyelik tabanlı olarak çevrimiçi çalışması, erişimi kolaylaştırmıştır. Verilerin sunucu tabanlı veri tabanında tutulması, kullanılan elektronik cihazda herhangi bir veri kaydedilmemesi ve herhangi bir kurulum istenmemesi otomasyonun güçlü yönlerindedir. Ayrıca veri girişleri esnek halde olması istenilen boyutta matris girilmesine olanak sağlamıştır.

Otomasyon geliştirilirken temel yaklaşımlar kullanılmıştır. Her yaklaşım için farklı algoritmalar kullanılmaktadır. İleriki dönemlerde tüm yöntemleri kapsayacak hale getirilerek kullanımın artması sağlanabilir. Kullanıcı anketleri yapılarak yazılımda güncellemeler yapılabilir.

Otomasyonun online olması kullanıcılar büyük avantaj sağlamakla beraber yayımcı açısından maddi külfet getirmektedir. Bu sorun ücretsiz hizmetler kullanılarak çözülmüştür. Ücretsiz barınma hizmeti alınması veri güvenliğini ve sürekliliğini risk altında bırakmıştır. Otomasyon daha da geliştiğinde içerisinde reklam yayınlanarak gelir elde edilip ücretli sunucularda yayına geçirilebilir.

## KAYNAKÇA

- Ağaç., G. & Baki, B. (2016). Sağlık alanında çok kriterli karar verme teknikleri kullanımı: Literatür incelemesi. Hacettepe Sağlık İdaresi Dergisi, 19(3).
- Aksakal E. ve Dağdeviren M. (2010). “ANP ve DEMATEL Yöntemleri ile Personel Seçimi Problemine Bütünleşik Bir Yaklaşım”, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, Cilt 25, Sayı 4, ss. 905-913, Ankara, 2010.
- Arslan, R., (2018), Çok Kriterli Karar Verme Yöntemlerinin Karşılaştırılması ve Bütünleştirilmesi: OECD Verileri Üzerine Bir Uygulama, Cumhuriyet Üniversitesi Sosyal Bilimleri Enstitüsü, Doktora Tezi, 184 sayfa, Sivas, 2018.
- Çatı K., Eş A. ve Özevin O. (2017). “Futbol Takımlarının Finansal ve Sportif Etkinliklerinin Entropi ve TOPSIS Yöntemiyle Analiz Edilmesi: Avrupa'nın 5 Büyük Ligi ve Süper Lig Üzerine Bir Uygulama”, International Journal of Management Economics & Business, 13(1), 199-222.
- Çetinbaş, M. (2017), Çok Kriterli Karar Verme Yöntemleriyle Excel Uygulamaları. Pamukkale Üniversitesi İşletme ve Bilişim Yönetimi Dergisi, 4(1), 12-29.
- Çınar, Y. (2013), Kariyer Tercihi Probleminin Yapısal bir Modeli ve Riske Karşı Tutumlar: Olasılıklı DEMATEL Yöntemi Temelli Bütünleşik bir Yaklaşım, Sosyoekonomi Dersisi, 19, 157-186.
- Dalbudak, E., & Rençber, Ö. F. Çok Kriterli Karar Verme Yöntemleri Üzerine Literatür İncelemesi. Gaziantep Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 4(1), 1-17.
- Demir, G & vd. (2021) Çok Kriterli Karar Verme Yöntemleri ve ÇKKV Yazılımı ile Problem Çözümü, Nobel Akademik Yayıncılık, 1.baskı, 2021.
- Dursun, B., (2018), TOPSIS ve ELECTRE Yöntemi İle Tedarikçi Seçimi: Kozmetik Sektöründe Bir Uygulama, Bahçeşehir Üniversitesi Sosyal Bilimleri Enstitüsü, Yüksek Lisans Tezi, 71 sayfa, İstanbul, 2018.
- Gabus ve Fontela (1972). “World Problems: An Invitation to Further Thought Within The Framework of DEMATEL”, Switzerland, Geneva: Battelle Geneva Research Centre, 1972.
- Keshavarz Ghorabae, M., Zavadskas, E. K., Olfat, L., & Turskis, Z. (2015). Multi-criteria inventory classification using a new method of evaluation based on distance from average solution (EDAS). Informatica, 26(3), 435-451.

- Hwang, C. L., and K. Yoon. 1973. Multiple Attribute Decision Making, Methods and Applications. Lecture Notes in Economics and Mathematical Systems. Vol. vol.186. Now York: Springer-Verlag.
- Karaca, E., (2013), ELECTRE I ve TOPSIS Yöntemlerini Kullanarak Bir Otomotiv Firması İçin Bayi Seçimi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 95 sayfa, Kocaeli, 2013.
- Manisalı, E. (1981). Yatırım Projelerinin Değerlendirilmesinde Çok Ölçütlü Model Yaklaşımı, İ.T.Ü Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul, 1981.
- Opricovic, S., & Tzeng, G. H. (2004). Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. European journal of operational research, 156(2), 445-455.
- Opricovic, S., & Tzeng, G. H. (2007). Extended VIKOR method in comparison with outranking methods. European journal of operational research, 178(2), 514-529.
- Ömürbek, V., & Kınay, B. (2013). Havayolu Taşımacılığı Sektöründe Topsis Yöntemiyle Finansal Performans Değerlendirmesi. Süleyman Demirel Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi, 343-363.
- Özbek, A., (2017), Çok Kriterli Karar Verme Yöntemleri ve Excel İle Problem Çözümü, Seçkin Yayıncılık,1.baskı, Kırıkkale, 2017.
- Özdağoğlu, A., Yakut, E., & Bahar, S. (2017). Machine selection in a dairy product company with entropy and SAW methods integration. Dokuz Eylül Üniversitesi İktisadi İdari Bilimler Fakültesi Dergisi, 32(1), 341-359.
- Roy, B. (1968). "Classement et choix en presence de points de vue multiples (la method Electre)". Rev Fr Inf Resch Oper 2: 57-75.
- Saaty, T. (1986). "Axiomatic Foundations Of The Analytic Hierarchy Process", Management Science, 32(7), (1986), s. (841-855).
- Senvar O., Tuzkaya U. R. ve Kahraman C., (2014), Supply Chain Performance Measurement: An Integrated DEMATEL and Fuzzy-ANP Approach, Supply Chain Management Under Fuzziness, Volume 313, ss.143-165.
- Shemshadi, A., Shirazi, H., Toreihi, M., & Tarokh, M. J. (2011). A fuzzy VIKOR method for supplier selection based on entropy measure for objective weighting. Expert Systems with Applications, 38(10), 12160-12167.
- Tsai W.H. ve Chou W.C., (2009), Selecting Management Systems for Sustainable Development in SMES:A Novel Hybrid Model Based on DEMATEL, ANP and ZOGP, Expert Systems with Applications, Volume 36, Issue 2, ss.1444-1458.
- Turan, G. (2018). Çok Kriterli Karar Verme. B. F. Yıldırım, & E. Önder içinde, Çok Kriterli Karar Verme Yöntemleri (s. 15-20). Bursa: Dora Basım Yayınları.
- Tzeng, G.-H. ve Huang, J.-J. (2011), Multiple Attribute Decision Making: Methods and Applications; CRC Press: Boca Raton, FL, USA, A Chapman and Hall Book.
- Wang, Y.M., 1998. Using the Method of Maximizing Deviations to Make Decision for Multi-indices. Systems Engineering and Electronics, 7(31), 24-26.
- WEB\_1,International Society on MCDM <https://www.mcdmsociety.org/content/software-related-mcdm-0>.
- WEB\_2, W3Techs,  
[http://w3techs.com/technologies/overview/programming\\_language/all,01/10/2022](http://w3techs.com/technologies/overview/programming_language/all,01/10/2022)

WEB\_3,MySQL 8.0.18 Release Notes,  
<https://dev.mysql.com/doc/relnotes/mysql/8.0/en/,04/10/2022>.  
WEB\_4,solid IT, Method of calculating the scores of the DB-Engines Ranking,  
<https://db-engines.com/en/ranking, 21/10/2022>

# Yükseköğretimde Yapay Zekânın Rolü: Anatomi Dersi için ChatGPT Değerlendirmesi

\*\*\*

## The Role of Artificial Intelligence in Higher Education: ChatGPT Assessment for Anatomy Course

Tarık TALAN<sup>1</sup> 

Yusuf KALINKARA<sup>2</sup> 

DOI:10.33461/uybisbbd.1244777

### Makale Bilgileri

#### Makale Türü:

Araştırma Makalesi

#### Geliş Tarihi:

30.01.2023

#### Kabul Tarihi:

06.03.2023

©2023 UYBISBBD  
Tüm hakları saklıdır.



### Öz

Uzun bir süre boyunca yürütülen kapsamlı araştırmalar, yapay zeka (AI) teknolojisinin hızla ilerlemesiyle sonuçlanmış ve çok sayıda uygulamanın geliştirilmesine yol açmıştır. Özellikle yakın zamanda piyasaya sürülen AI uygulamalardan biri de ChatGPT'dir. Kısa süre içerisinde milyonlarca kullanıcının beğenisini kazanan ChatGPT, metin okuyup yazabilen AI kodudur. Bu çalışmanın amacı, ChatGPT'nin bir dersteki performansını lisans öğrencilerinin performansıyla karşılaştırmaktır. Katılımcılar Türkiye'de bir devlet üniversitesinin Sağlık Bilimleri Fakültesi'nde daha önce anatomi dersi almış öğrencilerdir. Sınav çoktan seçmeli test şeklinde olup 40 sorudan oluşmaktadır. ChatGPT'nin öğrencilerden daha iyi performans gösterdiği tespit edilmiştir.

**Anahtar Kelimeler:** Yapay zeka, ChatGPT, Sağlık eğitimi, Anatomi.

### Abstract

The extensive research conducted over a prolonged period of time has resulted in the rapid advancement of artificial intelligence (AI) technology, leading to the development of numerous applications. One such recent AI application is ChatGPT, an AI chatbot that has gained millions of users in a short span of time, and can read and write texts. The aim of this study is to compare the performance of ChatGPT in an anatomy course with that of undergraduate students. The participants were students from the Faculty of Health Sciences at a state university in Turkey, who had previously taken an anatomy course. The examination was in the form of a multiple-choice test consisting of 40 items. It was found that ChatGPT outperformed the students in the examination.

**Keywords:** Artificial intelligence, ChatGPT, Medical education, Anatomy.

©2023 UYBISBBD  
All rights reserved.



**Atf/ to Cite (APA):** Talan, T., & Kalinkara, Y. (2023). The role of artificial intelligence in higher education: ChatGPT assessment for anatomy course. *International Journal of Management Information Systems and Computer Science*, 7(1), 33-40.

<sup>1</sup> Asst. Prof., Gaziantep Islam Science and Technology University, ttalan46@hotmail.com

<sup>2</sup> Lecturer, Gaziantep Islam Science and Technology University, yusufkalinkara@gmail.com

## 1. INTRODUCTION

As scientific knowledge continues to grow exponentially, new technological developments emerge every day. These technological developments and changes have the potential to facilitate, transform, and improve our lives, and can bring great benefits to the fields in which they are used. In fact, it is difficult to think of an area that is not affected by technology today. However, as technology continues to rapidly advance, questions are being raised about how it can be effectively used in various fields and what impact it will have. One area that has seen significant investment in recent years is education, with virtual reality, augmented reality, metaverse, blockchain, simulation, mobile technologies, robotics and automation, and online learning environments all being implemented. Among these technological advancements, artificial intelligence (AI) stands out as one of the most successful and widely-used technologies in many sectors.

The primary objective of AI is to enhance the comprehension of human intelligence and to augment machine intelligence to attain maximum benefit from them (Tektaş, Akbaş & Topuz, 2002). AI is a wide-ranging domain that is constantly advancing and leading the way in technological progress (Büyükgöze & Dereli, 2019). Although various research endeavors have been undertaken in AI across multiple fields, its impact on education has also been investigated. AI is applied across several disciplines, including law, science, mathematics, health, engineering, and architecture (Korkmaz & Büyükgöze, 2019; Taşçı & Çelebi, 2020). Research in this field is gaining momentum, and progress is being made through continued research and development activities.

Over the years, AI technologies have evolved and taken various forms. Despite the fact that AI research has a long history, these systems have now become an essential part of human life thanks to the investments made over the years and the widespread use of technological tools such as the internet and smart devices. Currently, there are numerous AI technologies, most of which are still in the research stage. In recent years, AI technologies have been applied in diverse fields, including smart cities, smart watches, robotics, drone systems, defense industry, cybersecurity, and healthcare (Sarica, 2021; Talan, 2021).

However, the potential use of AI in education, its contribution to education, and its impact on education are still subjects of debate, with numerous predictions and considerations. While developments in AI offer significant opportunities for the education sector, they also pose a threat at times. Thus, AI technology needs to be carefully considered and evaluated in many ways. AI's potential to be one of the most important technologies of the future increases when the potential risks and benefits it offers are evaluated, and the necessary precautions are taken.

### 1.1 Artificial Intelligence (AI)

AI, first introduced by John McCarthy at the Dortmund Conference in 1956, has emerged as a significant technological advancement in recent times (Arslan, 2017). AI is an interdisciplinary field that has developed through the integration of various disciplines, including computer science, control theory, information theory, neurophysiology, psychology, linguistics, and philosophy (Wei, 2018). The primary objective of AI is to simulate and enhance human intelligence, and remarkable progress has been made in this domain over the years (Shi & Zheng, 2006).

AI can be defined as a computer-based simulation of human mental processes, such as reasoning, argumentation, learning, sense-making, communication, decision-making, and generalization (Akyürek, 2013). In essence, AI is an analytical system that aims to imitate life (Gordon, 2011). While several definitions of AI exist in the literature, they focus on different concepts, including human-like thinking, intelligent programming, rational action, and humanoid responses (Arslan, 2017; Büyükgöze & Dereli, 2019; Russel & Norvig, 2010).

AI is transforming human life in various ways, ranging from internet search engines and smartphone apps to public transportation, autonomous cars, and household appliances. AI has become an integral part of people's lives, and its applications are increasingly being used as control and



decision-making mechanisms in various fields, including health, engineering, architecture, military, psychology, energy, mining, agriculture, meteorology, and forensics (Sarıca, 2021). Furthermore, AI investment in numerous sectors is rapidly increasing.

In higher education, if appropriately and effectively utilized, AI tools can provide significant benefits (Taşçı & Çelebi, 2020). However, there is still a debate on how to effectively integrate AI applications into education. It is anticipated that with the introduction of AI in education, investment and research in this area will grow significantly in the coming years (İşler & Kılıç, 2021).

## 1.2 ChatGPT

Technology is advancing rapidly, bringing new concepts and innovations to various aspects of our lives. Among these concepts, AI technology has become increasingly prevalent in recent years, thanks to extensive research and development efforts. One of the most talked-about AI applications is the AI chatbot, which employs deep-learning algorithms trained on vast amounts of data to generate human-like responses to user queries (Gilson et al., 2022).

In November 2022, the public release of ChatGPT (Chat Generative Pre-trained Transformer) provided a remarkable example of human-computer communication, thanks to its advanced technology (Cotton, Cotton & Shipway, 2023; de Winter, 2023; Topsakal & Topsakal, 2022; Wenzlaff & Spaeth, 2022; Zhai, 2022). This particular AI technology, which has been at the forefront of the technological agenda, is a natural language processing model with 175 billion parameters, developed by OpenAI (Gilson et al., 2022). ChatGPT is one of the most potent NLP systems, boasting an enormous number of parameters, making it one of the largest language models currently available (Cotton, Cotton & Shipway, 2023).

Founded in 2015 by prominent technology leaders, OpenAI is a research institute that focuses on the development of AI technologies. The institute is recognized for its research endeavors in diverse fields and possesses the capability to provide conversational responses, reject inappropriate queries, challenge erroneous assumptions, admit mistakes and learn from its own errors with the aid of ChatGPT, an optimized language model (Jiao et al., 2023). This technology, which incorporates natural language processing and deep learning, utilizes a vast dataset and generates text that resembles human language (Qadir, 2022). Although initially criticized for its factual accuracy, the technology has gained popularity due to its ability to provide detailed and comprehensible answers to inquiries. Employing the GPT-3 text interpreter, this AI code is a form of NLP and can read and produce written texts (Pavlik, 2023).

According to Güçlütürk (2022), the output produced by ChatGPT is influenced by the preceding content, and even if the same command is entered, the output may differ. Consequently, the content produced in ChatGPT is highly personalized and original, and it relies on the input provided by the users and the given content (Wenzlaff & Spaeth, 2022). ChatGPT is designed to communicate and interact with people in a manner similar to human-to-human interaction, and it can respond in various languages (O'Connor & ChatGPT, 2023; Wenzlaff & Spaeth, 2022). While ChatGPT was initially intended for online customer service, it is now utilized for diverse applications and purposes, such as healthcare, software development, content creation, language translation, increasing business efficiency, cost reduction, and customer service through the utilization of AI (Gilson et al., 2022; Qadir, 2022). It is plausible to assume that AI and chatbots will continue to evolve with the advancement of technology.

It can be asserted that ChatGPT technology, designed to converse with users and provide meaningful responses to their inquiries (de Winter, 2023), will continue to evolve and become more impressive in the future (Qadir, 2022). The primary attribute of ChatGPT, which has garnered the admiration of millions of users in a brief timeframe, is its ability to supply accurate responses to user queries in real-time (Qadir, 2022). Another impressive capability is its capacity to generate high-quality, error-free text that is difficult to differentiate from human composition (Susnjak, 2022).

Similar to other productive AI systems, ChatGPT delivers answers to inquiries rapidly while preserving semantic coherence. Nonetheless, it is important to note that the answers are not always correct or appropriate, and they frequently contain erroneous and biased references (de Winter, 2023; O'Connor & ChatGPT, 2023; Qadir, 2022). Moreover, ChatGPT has limited information on events that occurred post-2021. However, as bots such as ChatGPT, believed to increase in competency over time, continue to refine themselves, such problems are expected to diminish.

### 1.3 Purpose of the Study

According to the literature, ChatGPT has been shown to offer extensive knowledge on almost any subject, and to provide reliable and accurate responses to challenging inquiries that necessitate advanced information analysis, synthesis, and application (Susnjak, 2022). Furthermore, Qadir (2022) suggests that ChatGPT has the potential to offer effective learning opportunities by creating realistic virtual simulations for applied learning. Additionally, AI chatbots such as ChatGPT can offer prompt feedback and corrections to questions, thereby assisting students in comprehending intricate concepts that are difficult to learn (Cotton, Cotton & Shipway, 2023). By using AI chatbots, students can benefit from personalized, interactive learning experiences, and better manage their time by planning their assignments and homework (Cotton, Cotton & Shipway, 2023; Huh, 2023; O'Connor & ChatGPT, 2023).

Despite the potential benefits of ChatGPT as an educational tool, the full extent of its impact on education remains uncertain and requires further investigation (de Winter, 2023; Qadir, 2022; Zhai, 2022). It is crucial to consider both the potential advantages and risks associated with emerging technologies like ChatGPT in order to anticipate and prepare for the future of education. One significant concern is the possibility of students using ChatGPT to cheat, particularly on online exams, due to its ability to generate personalized and authentic responses (de Winter, 2023; Stokel-Walker, 2022; Susnjak, 2022). As online education becomes increasingly widespread, ensuring the validity and reliability of online exams is a critical issue that must be addressed. It is important to acknowledge that further research is required to develop effective strategies that mitigate potential risks and leverage the benefits of AI. Furthermore, there is a scarcity of literature that delves into the potential educational use of ChatGPT, a novel tool in this domain. Hence, investigating the capabilities of this AI agent is anticipated to augment the current body of knowledge. This study aims to evaluate the performance of the newly launched ChatGPT on anatomy course examination questions among students enrolled in the Faculty of Health Sciences in Turkey.

## 2. METHOD

This study aimed to compare the performance of ChatGPT with that of health sciences faculty students in answering anatomy course questions. A descriptive study design was employed, and 37 students from a state university in Turkey participated in the study. The students received four weeks of training on a specific anatomy topic and then took a multiple-choice test comprising 40 questions on the covered material. The same test was also given to ChatGPT, and the answers generated were compared with those of the students. The data were analyzed using descriptive statistics, including the number, percentage, and mean.

## 3. RESULTS

In the study, a multiple-choice test was applied to students taking the anatomy course with ChatGPT. Below are some sample questions and screenshots of the answers from ChatGPT.

**YU** 1- Which of the papillae listed below is not responsible for receiving taste sensations?  
Papillae Vallatae  
Papillae Foliatae  
Papillae Conicae  
Papillae Lentiformes  
Papillae Fungiformes


 Papillae Conicae is not responsible for receiving taste sensations.

Figure 1. Screenshot of ChatGPT's correct answer to a sample question in the anatomy course

**YU** "I- Gl. Parotidea  
II- Gl. Sublingualis  
III- Gl. Submandibularis  
IV- GII. Buccalis  
V- GII. Labiales

Which of the following salivary glands' secretions open into the vestibule of the mouth?

II and III  
I, II, and III  
I, II, and IV  
I, IV, and V  
II, IV, and V"


 The secretions of salivary glands II and III open into the vestibule of the mouth. Therefore, the answer is: "II and III".

Figure 2. Screenshot of ChatGPT's wrong answer to a sample question in the anatomy course

The statistical values of the scores obtained from the tests applied in the study are shown in Table 1.

Table 1. Statistical values of the answers of the students and ChatGPT regarding the exam in the anatomy course

	Correct Responses	Incorrect Responses	Grade (0-100)
ChatGPT	27	13	67.5
Students	21.1	18.9	52.8

The results presented in Table 1 indicate that ChatGPT correctly answered 27 out of 40 questions, corresponding to a success rate of 67.5%. By comparison, the students' average score was

21.1 out of 40, equivalent to a success rate of 52.8%. These findings suggest that ChatGPT's performance was better than the average performance of the students enrolled in the anatomy course at the Faculty of Health Sciences. Based on these results, it can be concluded that ChatGPT demonstrated a successful performance on this exam. Figure 3 shows an example of how ChatGPT compared to all students on the exam.

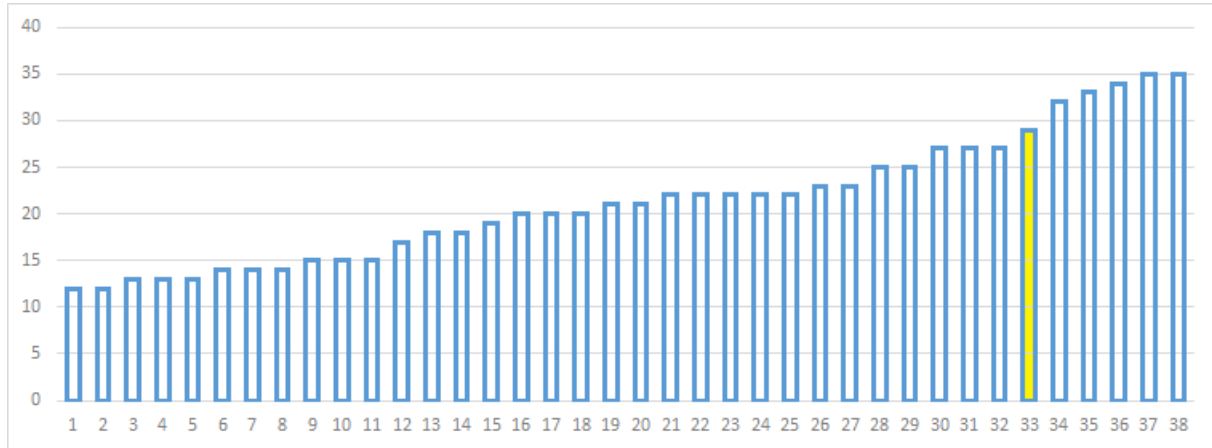


Figure 3. Distribution of anatomy test results. ChatGPT performance is highlighted in yellow

Current results show that ChatGPT performs similarly to the average student in understanding anatomy.

#### 4. DISCUSSION

The objective of this study was to compare the ability of an AI agent, ChatGPT, to answer questions on anatomy course exams with the performance of students from the Faculty of Health Sciences in Turkey. Results showed that ChatGPT had a significantly higher ratio of correct answers compared to students. Furthermore, ChatGPT's knowledge in answering the anatomy exam was found to be superior to that of the students. Other studies also support the potential of ChatGPT as a new educational tool. For instance, Gilson et al. (2022) found that ChatGPT performed comparably to medical students on medical licensing exams, while de Winter (2022) reported that ChatGPT's performance on English comprehension was similar to that of university students. Conversely, Huh (2023) found that ChatGPT's overall performance on a parasitology exam was lower than that of medical students in Korea. Similarly, Geerling et al. (2023) found that ChatGPT achieved high performance in a university exam measuring economics knowledge.

In the future, ChatGPT is expected to improve its performance through deep learning, which is a promising development that educators and researchers should be aware of in terms of its potential applications in teaching and learning. However, there have been varying results in other studies related to ChatGPT. In a study conducted by Nisar and Aslam (2023), they requested ChatGPT to answer questions in the field of pharmacology. Although the answers provided by ChatGPT were generally accurate, the source or reference of the answers was not provided. Moreover, the study found that ChatGPT could potentially serve as a self-study tool for students struggling with pharmacology. Nonetheless, it is worth noting that the ability of ChatGPT to generate highly realistic texts poses a potential risk to the integrity of online exams, and precautions should be taken to prevent this from happening (de Winter, 2022; Susnjak, 2022).

In the current state, ChatGPT is capable of producing accurate responses within seconds. However, it has limitations in interpreting visual aids such as diagrams, shapes, and tables, which can

be easily comprehended by human students. Thus, these visual aids need to be explained in text form for ChatGPT to understand. Additionally, if a question is ambiguous or incomprehensible, ChatGPT may produce an incorrect response. To mitigate this issue, it is advisable to rephrase the question in a clear and precise manner.

The recent publication of ChatGPT implies that there are limited studies in the literature that compare its performance with that of students. The current study utilized a multiple-choice test to assess performance. Future studies may explore the performance of ChatGPT and other AI language models in various exam formats. Furthermore, this study focused on a university-level anatomy course. To make meaningful comparisons, other researchers can conduct similar studies at different educational levels and in different courses, enabling a more in-depth investigation of their efficacy.

## REFERENCES

- Akyürek, H. A. (2013). *Intelligent workforce management by using artificial intelligence techniques*. Master Thesis, Türkiye.
- Arslan, K. (2017). Eğitimde yapay zeka ve uygulamaları. *Batı Anadolu Eğitim Bilimleri Dergisi*, 11(1), 71-88.
- Büyükgoze, S., & Dereli, E. (2019). Güncel sağlık bilimleri çalışmaları II, Dijital Sağlık ve Yapay Zeka, Akademisyen Kitabevi A.Ş., Editör: Tuncay ÖZGÜNEN, ISBN: 978-605-258-626-6i.
- Cotton, D. R., Cotton, P. A., & Shipway, J. R. (2023). Chatting and cheating. Ensuring academic integrity in the era of ChatGPT. Preprint. <https://doi.org/10.35542/osf.io/mrz8h>
- de Winter, J. C. F. (2023). Can ChatGPT pass high school exams on English language comprehension?. Researchgate. Preprint.
- Geerling, W., Mateer, G. D., Wooten, J., & Damodaran, N. (2023). Is ChatGPT smarter than a student in principles of economics?. Available at SSRN 4356034.
- Gilson, A., Safranek, C., Huang, T., Socrates, V., Chi, L., Taylor, R. A., & Chartash, D. (2022). How does ChatGPT perform on the medical licensing exams? The implications of large language models for medical education and knowledge assessment. medRxiv. Preprint. <https://doi.org/10.1101/2022.12.23.22283901>
- Güçlütürk, O. G. (2022). ChatGPT ile üretilen içeriklerin eser niteliğinin 5846 sayılı fikir ve sanat eserleri kanunu bakımından değerlendirilmesi. *Galatasaray Üniversitesi Hukuk Fakültesi Dergisi*, 2, 1899-1918.
- Huh, S. (2023). Are ChatGPT's knowledge and interpretation ability comparable to those of medical students in Korea for taking a parasitology examination?: A descriptive study. *Journal of Educational Evaluation for Health Professions*, 20(1), 1-13. <https://doi.org/10.3352/jeehp.2023.20.1>
- İşler, B., & Kılıç, M. (2021). Eğitimde yapay zekâ kullanımı ve gelişimi. *e-Journal of New Media*, 5(1), 1-11.
- Jiao, W., Wang, W., Huang, J., Wang X, & Zhaopeng, T. (2023). Is chatGPT a good translator? A preliminary study. arXiv. Preprint.
- Korkmaz, A., & Büyükgoze, S. (2019). Sahte web sitelerinin sınıflandırma algoritmaları ile tespit edilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, 16, 826-833. <https://doi.org/10.31590/ejosat.598036>
- Nisar, S., & Aslam, M. S. (2023). Is ChatGPT a good tool for T&CM students in studying pharmacology?. Available at SSRN 4324310.

- O'Connor, S., & ChatGPT. (2023). Open artificial intelligence platforms in nursing education: Tools for academic progress or abuse? *Nurse Education in Practice*, 66, 103537. <https://doi.org/10.1016/j.nepr.2022.103537>
- Pavlik, J. V. (2023). Collaborating with chatGPT: Considering the implications of generative artificial intelligence for journalism and media education. *Journalism & Mass Communication Educator*, 1-10. <https://doi.org/10.1177/10776958221149577>
- Qadir, J. (2022). Engineering education in the era of ChatGPT: Promise and pitfalls of generative AI for education. *TechRxiv*. Preprint. DOI: 10.36227/techrxiv.21789434.v1
- Sarica, R. (2021). *Eğitimde yapay zeka: Kavramsal temeller*. Köse, U. (Ed.) Eğitimde zeki ve esnek teknolojiler (pp.121-139), Ankara: Pegem Akademi.
- Shi, Z.-Z., & Zheng, N.-N. (2006). Progress and challenge of artificial intelligence. *Journal of Computer Science and Technology*, 21(5), 810-822. <https://doi.org/10.1007/s11390-006-0810-5>
- Stokel-Walker, C. (2022). AI bot ChatGPT writes smart essays — should professors worry? <https://doi.org/10.1038/d41586-022-04397-7>
- Susnjak, T. (2022). Chatgpt: The end of online exam integrity? arXiv. Preprint. <https://doi.org/10.48550/arXiv.2212.09292>
- Talan, T. (2021). Artificial intelligence in education: A bibliometric study. *International Journal of Research in Education and Science (IJRES)*, 7(3), 822-837. <https://doi.org/10.46328/ijres.2409>
- Taşçı, G., & Çelebi, M. (2020). Eğitimde yeni bir paradigma: “Yükseköğretimde yapay zekâ”. *OPUS International Journal of Society Researches*, 16(29), 2346-2370. <https://doi.org/10.26466/opus.747634>
- Tektaş, M., Akbaş, A., & Topuz, V. (2002). Yapay zekâ tekniklerinin trafik kontrolünde kullanılması üzerinde bir inceleme. *Uluslararası Trafik ve Yol Güvenliği Kongresi ve Fuarı*.
- Topsakal, O., & Topsakal, E. (2022). Framework for a foreign language teaching software for children utilizing AR, voicebots and ChatGPT (large language models). *The Journal of Cognitive Systems*, 7(2), 33-38. <https://doi.org/10.52876/jcs.1227392>
- Wei, X. (2018). The application and development of artificial intelligence in smart clothing. *IOP Conference Series: Materials Science and Engineering*, 320, 012017. <https://doi.org/10.1088/1757-899X/320/1/012017>
- Wenzlaff, K., & Spaeth, S. (2022). *Smarter than humans? Validating how OpenAI's chatGPT model explains crowdfunding, alternative finance and community finance*. University of Hamburg – Digital Markets – Working Paper Series. Available on SSRN. Electronic copy available at: <https://ssrn.com/abstract=4302443>

# Yazılım Dağıtım Sürecinin Otomatikleştirilmesine İlişkin Uygulamalı Bir Çalışma

\*\*\*

## A Practical Study on Automating the Software Deployment Process

Ender ŞAHİNASLAN<sup>1</sup> 

Nusret ARPACIOĞLU<sup>2</sup> 

Önder ŞAHİNASLAN<sup>3</sup> 

DOI:10.33461/uybisbbd.1206484

### Öz

#### Makale Bilgileri

##### Makale Türü:

Araştırma Makalesi

##### Geliş Tarihi:

24.11.2022

##### Kabul Tarihi:

30.04.2023

©2023 UYBISBBD  
Tüm hakları saklıdır.



Yazılım dağıtım, geliştirilen bir uygulamanın çalıştırılacak dijital ortamlara yüklenmesidir ve yazılım geliştirme yaşam döngüsünde önemli bir aşamadır. Herhangi bir hataya veya kesintiye yol açmayacak şekilde titizlikle uygulanmalıdır. Küçük organizasyon ve sistemlerde bu sürecin işleyişinin manuel olarak işletilebilmesi mümkündür. Ancak orta ve büyük organizasyonlarda dağıtım yapılacak ortamların çeşitliliği, büyüklüğü, karmaşıklığı buna engeldir. Ayrıca iş sürekliliği, güvenlik ve uyum gibi gereksinimler dikkate alındığında manuel dağıtımdan kaynaklı birçok risk söz konusudur. Bu tür risk ve sorunların aşımında otomatik yazılım dağıtım araçlarına ihtiyaç vardır. Bu çalışmada yazılım yaşam ve dağıtım süreci, manuel dağıtımdan kaynaklı sorunlar, sorunların çözümünde kullanılan otomatik dağıtım araçları incelenmiştir. Yazılım dağıtım sürecinin otomatikleştirilmesi uygulamalı olarak çalışılmıştır. Bu çalışma daha önce yapılan çalışmalardan farklı olarak, konusu ve uygulamalı bir sunum olması sebebiyle ilgili kurum çalışanları, öğrenci ve araştırmacılar için teknik bir rehber niteliğindedir. Yazılım dağıtım süreci ve bu sürecin otomatikleştirilmesine yönelik uygulamalı çalışma literatüre kazandırılmıştır.

**Anahtar Kelimeler:** Teknoloji ve Yenilik, Bilişim, Otomatik Yazılım Dağıtım, Süreç Yönetimi.

### Abstract

#### Article Info

##### Paper Type:

Research Paper

##### Received:

24.11.2022

##### Accepted:

30.04.2023

©2023 UYBISBBD  
All rights reserved.



Software deployment is the installation of a developed application on digital media to be run and it is an important stage in the software development life cycle. It should be applied meticulously so as not to cause any errors or interruptions. In small organizations and systems, it is possible to operate this process manually. However, the diversity, size, and complexity of the environments to be distributed in medium and large organizations prevents this. In addition, there are many risks arising from manual deployment when considering requirements such as business continuity, security and compliance. Automatic software deployment tools are needed to overcome such risks and problems. In this study, software lifetime and deployment process, problems arising from manual deployment and automatic deployment tools used to solve problems are examined. Automation of the software deployment process has been practiced. This study, unlike previous studies, is a technical guide for the relevant institution staff, students and researchers due to its subject and practical presentation. The software deployment process and practical study on automating this process have been brought to the literature.

**Keywords:** Technology and Innovation, Informatics, Automatic Software Deployment, Process Management.

**Atıf / to Cite (APA):** Şahinaslan, E., Arpacioğlu, N., & Şahinaslan Ö. (2023). Yazılım Dağıtım Sürecinin Otomatikleştirilmesine İlişkin Uygulamalı Bir Çalışma. Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi, 7(1), 41-67.

<sup>1</sup> Dr. Öğr. Üyesi, Mudanya Üniversitesi, Bilgisayar Mühendisliği Bölümü, [dr.endsa@gmail.com](mailto:dr.endsa@gmail.com)

<sup>2</sup> Yüksek Lisans, Maltepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, [nusretar@gmail.com](mailto:nusretar@gmail.com)

<sup>3</sup> Dr. Öğr. Üyesi Maltepe Üniversitesi, Bilişim Bölümü, [ondersahinaslan@maltepe.edu.tr](mailto:ondersahinaslan@maltepe.edu.tr)

## 1. GİRİŞ

Yazılım ürünleri bir ihtiyacın karşılanması için tasarlanır, kodlanır, uygulama/ürün haline getirilerek servis edilir. Dijitalleşen bir dünyada yazılımlara duyulan ihtiyaç sürekli artmaktadır. Bu ihtiyaç bazen yeni bir uygulama bazen de var olan bir uygulamada yapılan değişiklikler yoluyla giderilmektedir. Yeni geliştirilen yazılım ürününün ya da mevcut bir yazılımın küçük bir parçasının değişimi sonucunda oluşan uygulamanın dağıtımını sürekliliği olan bir süreçtir (Highsmith & Cockburn, 2001). Bir uygulamanın dağıtımında konuşlandırılacak altyapı, sistem ve uygulama ortamlarının özel yapılandırma ihtiyaçları dikkate alınır. Dağıtımın eksiksiz, tam, güvenli ve çalışabilir şekilde uygulanması gerekir. Yazılım geliştirme yaşam döngüsünün son aşamalarından biri olan dağıtım ('*deployment*') kısmı yakın bir geçmişe kadar genellikle işinde uzman kişiler tarafından manuel olarak gerçekleştirilmekteydi. Ancak insana dayalı manuel olarak gerçekleştirilen bir dağıtım işlemi insanın dikkat ve becerisine de bağlı olarak birçok hata ve riske açıktır. Bu tür sorunlar dikkate alındığında manuel yerine otomatik dağıtım süreçlerine geçiş kaçınılmaz olmuştur. Özellikle bankacılık, sigortacılık ve savunma gibi çok farklı sistem ve mimarilerdeki uygulamaların çalıştığı, hata ve gecikmelere tahammül edilemeyen sektörlerde yazılım dağıtım işlerinin manuel olarak yürütülebilmesi imkânsız hale gelmiştir. Hızlı ve hatasız gerçekleştirme arzusu, teknolojik ilerlemeler, BT standart gereksinimleri, güvenlik ve yasal uyum gibi ihtiyaçlar neticesinde yazılım dağıtımının otomatik kullanımı yaygınlaşmaya başlamıştır. Dağıtım süreci ve sürecin otomatikleştirilmesinde kullanılan araçlarla ilgili bilgi ve tecrübe yetersizlikleri hatalı veya eksik yapılandırmalara sebep olabilmektedir. Bu tür sorunların varlığı dağıtım sürecinin başarılı bir şekilde yürütülebilmesi önünde birer engeldir.

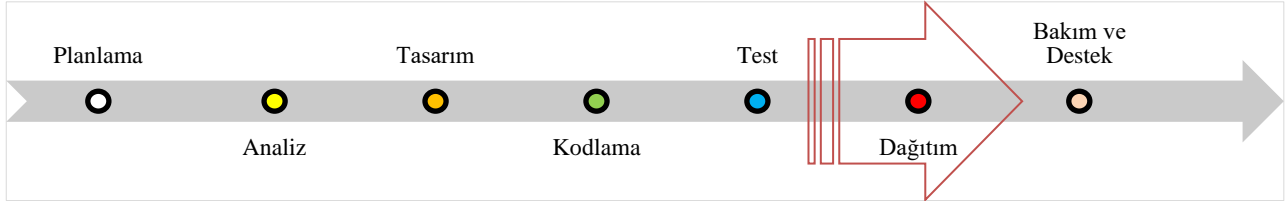
Yazılım dağıtım süreci, geliştirilen bir uygulamanın ilgili sistemlere dağıtımını, kullanıma hazır hale getirilmesini diğer bir deyişle uygulamanın çalışır durumda kalmasını içeren üretim sonrası karmaşık bir süreçtir (Arcangeli vd., 2015). Her bir yazılım ürününün farklı bir yapılandırma ve dağıtım yöntemine ihtiyaç duyması, bir dağıtımın farklı zamanlarda benzer şekilde gerçekleştirilme talebi ek insan kaynağı ihtiyacına dolayısıyla iş yükünün artmasına sebep olmaktadır. Aynı zamanda operasyonel işlemlerden kaynaklı hata ve risklere açık ve maliyetli bir yazılım dağıtım sürecinin manuel bir şekilde yönetilebilmesi oldukça güçtür. Oysa otomatik dağıtım araçlarının kullanılması durumunda; başlangıçta dikkatlice tasarlanan bir otomatik dağıtım süreci ile insan müdahalesi olmadan, neredeyse sıfır hata ile istenildiği kadar dağıtım gerçekleştirilebilir. Otomatik dağıtım süreci, dağıtım yapılacak uygulama ve ortamları bilen teknik uzmanlar tarafından bir defaya mahsus modellendikten sonra manuel herhangi bir müdahale olmadan istenildiği kadar kullanılabilir.

Yazılım yaşam döngüsü, bir değişiklik yapma kararından üretim safhasına kadar geçen süre olarak bilinir (Farley & Humble, 2010). Bu süre birçok kurumda yazılım geliştirme süreci haftalar hatta aylarca sürerken, kritik bir hataya yönelik yeni bir özelliğin eklenmesi veya küçük bir düzeltme de çoğu zaman günlerce veya haftalarca sürebilmektedir (Poppendieck ve Poppendieck, 2007). Yazılım geliştirme sürecindeki küçük iyileştirme, yazılım kalitesinde önemli bir iyileşmeyle sonuçlanabilir (Pai, 2002). Ancak küçük bir değişikliğin bile uzun sürmesi bir şeylerin yanlış olduğunun açık bir göstergesidir (Morris, 2016). Özellikle orta ve büyük ölçekli kurumlarda yazılım dağıtım sorunlarının aşılmasına yönelik çalışmalar yürütülmektedir. Sunulan çözümlerden birisi yazılım dağıtımında otomatik araç kullanımınıdır. Yazılım dağıtımında otomatik araç kullanımı sürecin karmaşıklığını azaltırken yazılım uyarlamasında geçen süreyi de en aza indirir (Ruiz-Rube vd., 2015). Karmaşık yazılımlarla bile, kontrollü bir otomatik süreç olarak oluşturma, devreye alma, test etme ve dağıtım işlemleri otomatikleştirilebilir, döngü süresi saatlere hatta dakikalara düşürülebilir (Eloranta, 2018). Bir yazılım uygulamasının yeni sürümünde yapılan tüm değişiklikler, eklenen özellikler, iyileştirmeler, hata düzeltmeleri ve kullanımdan kaldırılan özelliklere dair tüm bilgiler sürüm notları içerisinde işlenir. Sürüm notları ise yazılım geliştirmede hayati bir rol oynamasına rağmen genellikle el yordamıyla oluşturur (Ali vd., 2020). Sürekli tekrarlanan bir yazılım dağıtım sürecinin otomatikleştirilmesi yazılım dağıtım sürecinin daha hızlı, doğru ve çevik bir biçimde yürütülmesine katkı sunar.



## 1.1. Yazılım Geliştirme Süreci

Yazılım geliştirme süreci, kullanıcı ihtiyaçlarını karşılamak için kabul edilebilir bir sürede kullanışlı çıktılar üretecek şekilde belli yöntem, adım ve işlemler kullanarak bir standart oluşturma işidir (Şahinaslan, 1998). Bu süreç diğer ürün geliştirme süreçlerine göre daha uzun zaman alan, karmaşık ve zor bir süreçtir (Borandag & Yücalar, 2020). Genel kabul gören bir yazılım yaşam döngüsü, talep ve ihtiyaçlara göre kalite güvence gereksinimleri, insan, ortam, süre, maliyet, risk gibi unsurların göz önüne alındığı fizibilite ve planlama çalışmaları ile başlar. Analiz, tasarım, kodlama, test, dağıtım, bakım ve destek süreçleriyle devam eder. Şekil-1’de yazılım geliştirme süreci temel aşamaları gösterilmektedir.



Şekil 1. Yazılım geliştirme süreci aşamaları

Bu temel aşamalar tek yönlü bir akış gibi gösterilmekle birlikte uygulamada her bir adımdan önceki bir adıma geri dönüş de mümkündür. Ancak asıl hedeflenen her bir adımı mümkün olduğunca eksiksiz tamamlayarak sonraki aşamaya geçmek, süreci başarıyla tamamlamaktır.

## 1.2. Dağıtım Süreci

Çalışmanın ana konusu olan dağıtım süreci, yeni geliştirilen veya güncellenen bir yazılımı kullanıma hazır hale getirme, kullanıcıya sunma evresidir. Yazılım dağıtımını genelde kolay bir süreç gibi görünmekle birlikte titiz ve detaylı bir çalışma gerektiren bir iştir. Test sürecinin başarıyla geçilmesine rağmen dağıtım sonrası istenmeyen sorunlarla karşılaşılması da muhtemeldir. Özellikle test ortamı ile çalışma ortamındaki kullanıcı ve rol yapısı farklılığı, yetkilendirme, güvenlik, iş sürekliliği, sistem kapasite ve performans farklılıklarından kaynaklı birçok sorunla karşılaşılabilir. Bu süreç kodlama ve test aşamasını geçen bir uygulamanın kullanılacağı ortamlara dağıtımını ve başarıyla yürütülmesinde geçen aşamadır. Bu süreç kapsamın belirlenmesi, planlama, tasarım, dağıtım/yüklenme/yaygınlaştırma, çalıştırma, izleme ve ihtiyaç kalmayan kod parçalarının ortamdaki kaldırılması, kullanıcı hak ve yetkilerinin düzenlenmesi gibi süreç sonunda yürütülen faaliyetleri içerir. Dağıtım süreci aşamaları Şekil 2’de gösterilmektedir.



Şekil 2. Uygulama dağıtım süreçleri

Uygulama, veri tabanı, işletim sistemi, kullanıcı sayısı, etkilenecek sistemler dikkate alınarak *kapsam* belirlenir. Dağıtımda yapılacak işlerin listesi, uygulanmasında karşılaşılabilecek bir başarısızlık veya acil bir durumda yapılacaklar ve alınacak önlemler, yedekleme ve *dağıtım planı* oluşturulur. Dağıtım süreci ve öncesinde yapılacak iş, işlem, ihtiyaç ve gereksinimleri dikkate alınarak *tasarım* gerçekleştirilir. Uygulama yardımcı araçlar kullanarak belirlenen sunucu ve sistemlere *yüklenir*. Yükleme öncesinde ilgili sunucular üzerinde ihtiyaç duyulan uygulama ve sistemler erişim yetkileri de verilerek çalışmaya hazır hale getirilir. Yüklenen uygulamalar kontrollü bir şekilde *çalıştırılır*, test ve ön kontrol sonuçlarına göre ilgili sunucularda ihtiyaç duyulmayan kod parçaları silinir, erişim yetkileri gözden geçirilerek uygulama yaygın kullanıma açılır. Canlı (PROD) ortam üzerinde çalışmaya alınan uygulamalar izleme araçları yardımıyla *izlemeye* alınır. Uygulamalar, yeni bir talep ya da güncelleme gelmediği sürece çalışır.

### 1.3. Dağıtım Araçları

Geliştirilmesi tamamlanmış olan bir uygulamanın çalışacağı ortama manuel yöntemlerle yapılan dağıtım ve yaygınlaştırmada yaşanan problemlere karşı dağıtım sürecinin otomatik dağıtım araçları yardımıyla gerçekleştirilmesi bu sürece pozitif katkı sunar. Dağıtım sürecinin otomatikleştirilmesine yardımcı birçok uygulama yazılım aracı vardır. Yaygın kullanılan uygulamalara ait liste Tablo-1’de sunulmaktadır.

**Tablo 1:** Yazılım Dağıtımının Otomatikleştirmesinde Kullanılan Uygulama Araçları

Uygulama	Açıklama	Kaynak
AWS CodeDeploy	Amazon EC2 bulut sunucuları ile kurumsal ağ ya da herhangi bir bulut sunucusuna yazılım dağıtımını otomatikleştiren bir hizmet aracıdır.	(AWS Code Deploy, 2022)
Azure DevOps	Kullanıcıların sürüm tanımlama, otomasyon çalıştırma, sürüm izleme imkânı sunan, farklı platformlara aynı anda dağıtım yapabilen, Windows, Linux veya Mac işletim sistemlerinde derlenebilir açık kaynak kodlu bir uygulama aracıdır.	(DevOps, 2022)
Bamboo	Farklı ölçekteki yapılar da koddan devreye almaya kadar otomatikleştirilmiş iş akışlarıyla güvenli ve ölçeklenebilir bir çözüm sunan dağıtım uygulamasıdır.	(Bamboo, 2022)
Capistrano	Ruby’de yazılmış, farklı dilleri destekleyen açık kaynak kodlu bir uzak sunucu otomasyon dağıtım aracıdır.	(Capistrano, 2022)
CircleCI	Esnekliğine, güvenilirliğine ve hızına önem veren bir CI çözümüdür. GitHub, GitLab SaaS veya Bitbucket kullanarak derlemeleri birçok ortamda otomatikleştirme yeteneklerine sahiptir. Farklı dil ve uygulamaları destekler.	(CircleCI, 2022)
Codar	HP’nin sürekli dağıtım çözümüdür. Bir paketi dağıtıldıktan sonra aynı şablonu kullanarak paketin daha yeni bir sürümlerinde kullanmayı hedeflemektedir.	(Codar, 2022)
CloudBees	Kurumsal boyuttaki gereksinimler için tasarlanmış esnek, ölçeklenebilir ve yönetilen sürekli dağıtım çözümüdür.	(CloudBees, 2022).
DeployBot	Çoklu ortamlar için manuel veya otomatik dağıtımlara izin verir, aynı anda birden fazla sunucuya dağıtım yeteneğine sahiptir.	(DeployBot, 2022)
Digital.ai	Uygulama dağıtımını herhangi bir teknoloji ortamına göre otomatikleştiren, dağıtım hızı ve ölçeklenebilirliğini artıran, model tabanlı, aracısız bir dağıtım otomasyon çözümüdür.	(Digital, 2022)
GoCD	Uygulamanın sürümünü istenilen yere istenilen zamanda güvenli ve denetlenebilir bir yapıda dağıtım olanak sunan açık kaynaklı bir dağıtım çözümüdür.	(GoCD, 2022)
Gradle	Yazılımların daha hızlı oluşturmaya, otomatikleştirmeye ve sunmasına yardımcı bir araçtır. LinkedIn, Netflix, Adobe, android tarafından kullanılan bir yaygınlaştırma aracıdır.	(Gradle, 2022)
IBM UrbanCode	Sürekli dağıtım ve devreye alma sürecini otomasyonu sağlayan, izlenebilirlik ve denetleme yetenekleri olan bir uygulama dağıtım çözümüdür.	(IBM, 2022)
Jenkins	Yazılım oluşturma, test ve teslim etme veya dağıtım ile ilgili her türlü görevi otomatikleştirmede bağımsız kullanılabilen açık kaynaklı bir otomasyon aracıdır.	(Jenkins, 2022).
Octopus Deploy	Modern DevOps sürekli yazılım dağıtım, geri bildirim alan, değişiklik yapan ve yeniden dağıtım ekipleriyle çalışmak üzere tasarlanmıştır. .NET uygulamaları dağıtımını otomatikleştirmek amacıyla oluşturulmuştur.	(Octopus, 2022).
TeamCity	Esnek iş akışlarına, iş birliğine ve geliştirme uygulamalarına izin veren genel amaçlı bir CI/CD yazılım platformudur. Sürekli dağıtım olanak tanır.	(TeamCity, 2022)
Travis CI	Projenin bulut ve şirket içi test ve dağıtımına izin veren açık kaynak kodlu bir uygulamadır. PHP, Java, Ruby, Python gibi 30’dan fazla program dilini desteklemektedir.	(Travis, 2022)

## 2. LİTERATÜR TARAMASI

Yazılım dağıtımı ve otomatikleştirilmesine yönelik çalışmalar ele alındığında oldukça kısıtlı sayıda yayına ulaşılabilmektedir. Bu çalışmada olduğu gibi uygulamalı bir örneğe rastlanamamakla birlikte elde edilen konuyla ilgili yakın çalışmalar aşağıda incelenmiştir.

(Poppendieck ve Poppendieck, 2007) tarafından yapılan çalışmada, bir kurumda yalnızca tek bir kod satırında yapılan bir değişikliğin dağıtım süreleri incelenerek yapılan bu işlemin sonraki zamanlarda güvenilir bir şekilde tekrarlanıp tekrarlanamayacağı sorgulanmıştır.

(Arcangeli vd., 2015) dağıtık yazılım sistemlerinde otomatik dağıtıma ilişkin son teknolojilerin ele alındığı, 2000'lerden itibaren ise Linux Redhat, Microsoft Windows, Microsoft .Net, Oracle JavaBeans (2013), VMware (2008) sanallaştırma gibi farklı teknolojiler üzerinde yazılım dağıtım araçlarının kullanılmaya başladığı vurgulanmıştır. O günün koşullarında otomatik sürüm güncellemeleri, dağıtıma ait süreç yönetimleri ve teknolojik çözümleri ele almışlardır. Dağıtım otomatikleştirmede mevcut teknik çözümlerin istemci-sunucu mimarideki kurumsal ağlarda uygulama kurulumu veya kaldırma ile sınırlı olduğunu tespit etmişlerdir. 2014 yılına ait yazılım dağıtımının otomatikleştirilmesine yönelik bu sektörde Ansible, Chef, Puppet ve Octopus Deploy yapılandırma yönetim araçlarının yaygın kullanıldığını belirtmişlerdir.

(Ruiz-Rube vd., 2015) yazılım süreci dağıtımı ve değerlendirmesine yönelik çalışmada, yazılım sürecinin bir dizi teknik ve yöntemlerle sistematik yazılım üretimini teşvik ettiğini, model tasarımı, doğrulama, geçerli kılma, uygulamaya alma ve değerlendirme gibi faaliyetler içerdiğini konu etmektedir. Her yazılım projesinde uygulamanın işleme alınma süresini azaltma ve dağıtım işlemlerini mümkün olduğunca otomatikleştirmenin amaçlandığı, böylece mevcut karmaşıklığın azaltacağı sonucuna ulaşılmıştır. Bunu gerçekleştirmede iyi bilinen teknik ve uygulamaların kullanımının önemi vurgulanmıştır. Kurumların yazılım geliştirme ve bakım süreçlerinde kalitenin iyileştirilmesinde stratejilerine uygun yöntem, teknik ve uygulama araçlarına sahip olmaları gerekliliği belirtilerek yazılım süreçlerinin dağıtım ve değerlendirilmesi için bir çerçeve sunulmuştur. Otomasyona yönelik dağıtım ve değerlendirme faaliyetleri üzerine daha fazla araştırma yapılmasının gerekliliği vurgulanmıştır.

(Lascu vd., 2015) bileşen tabanlı uygulamaların otomatik devreye alınması adlı çalışmada, yazılım bileşenlerinin sayısı arttıkça özellikle karmaşık dağıtılmış eski sistemlerde bunların yönetilmesi ve otomatik devreye alınmasındaki zorluklara değinilmiştir. Günümüz bulut teknolojilerinde artık kritik bir zorluk haline geldiği, bulut tabanlı dağıtık sistemlerin heterojen yapıda çok sayıda bileşenin birleşiminden oluştuğu, bu tür karmaşık sistemlerde dağıtım sürecinin kolaylaştırılmasında otomasyon araç ve tekniklerin vazgeçilmez bir ihtiyaç olduğu öne sürülmüştür.

(Hofmann vd., 2019) robotik süreç otomasyonu (RPA) adlı çalışmada, dijital dönüşümün artmasıyla birlikte kurumsal yapılarda robotik süreç otomasyon popüler olduğunu ancak akademik araştırmaların yetersiz olduğunu, yazılım robotlarının insan tarafından gerçekleştirilen süreçleri otomatikleştireceğine vurgu yapmışlardır. Kullanımının kolay ve uyarlanabilir olması nedeniyle şirketlerin çevik projelerde yazılım robotlarını tasarlayarak uygulayabileceklerine, böylece şirket organizasyonuna, BT strateji ve yönetim yapılarına, yönetim sistemlerine doğrudan ve dolaylı etkilerini ele almışlardır. Sonuç olarak kuruluşların, süreç performansı, verimlilik, ölçeklenebilirlik, denetlenebilirlik, güvenlik, uygunluk ve uyumluluk hedeflere ulaşmada RPA uygulayabilecekleri sonucu yanında iş akışı otomasyonu gibi diğer otomasyon yaklaşımlarına karşı bu yaklaşımın avantaj ve dezavantajlarını da göz önünde bulundurmanın gerekliliğine dikkat çekmişlerdir.

(Ali vd., 2020) çalışmalarında sürüm notları ('*release notes*')'ın yazılım geliştirme sürecinde hayati bir role sahip olduğunu, yazılımın yeni bir sürümüne ait önemli bir işlem olmasına rağmen genellikle manuel olarak oluşturulduğunu, yazılımda yapılan lisans değişiklikleri, kullanımdan

kaldırılan kitaplıklar, yeni uygulama ara yüzü ve yazılımda yapılan değişikliklere ait açıklamalarının manuel olarak oluşturulmasının hatalara açık ve zaman alıcı bir işlem olduğunu vurgulamaktadır. Bu sorunun yazılım kaynak kod havuzundan otomatik olarak sürüm notları oluşturan çeşitli program açlarının varlığına dikkat çekmişlerdir. Araştırmacıların sürüm notlarının otomatikleştirilmesine yönelik çalışmalar yürüttükleri belirterek bunlara ait bir inceleme çalışması yürütmüşlerdir.

(Chen vd., 2020) çalışmalarında ağ işlevi sanallaştırma ortamlarında ağ hizmetlerinin otomatik dağıtım ve kontrolü üzerine çalışma yürütmüşlerdir. Bu teknolojinin hızla gelişimi ve uygulamaya alınmasıyla sanallaştırılan ortamlarda ağ hizmet sayısı ve türünde ciddi artışlar olduğunu, dağıtımın manuel yapılmasının hataya açık zaman alan bir süreç olduğunu, dağıtım ve kontrol sürecinin otomatikleştirilmesine yönelik yeni mekanizmalara acil ihtiyaç olduğunu vurgulamışlardır. Bu amaçla yazılım tabanlı ağ hizmeti dağıtım ve kontrolü için 'SDNSDC' adında yeni bir mekanizma önermişlerdir. SDNSDC'nin bir örneğini Linux konteyner teknolojisi üzerinde uygulaması sonucunda özelleştirilmiş ağ servislerinin verimli bir şekilde dağıtabildiği ve davranışlarını doğru bir şekilde kontrol edebildiğini ileri sürmüşlerdir.

(Mohd Nordin vd., 2021) yazılım geliştirme verimlilik modeli değerlendirmesinde uzman görüşlerinden yararlanmıştır. Çalışmada günümüz rekabet ortamında firmaların üretim maliyetlerini düşürürken kalitenin artırılması gerekliliği belirtilerek yazılım geliştirme giderlerini azaltmada etkili stratejinin verimliliği artırmak olduğu vurgulanmıştır. Bulut bilişim ve SaaS'ın yazılım geliştirme üretkenliğini artırmaya yardımcı olabileceği, literatür çalışması sonucunda üretkenliği etkileyen 19 faktör belirlendiği, bunların uzman görüşlerine göre değerlendirilmesi gerçekleştirilmiştir. Çalışmada, uzmanların 17 alt faktör ile üç ana faktör üzerinde hemfikir olduğu sonucuna ulaşılmıştır.

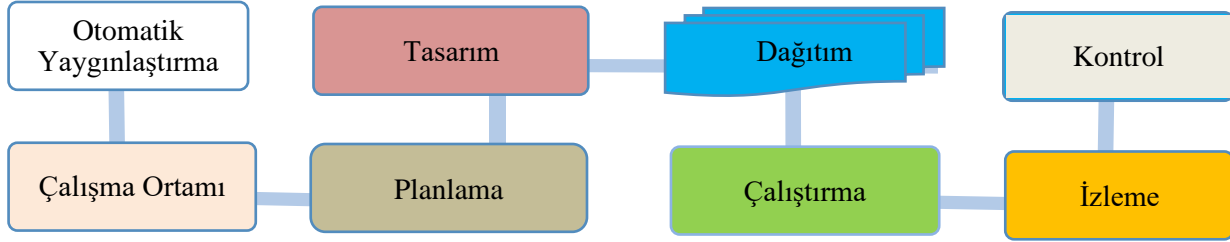
(Zai vd., 2021) çalışmalarında, son yıllarda bulut bilişimde yaşanan hızlı gelişmelere karşın bulut bilişime dayalı bilgisayar uygulamaları için otomatikleştirilmiş bir dağıtım sisteminin tasarımını tartışıldığını, dağıtım stratejisinde tahsis stratejisi, sistem esnekliğini artırmada uç sanal sunucuların kaynak kullanım performansını artıran yapılandırma ve uygulama hizmet kalitesini artıracaklarını vurgulamışlardır. Bileşen tabanlı programların dağıtım problemini biçimlendirmede dinamik ölçeklendirme algoritmalarına dayalı verimliliği optimize eden çalışma deney sonuçlarına göre önceki veri merkezlerine göre bulut veri merkezlerinin başarı oranı %87'nin üzerinde bulunmuştur. Araştırma sonucuna göre dağıtım sistem kullanıcılarının belirli uygulama gereksinimlerini karşıladığı, bazı mevcut sistemlere kurulum ve dağıtım yapılabildiği, bulut bilişim ortamının özelliğine göre daha iyi ölçeklenebilir sonuçlar elde edilebileceği sonucuna ulaşılmıştır.

(Zhang vd., 2021) çalışmalarında, endüstriyel internet platform mekanizmasında yaşanan çökme sorununun hayati öneme sahip olduğu, çok dilli yapısı, çeşitli çalışma modunun varlığı bunların dağıtım ve yönetiminde birçok zorluğa ve tekrarlı çalışmalara neden olduğu vurgulanmaktadır. Bu sorunun çözümünde OpenStack tabanlı, çok dilli bir endüstriyel mekanizma modelinde otomatik dağıtım şeması önerilmektedir. Dağıtım ve yönetimin çeşitliliğine göre geliştirmede talep ve otomatik dağıtım etkileşimine göre özel bir dağıtım ortamı oluşturmanın kolay ve geliştiricilerden tasarruf eden, aynı zamanda iş yükü karşısında çökmelere karşı endüstriyel eko sistem inşası ve endüstriyel gerçekleştirilmesine iyi bir destek olacağı sonucuna ulaşılmıştır.

DergiPark akademik sistemi (DergiPark, 2022) üzerinde yayınlanan akademik çalışmalar ile lisans ve lisansüstü tez çalışmalarını da içeren Piri Keşif aracı (PiriKesif, 2020) üzerinden 'yazılım dağıtımı', 'otomatik yazılım dağıtımı', 'otomatik dağıtım aracı' ve 'sürüm dağıtımı' anahtar alanları ile yapılan Türkçe literatür kaynakların varlığına dair yapılan incelemede herhangi bir çalışmaya ulaşılamamıştır.

### 3. MATERYAL VE METOT

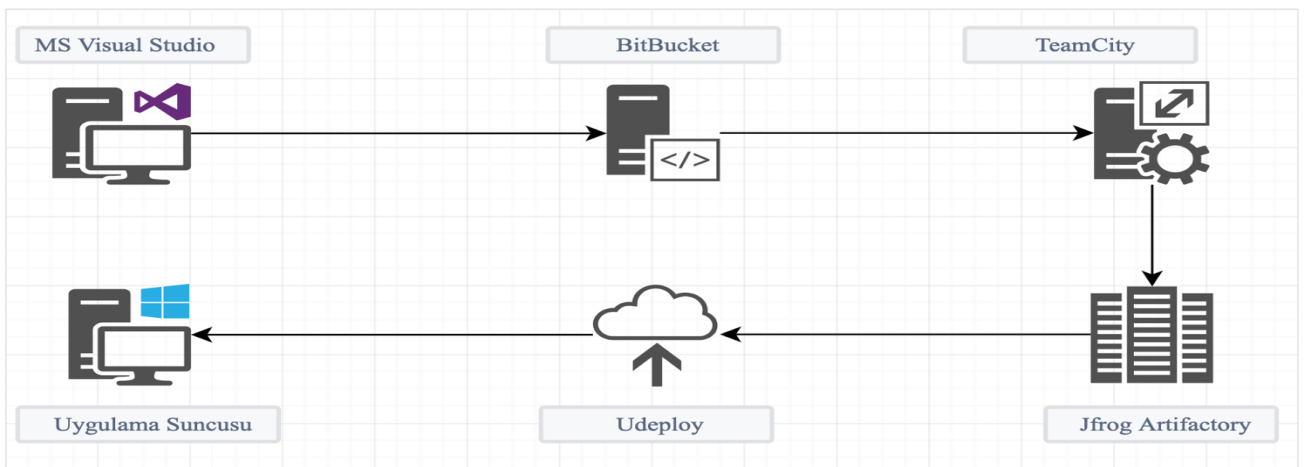
Bu çalışma, geliştirilen yazılımların dağıtımında manuel yaygınlaştırma yönteminin kullanımından kaynaklı sorunların çözümünde yararlanılan otomatik dağıtım yöntemini ele almaktadır. Otomatik yaygınlaştırmada kullanılan temel terim ve kavramlar açıklanarak süreç tasarlanmıştır. Uygulama aşamasında manuel dağıtım yapılan bir web yazılım paketinin otomatik dağıtım süreci çalışılmıştır. Otomatik dağıtım aracı olarak yaygın kullanılan IBM UrbanCode Deploy (UDeploy) uygulaması tercih edilmiştir. Uygulanmasında, UDeploy üzerinde yeni bir proje oluşturularak çalışma amacına uygun senaryo üzerinden planlama, tasarım, dağıtım, çalıştırma ve izleme aşamaları yürütülmüştür. Otomatik dağıtım süreci, kullanılan yöntem, tasarım ve uygulama örnekleriyle birlikte sunulmaktadır. Çalışma ana süreç aşamaları Şekil 3'te gösterilmektedir.



Şekil 3. Çalışma ana süreç aşamaları

#### 3.1. Çalışma Ortamı, Varsayım ve Sınırlılıklar

Bu çalışmada *IBM Urban Code Deploy* uygulama aracı kullanıldı. Uygulama sunucuları üzerinde *UDeploy* servisleri kurularak kullanıcı rol ve yetki tanımlandı. Uygulama kaynak kodları *Bitbucket* sunucuları üzerinde kullanılabilir şekilde hazırlandı. Derleme aşamasında kullanılan *TeamCity*, *.Net platformu*, *Microsoft SQL sunucu* veri tabanı uygulamalarının uygulama sunucusu üzerinde kurulu olduğu, dağıtım çıktılarının gönderileceği ve saklanacağı *Jfrog Enterprise Artifactory* platformunun çalışır şekilde hazır olduğu var sayıldı. *TeamCity* üzerinden derlenen paketin dağıtımını için *Jfrog* platformunda erişim yetkileri tanımlandı. Uygulamanın farklı uygulama ve *VT* sunucu ortamlarına ortak görevler üzerinden dağıtımını sırasında kullanılan dosya sunucusu ('file server'- *FS*) üzerinde lisanslı *UDeploy* servisi kurulmuştur. Ayrıca veri tabanı dağıtım sürecinde kullanılacak *SQL Server Data Tools (SSDT)* aracı *FS* üzerine kurulu şekilde hazırlandı. Uygulama, veri tabanı ve dosya sunucuları üzerinde *PowerShell* ve *.NET Framework*, *.NET Core*, *.NET 5* ve üzeri sürümlerin kurulu ve kullanılabilir olduğu varsayılmaktadır. Çalışma ortam ve ilişkileri Şekil 4'te sunulmaktadır.



Şekil 4. Çalışma ortamı

Kullanılan araçların sistemsel özellikleri; ProLiant DL380 Gen10 Sunucu, Windows Server 2016 64 bit işletim sistemi, Intel(R) Xeon(R) Gold 5118 işlemci, CPU @ 2.30 GHz, x64 tabanlı

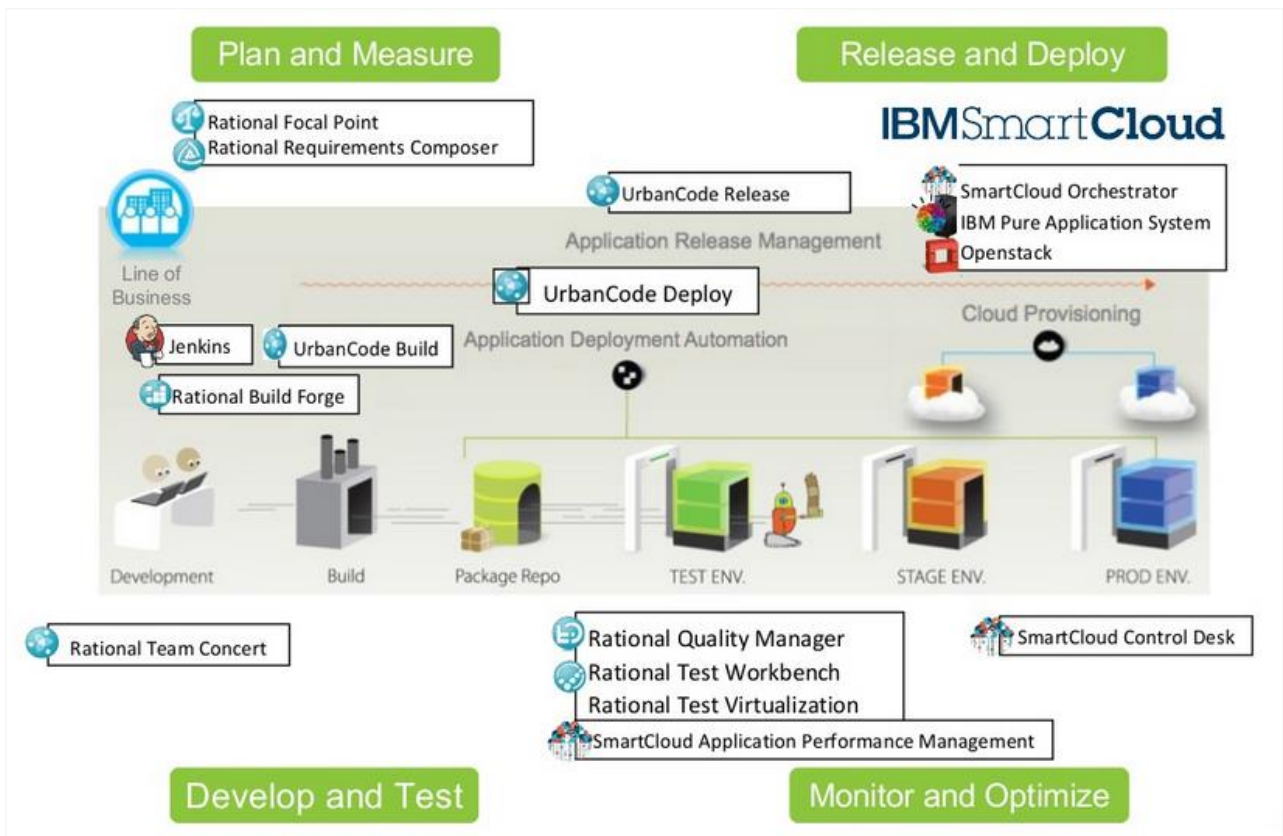
işlemci mimarisine ve 96 GB yüklü belleğe (RAM) sahip olan bir bilgisayar üzerinde gerçekleştirilmiştir. Uygulama geliştirme, kodlama ve test işlemlerinde Microsoft Visual Studio 2019 Enterprise Edition editörü ve C# 64 bit programlama dili kullanılmıştır.

Manuel dağıtım konusunda uzman bir insan kaynağı çalışmaya dâhil edilemediği için standart el yordamıyla(manuel) bir dağıtımın ortalama tamamlanma süresi hesaplanamamıştır. Bu nedenle süre bakımından otomatik dağıtım süreci ile kıyaslama yapmak mümkün olmamıştır.

## 4. UYGULAMA

### 4.1. Uygulama Mimarisi

*IBM UrbanCode Deploy*, uygulama dağıtım ve çalıştırma süreçlerinde yükleme, çalıştırma, izleme ve kontrol işlevleri olan bir yayımlama çözümdür. Uygulamaların dağıtımını otomatikleştirebilen derleme ve test araçlarını içerir. Şirket içi, bulut ve ana bilgisayar uygulamalarıyla uyumlu çalışır. Bu uygulamaya ait dağıtım mimarisi Şekil 5’de gösterilmektedir.



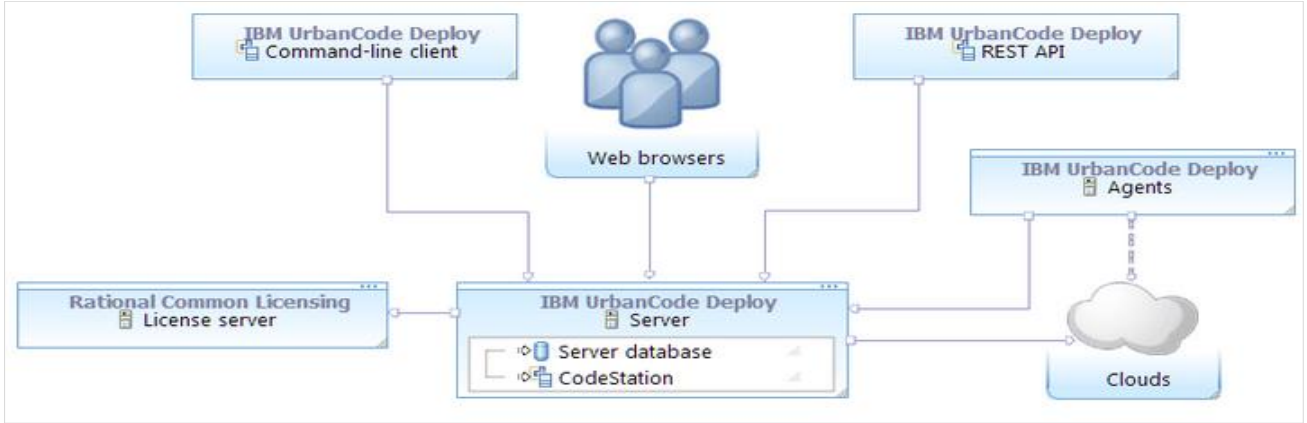
Şekil 5. Dağıtım mimarisi (UrbanCode, 2022)

Geliştirilen veya güncellenen yazılımlar ‘Jenkins’, ‘TeamCity’, ‘MS TFS Build’ gibi derleme araçları ile derlendikten sonra dağıtım yapılacak bir depolama kaynağına transfer edilir. IBM udeploy aynı zamanda kendi depolama kaynağına sahiptir. Ancak bu çalışmada harici bir depolama kaynağı tercih edilmiştir. Derlenen uygulama kodu, dağıtım hazır bir yazılım çıktısı olarak depolama kaynağında saklanır. Dağıtım aşamasında, IBM UrbanCode Deploy aracı önceden dağıtım hazırlanan dağıtım diyagramına göre süreçleri çalıştırır, dağıtım yapılacak uygulamayı test, üretim (PROD) gibi önceden hedeflenen ortamlarda gerçekleştirir.

Uygulamanın temel kurulumunda bir sunucu, ajanlar ve lisans sunucu ihtiyacı vardır. Kullanılan ajanların iki tipi vardır. Uygulamanın sürüm 7 ve sonrasında, sunucuya WebSocket ve güvenli (https) bağlantı yöntemini kullanır. Önceki sürümlerinde ise JMS ve güvenli (https) bağlantısı kullanır. Bu ajanlar bulut ortamlarına, sanal makinelere (VM'ler), kapsayıcılara veya fiziksel



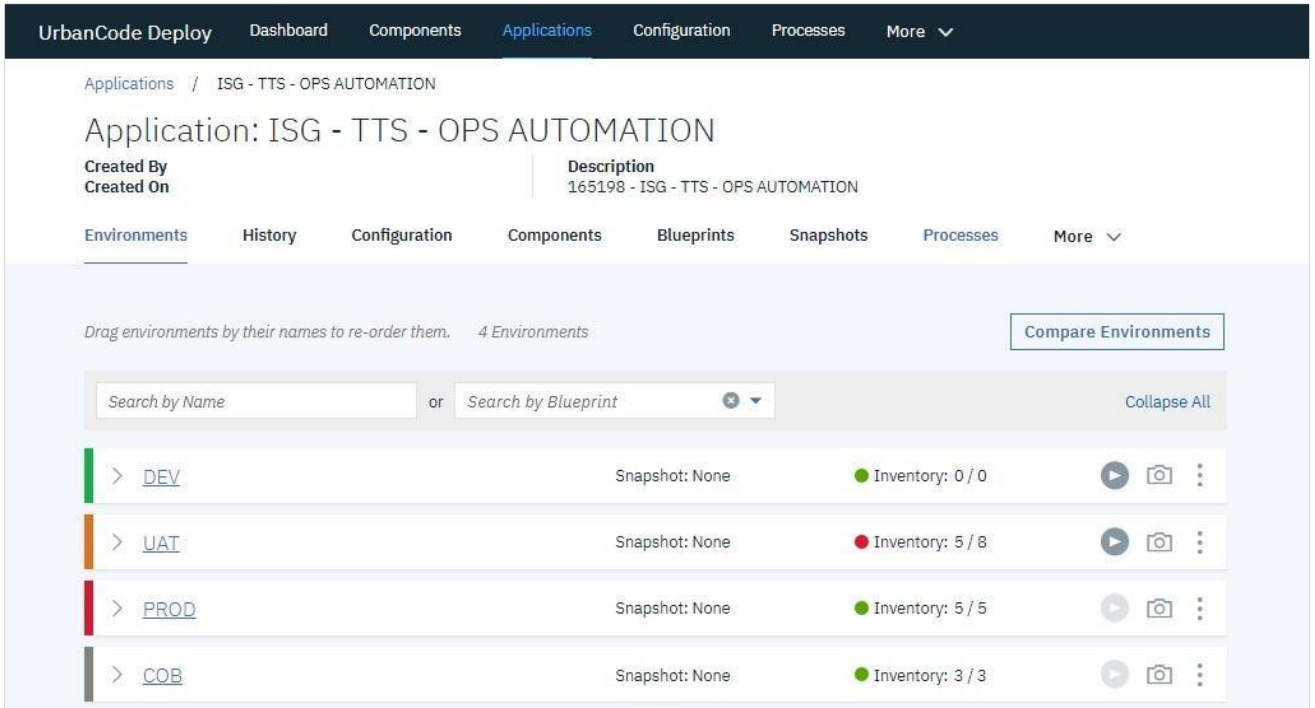
sistemlere yüklenebilir, birçok farklı türdeki sisteme kurulabilir. Uygulamaya ait çekirdek topolojisi Şekil 6’da gösterilmektedir.



Şekil 6. Çekirdek topoloji (IBM\_UrbanCode, 2022)

## 4.2. Uygulama Genel Görünümü

UrbanCode Deploy dağıtım uygulaması çalıştırıldığında Şekil 7’de gösterilen şekilde bir ekran ara yüzü elde edilmektedir.



Şekil 7. Uygulama genel görünümü

Uygulama kullanım kılavuzu, entegrasyon, destek gibi ana başlıklar yanında oluşturulan bir projeye ait uygulama geliştirme ortamı (DEV), kullanıcı kabul testi (UAT), üretim ortamı canlı (PROD) ve iş sürekliliği (COB) çalışma ortamları yer almaktadır. Yine uygulamaya ait dağıtım geçmişi, konfigürasyon yönetimi, bileşen tanımları, takvim, değişiklikler, geçmiş işlemler gibi alt menü başlıkları yer almaktadır.

### 4.3. Planlama

Bu çalışmada otomatik dağıtım aracı olarak yaygın kullanılan IBM UrbanCode Deploy (UDeploy) uygulaması kullanılmıştır. Bu başlık altında yetkilendirme, uygulama bileşen ve konfigürasyonu gibi uygulama alanlarına ilişkin plan ve ön tanımlamalar gerçekleştirilmiştir.

#### 4.3.1. Yetkilendirme

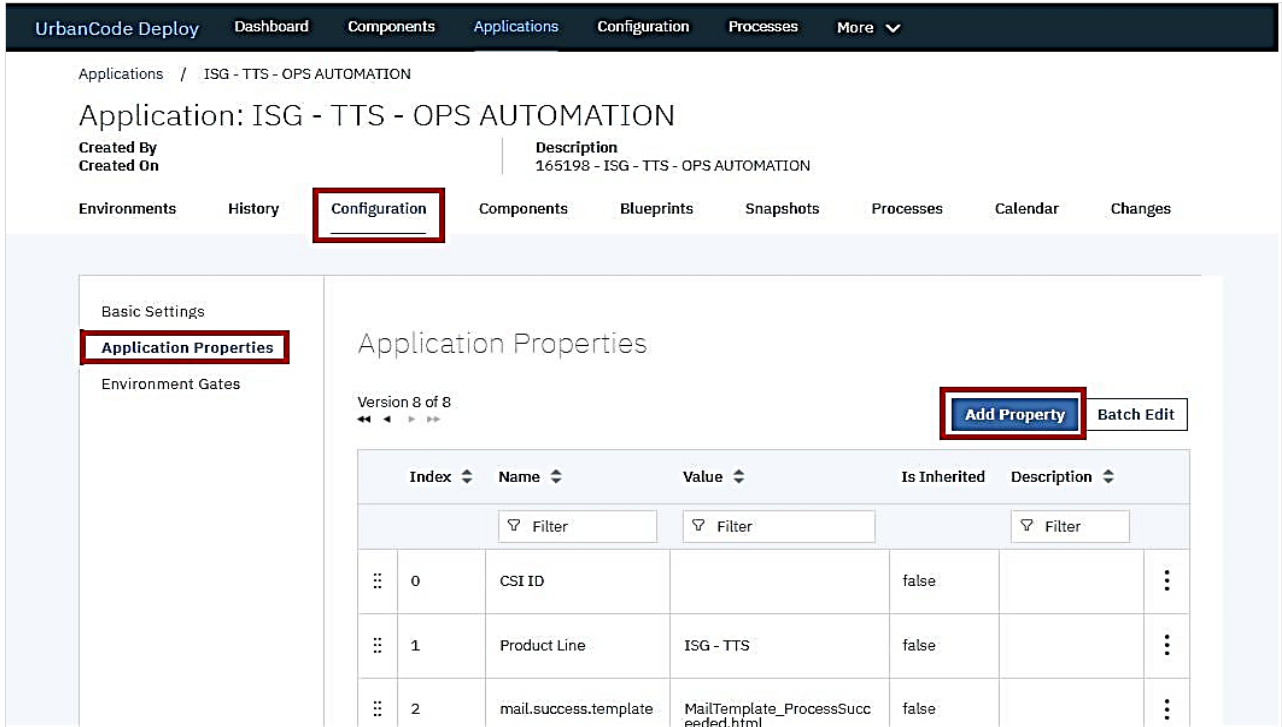
Bir dağıtım sürecinde ihtiyaç duyulan uygulama kullanıcı ve yetkilerine ilişkin örnek rol ve tanımları Tablo 2’de sunulmaktadır.

**Tablo 2:** Kullanıcı Yetkilendirme Rol ve Tanımları

Rol	Yetki Tanımı
Yönetici ( <i>‘Udeploy Admin’</i> )	En üst yetkiye sahip kullanıcı rolüdür. Yeni bir uygulama oluşturma, diğer rollerin yönetimi ile canlı üretim( <i>production</i> ) ve COB ortam sunucu ilişkilendirmelerini gerçekleştirir.
Konfigürasyon Yöneticisi ( <i>‘Configuration Manager’</i> )	Süreç tasarımı, yapılandırma tanımlama ve düzenleme, uygulamanın dağıtımı ve UAT ortamına onay verme yetkisine sahiptir. Ancak canlı üretim ve COB ortamına dağıtım başlatamaz ve onaylayamaz.
Uygulama Yöneticisi ( <i>‘Application Manager’</i> )	UAT ortamına dağıtım( <i>deployment</i> ) başlatma ve onay yetkisi bulunur. Süreç tasarım ve yapılandırma, Canlı ve COB ortamına dağıtım yetkisine sahip değildir.
Ürün Destek ( <i>‘Production Support’</i> )	Canlı ve COB ortamına onay gerekecek şekilde dağıtım başlatabilecek yetkiye sahip bir roldür. Süreç tasarım ve yapılandırma yapma yetkisine sahip değildir.

#### 4.3.2. Konfigürasyon Tanımlama

Otomatik dağıtım aşamasında çalışacak görevlere ilişkin tanımlamalar *‘administrator’* kullanıcısı tarafından uygulamanın *‘Home/Applications/Configuration’* sekmesinde yer alan *‘Add Property’* butonu kullanılarak tanımlanır. Bu özellikler ilgili görevin çalışma anında kullanılır. Çalışmada kullanılan örnek uygulama tanımların yer aldığı ekran görüntüsü Şekil 8’de yer almaktadır.



Şekil 8. Uygulama konfigürasyon tanımları



Uygulama özellikleri bölümünde yer alan değişkenler ve kullanım amaçları Tablo 3’de açıklanmaktadır. Yeni bir nitelik ekleme ‘*Add Property*’, daha önce tanımlanmış bir özelliği güncelleme ‘*Edit*’, silme ise ‘*Delete*’ alanları üzerinden gerçekleştirilir.

**Tablo 3:** Uygulama Değişkenleri

Özellik İsmi	Açıklama
CSI ID	Uygulamanın kurum envanter kaydı numarası
mail.alldisable	E-posta gönderim özelliğinin aktif - pasif yapılması
mail.allfrom	Gönderilecek e-postanın ‘kimden’ alanı
mail.allsmtp	E-posta SMTP sunucu adresi
mail.allto	Gönderilecek e-postanın ‘kime’ alanı
mail.approved.subject	E-posta ‘başlık’ şablonu
mail.approved.template	E-posta onaylanmış dağıtım için bildirim maili ‘gövde’ şablonu
mail.fail.subject	Hata almış dağıtım işleminin bildirim e-posta ‘başlık’ şablonu
mail.fail.template	Hata almış dağıtım için bildirim e-posta ‘gövde’ şablonu
mail.rejected.subject	E-posta onaylanmamış dağıtım için bildirim e-posta gövde şablonu
mail.rejected.template	Onaylanmamış dağıtım için bildirim e-posta ‘gövde’ şablonu
mail.review.subject	Dağıtımında çalıştırılacak kod parçasının gözden geçirme bildirim başlığı
mail.review.template	Dağıtımında çalıştırılacak kod parçasının gözden geçirilme bildirim içeriği
mail.start.subject	Başlatılan dağıtım işleminin bilgilendirme e-postası için başlık şablonu
mail.start.template	Başlatılan dağıtım işleminin bilgilendirme e-postası için gövde şablonu
mail.success.subject	Başarılı tamamlanan dağıtım işleminin bilgilendirme e-postası için başlık şablonu
mail.success.template	Başarılı tamamlanan dağıtım işleminin bilgilendirme e-postası için gövde şablonu
Product line	Uygulamanın kurum envanterinde bağlı bulunduğu bölüm

#### 4.3.3. İş Akışlarının Planlanması

Bu çalışmada iş akışlarının uygulamada var olan hazır ya da hazırlanan şablonlardan yararlanarak oluşturulması planlanmıştır. Aynı tür bileşenlerin dağıtım ihtiyaçlarının merkezi bir şekilde yönetimi amaçlanmıştır. Ana şablonda yapılacak bir güncellemenin bu şablondan türetilen iş akışlarına otomatik bir şekilde uygulanabilmesine imkân sunacak bir yapı kurgulanmıştır.

Bu çalışma kapsamı bir web uygulamasının farklı ortamlara dağıtımının otomatik yapılması ile sınırlandırılmıştır. Bu amaçla IIS, WEB, UD iş akış şablonları oluşturulmuştur.

**İnternet bilgi servisi (IIS):** Bu çalışmada uygulamanın internet üzerinden servis edileceği sunucu Microsoft IIS olarak planlanmıştır. Otomatik dağıtımda, IIS üzerinden yapılacak görevler için planlanan iş akışları Tablo 4’te sunulmaktadır.

**Tablo 4:** Microsoft IIS İş Akışları

İş Akışı İsmi	Açıklama
Backup Configuration	Mevcut IIS yapılandırma yedeğinin alınması
Configure Site	İnternet sitesinin yapılandırılması
Create Site	Yeni bir internet sitesinin oluşturulması
Delete Site	Mevcut internet sitesinin silinmesi
Recycle Site	İnternet sitesinin yeniden başlatılması
Start Site	İnternet sitesinin başlatılması
Stop Site	İnternet sitesinin durdurulması

**İnternet (WEB):** İnternet uygulamasının otomatik dağıtımında kullanılacak görev ve işlemlere ait akışların oluşturulmasında hazır WEB şablonlarının kullanılması planlanmıştır. Bu görevler IIS haricinde uygulamaya özel ihtiyaçlardan türetilmiştir. Bu aşamada planlanan işlemler ve kullanım amaçları Tablo 5’te tanımlanmıştır.

**Tablo 5:** WEB Dağıtım İş Akışları

İş Akışı İsmi	Açıklama
Add to Inventory	Mevcut versiyonun envanter kaydına eklenmesi
Backup Artifacts	Uygulamanın aktif sürümünün yedeğinin alınması
Clean Backups	Önceden alınmış yedeğin temizlenmesi
Deploy Artifacts	Dağıtım işlemi
Encrypt Config	'web.config dosyası'nın şifrelenmesi
Remove from Inventory	Envanter kaydından silinmesi
Replace Configuration Tokens	Ortama göre değiştirilecek dosyaların hazırlanması
Restore Artifacts	Daha önceki yedek üzerinden geri dönülmesi
Set FID	IIS üzerinde uygulama havuzuna 'Functional ID' tanımlanması.

**Uygulama Dağıtım (UD):** Otomatik dağıtım görevinin çalıştığı esnada e-posta bilgilendirmeleri, hata alınması durumunda hata kaydının tutulması gibi işlemlerin hazırlanmasında UD şablonları kullanılmıştır. Bu aşamada planlanan işlemler ve kullanım amaçları Tablo 6'de tanımlanmıştır.

**Tablo 6:**UD İş Akışları

Process	Description
Add Latest Deploy to Inventory	Son dağıtıma ait sürümün döküm kaydı.
Check Process Failure	İş akışındaki hata kontrolü
Create Snapshot	Mevcut sürüm bilgisinin kayıt altına alınması
Get Request Details	Dağıtım isteği detay bilgisi
Send Notification Mail	Bilgilendirme mailinin gönderilmesi
Set Process Failure Flag	İş akışındaki hata kontrolü bayrağına değer atanması
Set Process Start Time	İş akışının başlangıç zamanının kaydedilmesi

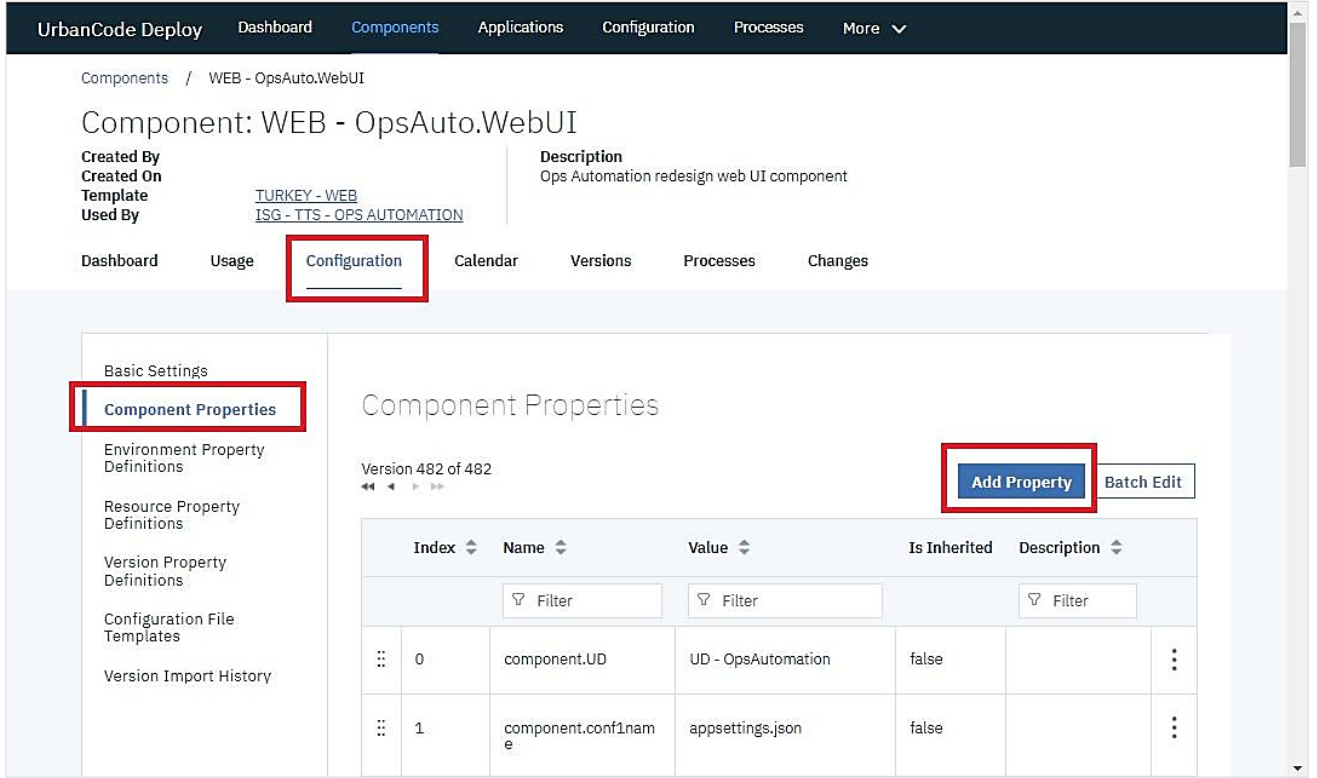
#### 4.4. Tasarım

Bu çalışmanın bileşen('component') ve iş akış tasarımları UDeploy uygulaması üzerinden gerçekleştirilmiştir. Bileşen kavramı, otomatik dağıtım yapılacak her bir tipteki uygulama ya da nesneyi ifade etmektedir. Bileşenleri ortamlara dağıtmak için uygulama süreçleri ('application process') tasarlanır ve daha sonra çalıştırılır. Bileşen ve uygulama dağıtım tasarımı bu kısımda ele alınmaktadır.

##### 4.4.1. Bileşen Tasarımları

Bileşen, dağıtım yapılacak olan her bir uygulama parçacığının ayrılması için kullanılmaktadır. Bileşenler daha önceden hazırlanmış olan şablonlardan türetilmiştir.

**İnternet (WEB) Bileşeni:** İnternet bileşenlerine ait iş akış tasarımları hazır şablonlar üzerinden gerçekleştirilmiştir. Dağıtım yapılacak uygulama paketi otomatik dağıtım sürecinin başlamasından önce derleme aracı çıktısı olarak depolama amacıyla kullanılan Linux tabanlı *Jfrog Artifactory* uygulamasına yüklenir. Udeploy, periyodik aralıklarla ya da manuel kullanıcı tetiklemesiyle bu uygulamaya bağlantı oluşturarak depo yerinde uygulamanın yeni bir sürümünün bulunup bulunmadığını kontrol eder. Eğer yeni bir sürümü varsa bu sürüm otomatik olarak bileşenin güncel sürüm bilgisi kaydedilir. Uygulama sürüm bilgilerine dair ekran görüntüsü Şekil 9'da sunulmaktadır.



Şekil 9. WEB bileşen sürüm örnekleri

Web bileşenleri aynı zamanda uygulama yapılandırma bilgilerini dağıtım yapılacak ortamlara göre farklılaştırmaya katkı sunar. Örneğin: Veri tabanı bağlantı bilgisi test ve canlı ortamlar için farklı değerler içerir. Bu değerler ortam değişkenleri kısmında değişken olarak tanımlanmaktadır. Değişkenlerin değerleri ise her ortam için ayrı ayrı girilmektedir. Dağıtım aşamasında yapılandırma dosyası girilen değerlere göre değiştirilerek ilgili uygulama klasörüne kopyalanır. Kullanılan internet bileşen özellikleri Tablo 7’de, ortam değişken ve tanımları Tablo 8’de sunulmaktadır.

Tablo 7: WEB Bileşen Özellikleri

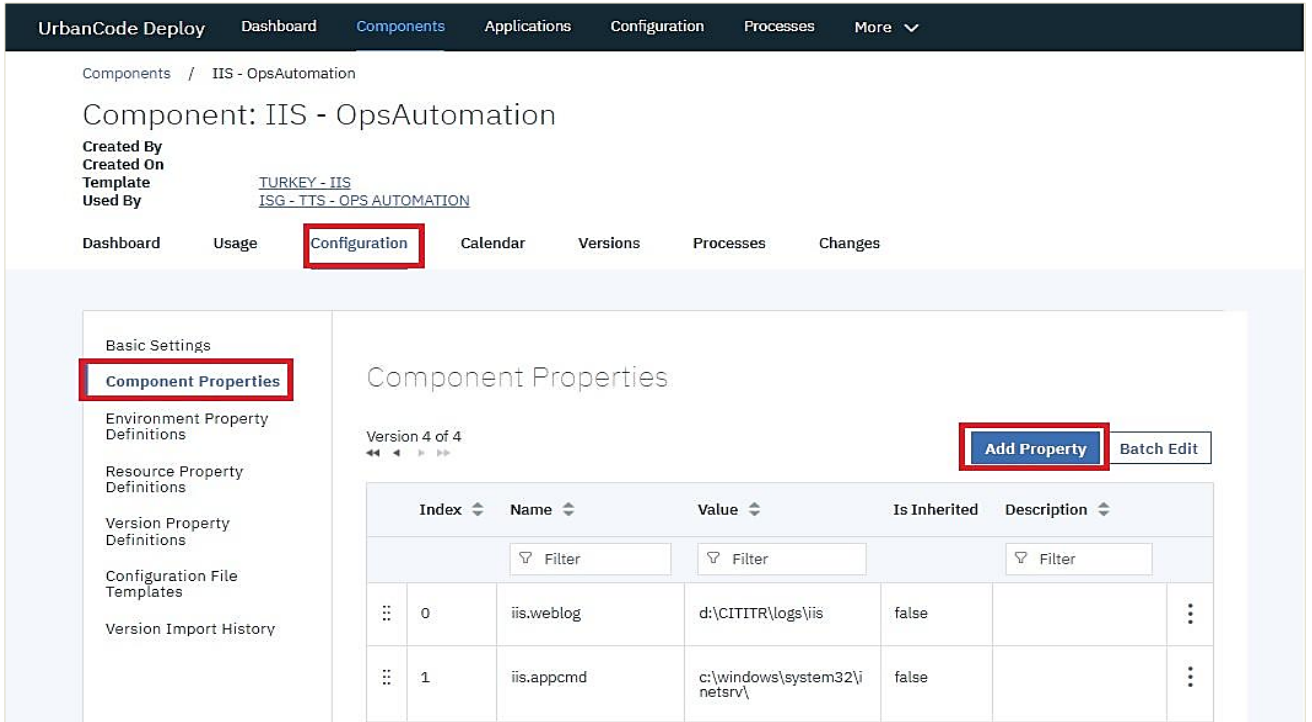
Özellik ismi	Açıklama
art.zippedartifact	Dağıtım yapılacak uygulama paketinin sıkıştırılmış olup olmadığı bilgisi.
component.artifactoffset	Dağıtım yapılacak uygulama dosyalarının klasör içerisindeki ofset bilgisi.
component.backupfolder	Yedek alınacak klasörün sunucu üzerindeki yolu.
component.clearstaging	Çalışma klasörünün operasyon sonrasında temizlenme seçeneği yer alır.
component.conf1diroffset	Tokenize edilecek dosyanın klasör içerisindeki ofset bilgisi.
component.conf1name	Tokenize edilecek dosyanın ismi
component.confdelimiterend	Tokenize edilecek dosya içerisindeki ilgili bölümün başlangıç karakteri
component.confdelimiterstart	Tokenize edilecek dosya içerisindeki ilgili bölümün bitiş karakteri.
component.latestbackuppath_COB	COB ortamında son alınan yedekleme(backup) konumu
component.latestbackuppath_PROD	Üretim (PROD) ortamında son alınan yedek konumu
component.latestbackuppath_UAT	UAT ortamında son alınan yedek konumu
component.latestbackupversion_COB	COB ortamında son alınan yedeğin sürüm bilgisi
component.latestbackupversion_PROD	Üretim (PROD) ortamında son alınan yedeğin sürüm bilgisi
component.latestbackupversion_UAT	UAT ortamında son alınan yedeğin sürüm bilgisi
component.type	Component Türü
component.UD	UD bileşen ismi
site.enable32bitapp	IIS üzerindeki uygulama havuzunun 32 bit olması bilgisi yer alır.
site.encryptconfig	web.config dosyasının şifrelenmesi bilgisi yer alır
site.IISComponent	IIS bileşenin ismi yer alır.
site.iisfolder	Uygulamanın sunucu üzerindeki klasörü
site.managedpipeline	Uygulama havuzu üzerindeki yönetilen işlem hat bilgisi

site.managedruntime	Uygulama havuzu üzerindeki yönetim çalışma zamanı bilgisi
site.name	Uygulamanın IIS üzerindeki web site ismi
site.poolname	Uygulamanın IIS üzerindeki uygulama havuz ismi
site.rapidfailprotection	Hızlı hata koruma bilgisi
tfs.latestdeploy_COB	COB ortamına yapılan son dağıtımın sürüm bilgisi
tfs.latestdeploy_PROD	Üretim (PROD) ortamına yapılan son dağıtımın sürüm bilgisi
tfs.latestdeploy_UAT	UAT ortamına yapılan son dağıtımın sürüm bilgisi

**Tablo 8:** Ortam Değişkenleri

Özellik İsmi	Açıklama
site.certfriendlyname	İnternet sitesi sertifika tanımı
site.port	İnternet sitesi port bilgisi
site.protocol	İnternet sitesi protokol bilgisi (http/https)
site.certlocation	Sunucu üzerindeki sertifikanın konumu
site.FIDType	İnternet sitesinin çalışacağı işlevsel (functional) ID tipi
site.FIDName	İnternet sitesinin çalışacağı işlevsel ID ismi
site.FIDPassword	İnternet sitesinin çalışacağı işlevsel ID şifresi
ART_SOURCE_REPO	Dağıtım edilecek paketin 'artifactory' uygulama kaynak konumu
ART_TARGET_REPO	Dağıtım yapılacak paketin taşınacak hedef konumu
conf.environment	Kişiselleştirilecek ortam verisine ait değişken
conf.connectionString	VT bağlantı tanımı gibi kişiselleştirilecek bağlantı değişken tanımı.

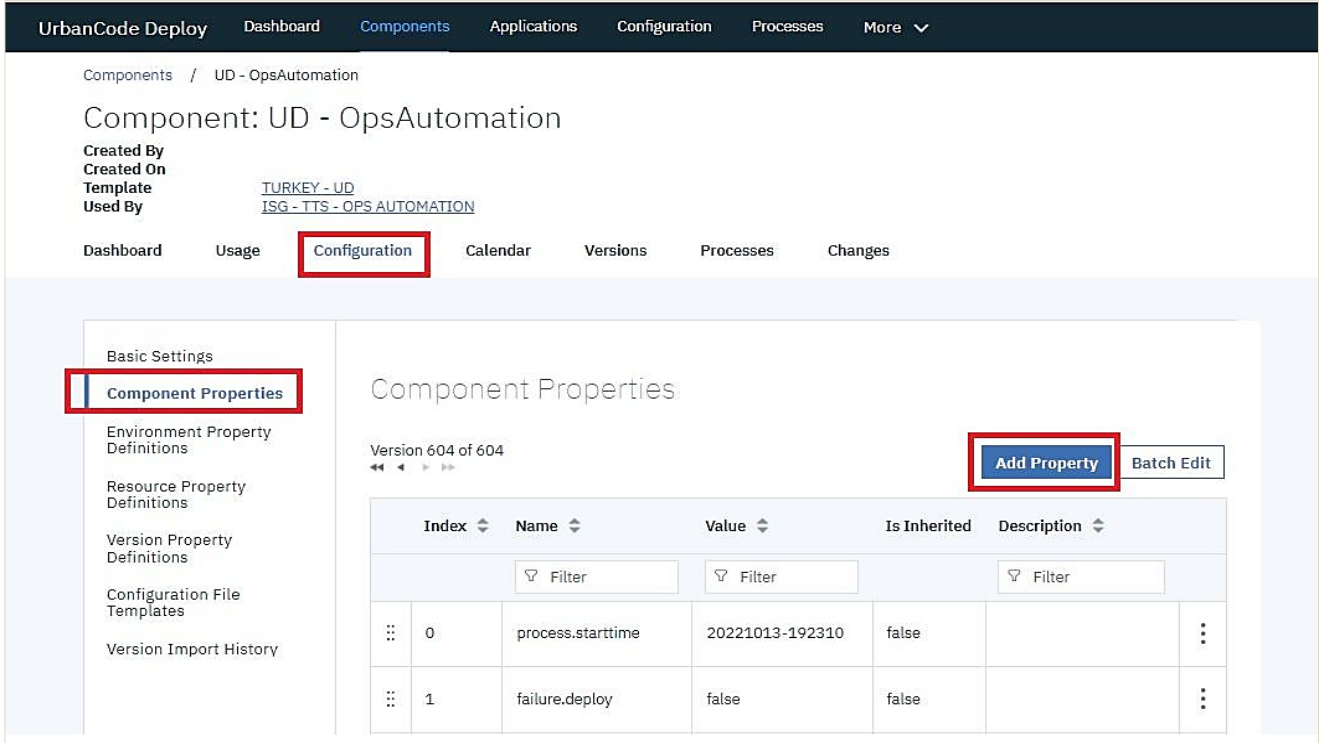
**ISS Bileşeni:** Uygulama dağıtım sürecinde derlenen uygulama paketinin sunucuya konumlandırılması aşamasında IIS tarafında yapılması gereken işlem adımları bulunmaktadır. Bu bileşen IIS hazır şablonu üzerinden türetildiği için şablonda tanımlanan iş akışları dağıtım aşamasında iş akışı tasarımında yer alır ve çalıştırılır. Bu bileşen üzerinde uygulama ya da parçası bulunmadığından sürüm yapısı söz konusu değildir. IIS bileşen ve tanımları Şekil 10'da yer almaktadır.



**Şekil 10.** IIS Bileşen Özellikleri

**UD Bileşeni:** Otomatik dağıtım sürecinde kullanılması planlanan yardımcı bir bileşen olarak tasarlanmıştır. Bileşen, UD iş akış şablonlarından tasarlanır. Üzerinde herhangi bir uygulama

bulunmadığı için bu bileşen de yardımcı bir bileşen olup IIS bileşeni gibi sürüm yapısı söz konusu değildir. Ortam değişkenleri bulunmamaktadır. Bu bileşene ait tasarım özellikleri Şekil 11’de gösterilmektedir.



UrbanCode Deploy Dashboard Components Applications Configuration Processes More

Components / UD - OpsAutomation

Component: UD - OpsAutomation

Created By  
Created On  
Template Used By

TURKEY - UD  
ISG - TTS - OPS AUTOMATION

Dashboard Usage Configuration Calendar Versions Processes Changes

Basic Settings  
Component Properties  
Environment Property Definitions  
Resource Property Definitions  
Version Property Definitions  
Configuration File Templates  
Version Import History

Component Properties

Version 604 of 604

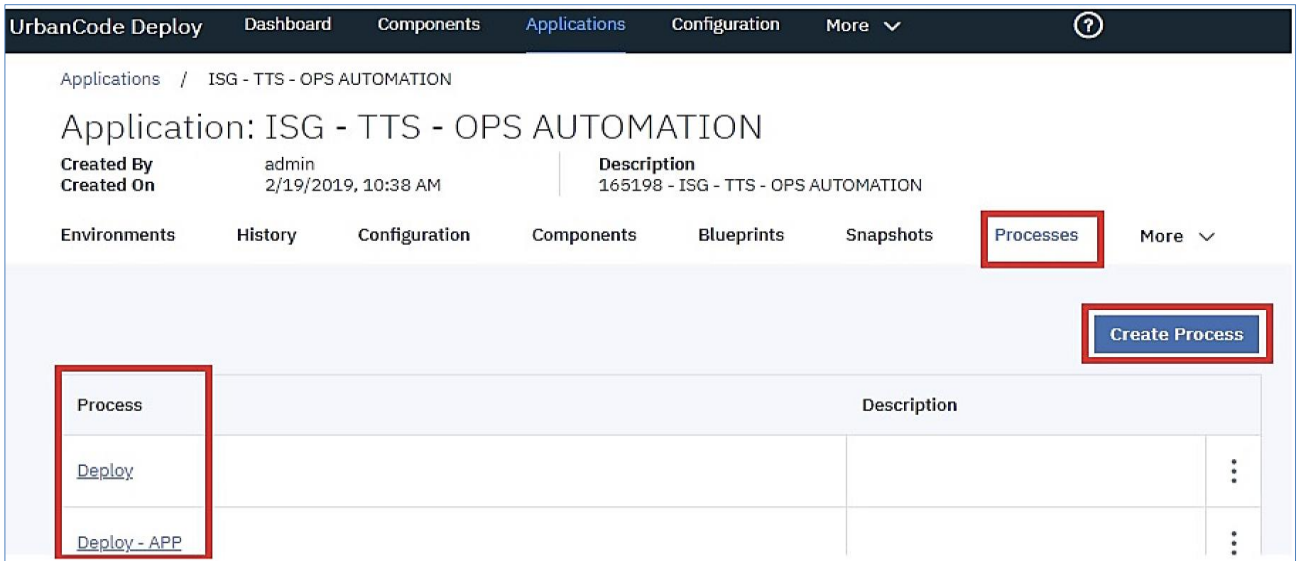
Add Property Batch Edit

Index	Name	Value	Is Inherited	Description
0	process.starttime	20221013-192310	false	
1	failure.deploy	false	false	

Şekil 11. UD bileşen özellikleri

#### 4.4.2. Uygulama İş Akışı Tasarımı

Bileşen hazır şablonları ve üzerlerinde tanımlı bulunan işlemler dağıtım sürecinin iş akışını oluşturmaktadırlar. Oluşturulan bu ana süreç ve bileşenler birbirleriyle ilişkili biçimde tasarlanır. Dağıtım aşamasında kullanılacak özelliklere ait değişkenlerin tanımı da bu bölümde gerçekleştirilmektedir. Uygulamaya ait iş akış listesinin örnek ekran Şekil-12’de gösterilmektedir.



UrbanCode Deploy Dashboard Components Applications Configuration More

Applications / ISG - TTS - OPS AUTOMATION

Application: ISG - TTS - OPS AUTOMATION

Created By: admin  
Created On: 2/19/2019, 10:38 AM  
Description: 165198 - ISG - TTS - OPS AUTOMATION

Environments History Configuration Components Blueprints Snapshots Processes More

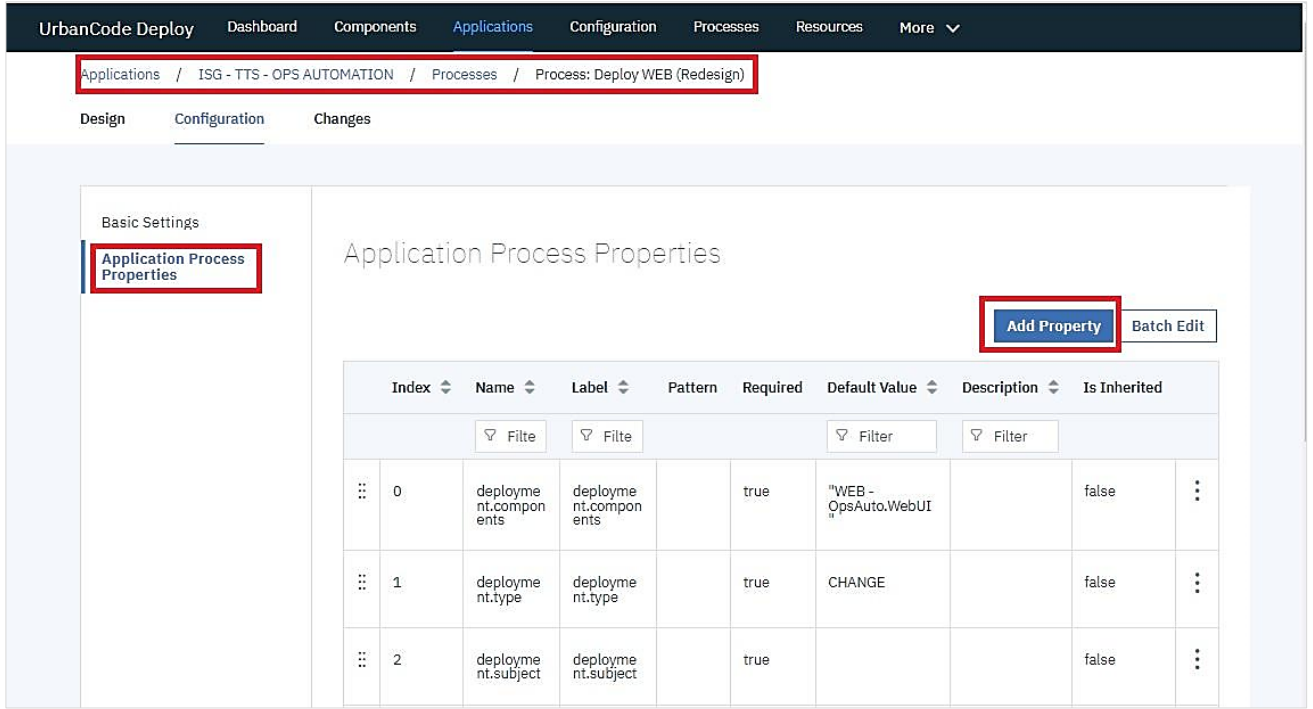
Create Process

Process	Description
Deploy	
Deploy - APP	

Şekil 12. İş akış listesi

İş akışı tasarımı UDeploy süreç tasarımı menüsü bileşenler(‘components’) menüsü altında yer alan süreçler (‘processes’) sekmesi ‘Create Process’ düğmesi kullanılarak gerçekleştirilir. Mevcut bir

akışın güncellenme, kopyalama veya silme işlemi de bu ekran aracılığıyla gerçekleştirilmektedir. Uygulamada kullanılan iş akışı özelliklerine ait örnek ekran görüntüsü Şekil 13'te sunulmaktadır.



Şekil 13. Uygulama iş akışı özellikleri

Otomatik dağıtım sürecinin tasarımında Tablo 9'da yer alan işlem adımları belirlenen sırada çalışacak şekilde tasarlanmıştır.

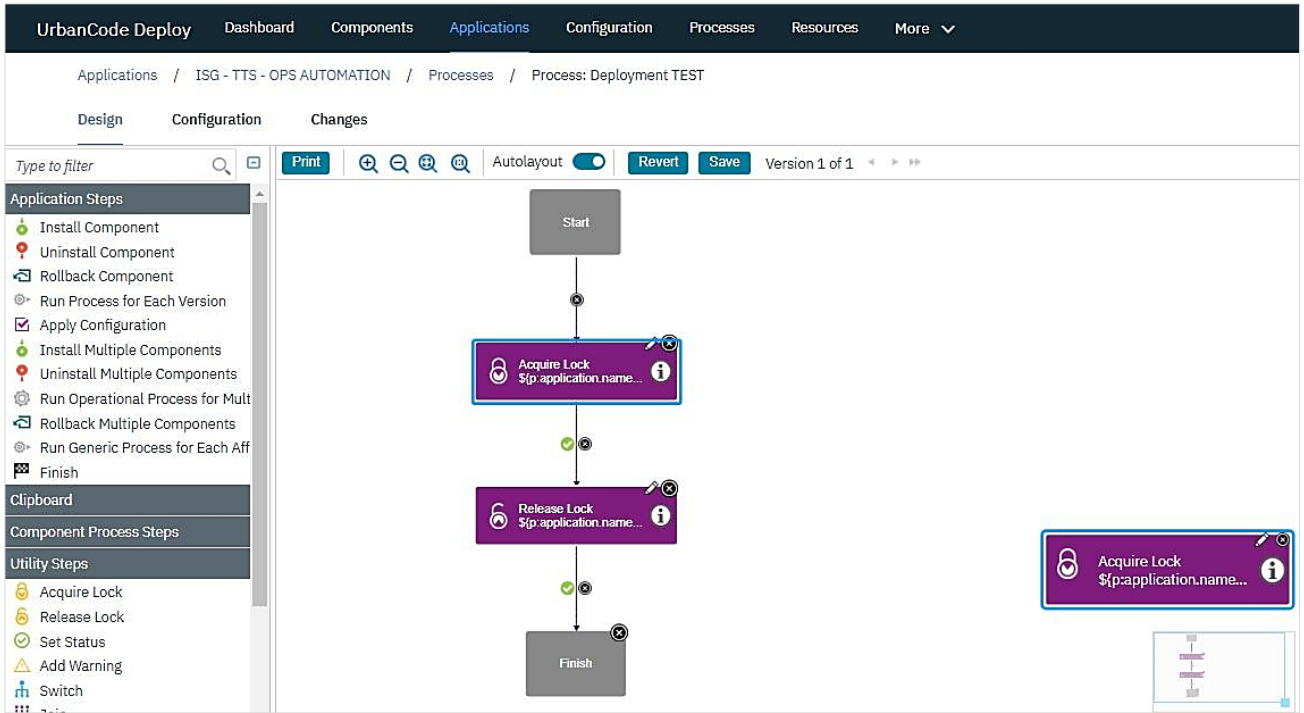
Tablo 9: Dağıtım Süreci İşlem Adımları

Görev İsmi	Açıklama
Start	Sürecin Başlangıç Adımı
Aquire Lock	Aynı sürecin birden fazla çalıştırılmasının engellenmesi
Sen Notification Mail(started)	Başlandı bilgisinin e-mail ile iletilmesi
Environment Selection	Deployment yapılacak ortamın bilgisi kontrolü
Ticket Validation	Canlı ortam için değişiklik kaydının sorgulanması.
Set Process Failure Flag	Süreçte hata takibi için bayrağın başlangıç değeri atama
Set Process Start Time	Sürecin başlangıç zamanının atanması
Create Rollback Snapshot	Mevcut sürümüne ait görüntünün kaydedilmesi
Backup	Mevcut uygulamanın yedeğinin alınması.
Stop Site	IIS üzerindeki site ve uygulama havuzunun durdurulması
Deploy	Dağıtım işlemi.
Add to inventory	Envanter kaydının güncellenmesi
Create Web	Deployment türü INITIAL olarak seçilmişse IIS altında site ve diğer bileşenlerin oluşturulması
Check Process Failure	Süreçte hata kontrolü adımı
Start Site	Deployment sonrası uygulamanın başlatılması
Send Notification Mail	Tamamlandı bilgisinin e-mail ile alıcılara iletilmesi
Release Lock	Süreç kilidinin kaldırılması ve yeni dağıtıma izin verilmesi
Finish	Sürecin bitiş adımı

Bu işlem adımları çalışırken herhangi bir hata oluşması durumunda tasarımda karar verilen önceki sürüme otomatik geri dönlür. Bu hata dağıtım işlemine başlamadan önce gerçekleşmesi durumunda uygulama henüz dağıtım gerçekleştirilmediğinden süreç hata olduğu bilgisi



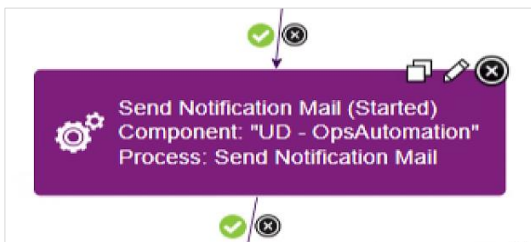
işaretlenerek sonlandırılmıştır. Süreç tasarımı, UDeploy ürünü üzerinde yer alan tasarım ('design') menüsü aracılığıyla yapılmaktadır. Süreç tasarımı örnek ekran görüntüsü Şekil 14'te sunulmaktadır.



Şekil 14. Süreç tasarımı örneği başlangıç bölümü

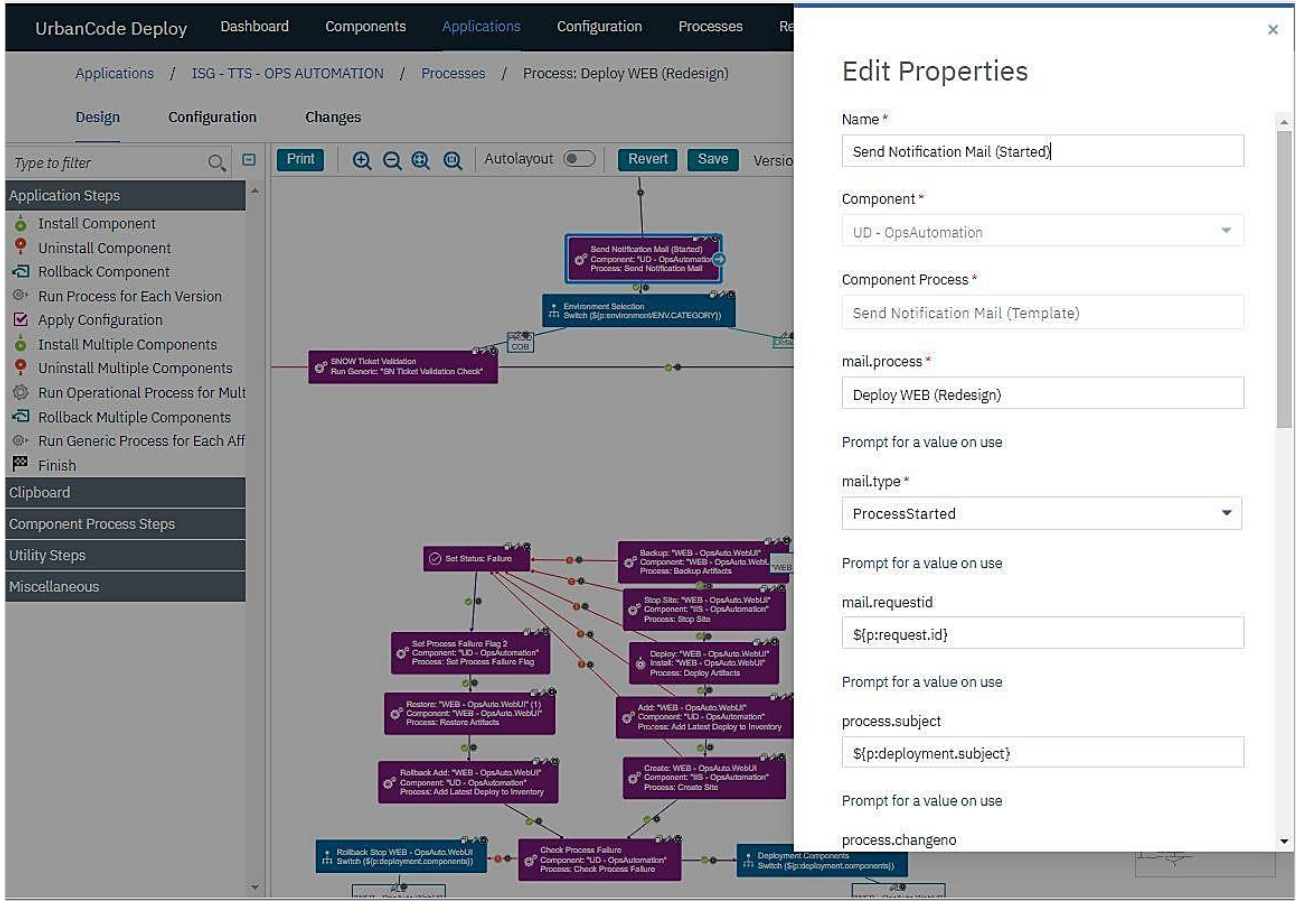
Tasarım menüsünde sunulan hazır bileşen şablonları ya da menünün sol bölümünde yer alan hazır görev listesi üzerinden sürükle-bırak yöntemi ile form üzerine yerleştirilir. Ardından bileşen üzerinde ilk olarak çalıştırılacak işleme ait bilgi tanımlanır. Başarıyla tamamlanan bir işlemten sonra süreç akışının devam edeceği bir sonraki işlem adımına bağlantı tanımlanır. Bu adıma geçişte bir hata ile karşılaşması durumunda yönleneceği görev ise bağlantı ok işaretinin ortasında bulunan 'check' sembolü tıklandıktan sonra hata okuna dönüştürülerek gerçekleştirilir. Her bir süreç 'start' adımıyla başlar ve 'finish' adımıyla sonlanır.

Bir bileşen işlemi ('component process') eklendikten sonra üzerinde, görev ismi, hangi bileşene ait olduğu ve işleme ait isim bilgisi görülmektedir. Buna ait örnek görüntü Şekil 15'te yer almaktadır.



Şekil 15 Bileşen işlemine ait örnek görünüm

Tasarım aşamasında bileşene ait değişken değerlerinin girilmesi gerekmektedir. Her bir işlem sağ üst köşesinde bulunan kalem sembolü ile düzenleme işlemi yapılmaktadır. Bu değişkenlerin değerleri başka bir değişkenin değerinden atanması şeklinde olması durumunda  $\{p: \text{değişken}\}$  şeklinde tanımlanır. Bu değişken değeri, süreç adımı başlatılırken kullanıcı tarafından girilen değerdir. Değişken yerine sabit değer verilmek istendiğinde değişken yerine sabit değer yazılır. Her iki türe ait tanımlama örneği Şekil 16'da gösterilmektedir.



Şekil 16. 'Send Notification Mail' parametre değerlerinin tanımlanmasına dair örnek ekran görüntüsü

Tasarım ekranında görünen oklar ile bileşenler bir iş akışı oluşturacak şekilde birbirlerine bağlanmıştır. Ok işaretlerinin üzerinde bulunan yeşil ve kırmızı ibareler bir sonraki bileşene iş akışını yönlendirirken hata ile karşılaşıldığında ya da önceki adımın başarılı çalışması sonucu devam edeceğini ifade etmektedir.

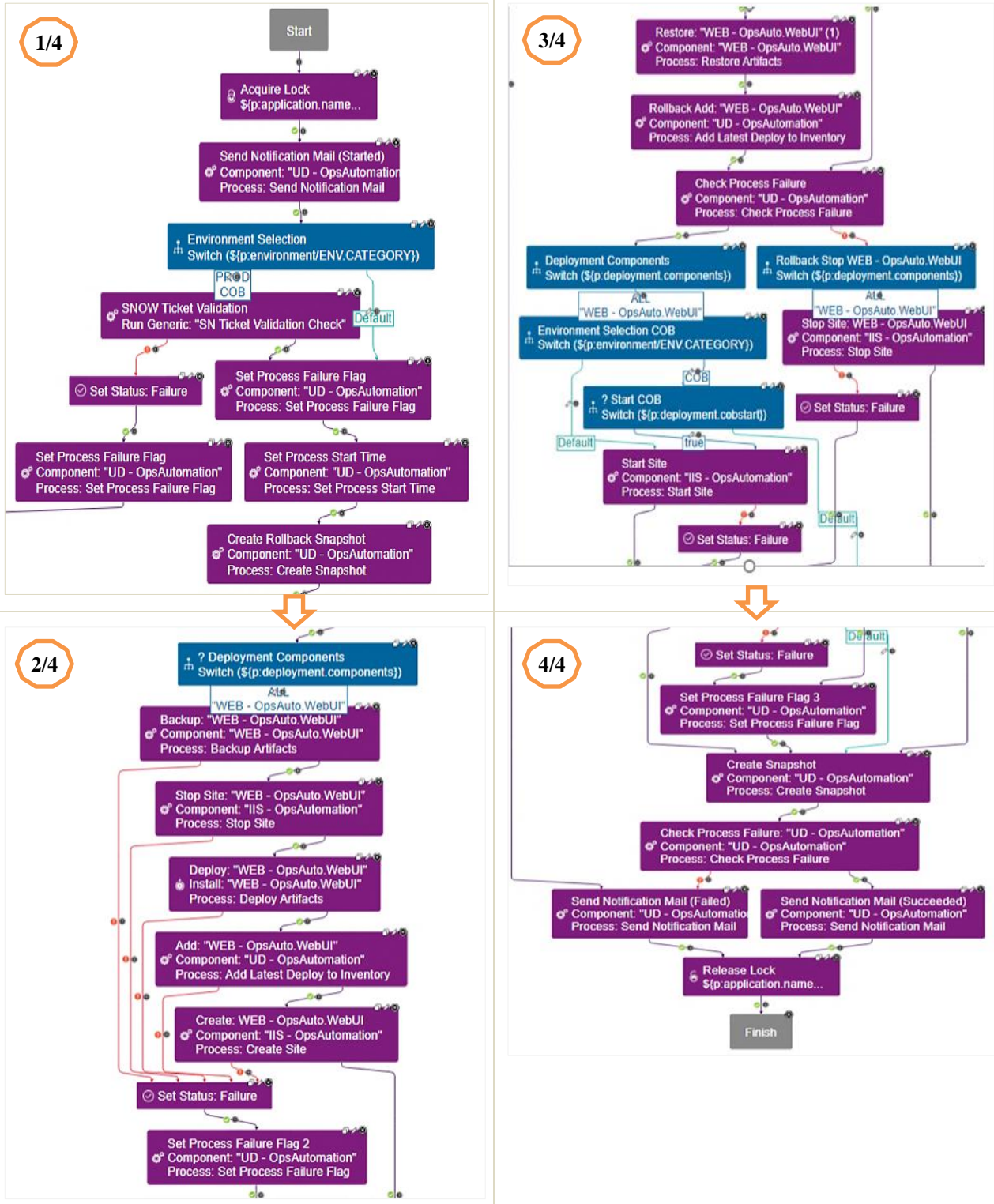
Tasarımda karar verme ('switch') bileşenleri kullanılmıştır. Bu bileşen iş akış akında, daha önceden tanımlanmış olan bir değişkenin değerine göre karar verip akışı istenilen yönde sürdürülmesi amaçlanmıştır. Örneğin, Şekil 16'da görünen 'Environment Selection' karar verme nesnesi dağıtım yapılacak ortama (UAT, COB, PROD) göre iş akışının yönlendirilmesi sağlanmıştır. Bu kullanımda, COB ve PROD için akış sol tarafta bulunan 'SNOW Ticket Validation' adımına yönlendirilirken, diğer bütün değerler için ikinci sağdaki ok ile devam eden akışı takip edecek şekilde tasarlanmıştır.

UD Bileşeni altında tasarlanmış olan 'Send Notification Mail' süreci, tasarım anında şablon olarak kullanılması planlanan değişkenleri içerir. Belirlenen değişken tipine göre bazı alanlar metin olarak girilebileceği gibi bazı alanların sadece belirlenen değerlerin girilmesi şeklinde tasarlanmıştır.

Değişkenlerin altlarında bulunan 'Prompt for value on use' seçeneği ise, değeri Şekil 16'da görünen özellik güncelleme ekranından girilmesi yerine, iş akışının başlatıldığı ekrandan çalıştırılma anında girileceğini ifade eder. Bu çalışmaya konu olan uygulamada sürece insan müdahalesinin mümkün olduğunca az olmasına özen gösterilmiştir. Dağıtım süreç tasarımı, eldeki kaynaklar kullanılarak olabildiğince otomatik işletilecek şekilde gerçekleştirilmiştir. Süreç tasarımı iş ve kurum hedeflerine uygun şekilde belirlenmelidir.

Çalışmada tasarımı gerçekleştirilen dağıtım sürecinin bütünü Şekil 17'de gösterilmektedir.



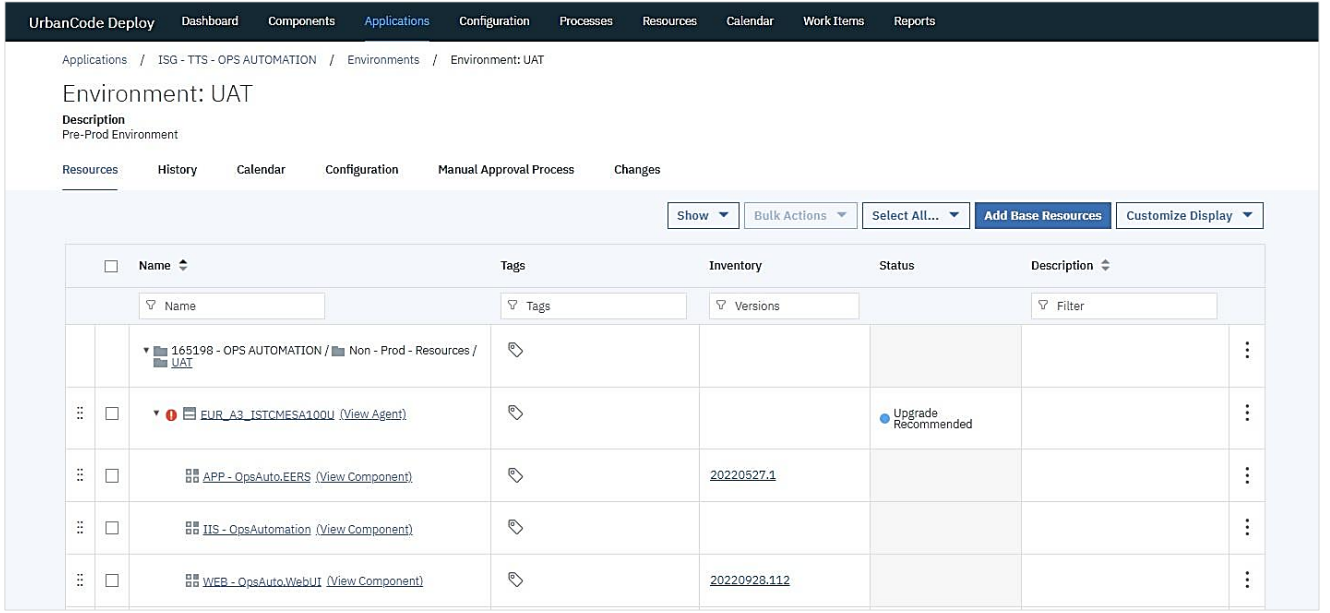


Şekil 17. Otomatik Dağıtım Uygulama Süreci Tasarımı

Süreç işlem adımlarının çalıştırılma aşamasında herhangi bir hata ile karşılaşılması durumunda hata adımlarına yönlenerak, bir önceki adımda alınan yedeği tekrar uygulama klasörüne taşıyacak ('restore') şekilde tasarım gerçekleştirilmiştir. Bir hata sebebiyle akış yedekten döndüğü için uygulama otomatik olarak yeniden başlatılmaz. Sürecin tasarımında bir hata ile karşılaşıldığı durumda sürecin nasıl bir yöntemle işletileceğine önceden karar verilmelidir. Bu karşılaşılan hatanın çözülmesi ve yeniden derlenmesi ya da yedekten değişiklik öncesi duruma geri dönüş şeklindedir.

Uygulama ('Application') menüsünden kaynak ('Environment / Resources') alt menüsü kullanılarak ilgili uygulama seçilerek açılan ekrandan sürükle-bırak yöntemi ile ilgili bağlantı tanımlanır.

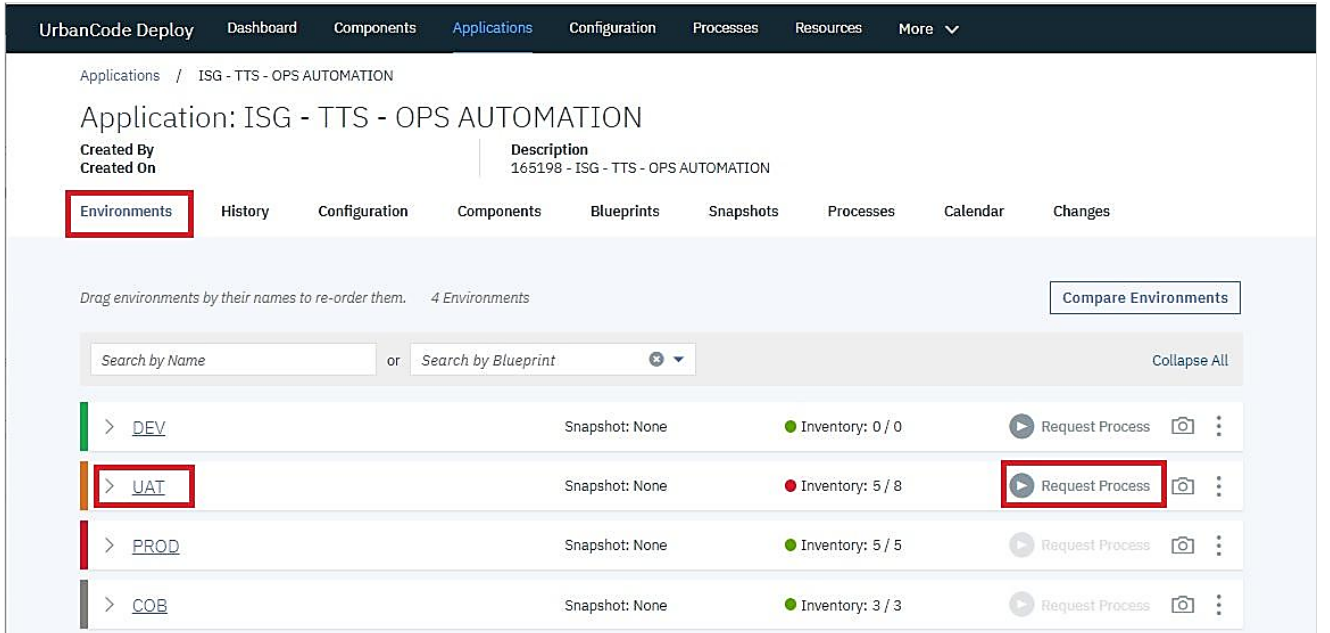
yapılır. Uygulamanın dağıtımının yapılacağı sunucu vb. bileşen bağlantı tanımları Şekil 18’de yer alan uygulama ekranı üzerinden gerçekleştirilir.



Şekil 18. Sunucu ve bileşen ilişkilendirme ekranı

#### 4.5. Dağıtım

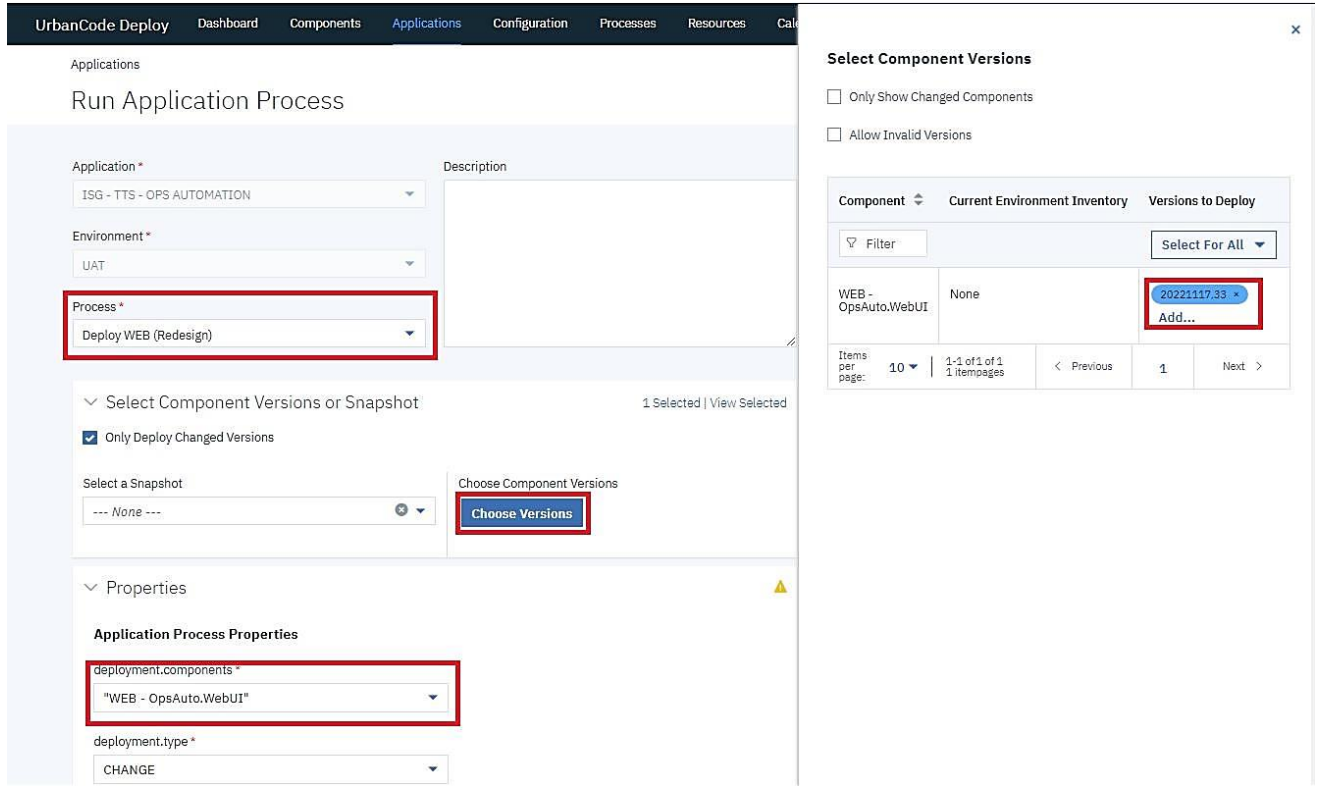
Bu aşama uygulamanın çalıştırılacağı ortamlara güvenli bir şekilde taşıma ve yaygınlaştırma aşamasıdır. Dağıtım yapılacak ortam seçimi ‘Application’ menüsü üzerinden ilgili uygulama seçildikten sonra ‘Application/Environment’ alt menüsü üzerinden dağıtım edilecek ortam satırındaki ‘Request Process’ komutu verilir. Dağıtım yapılacak ortamlara ilişkin uygulama ekran görüntüsü Şekil 19’da gösterilmektedir.



Şekil 19. Dağıtım yapılacak ortam seçimi

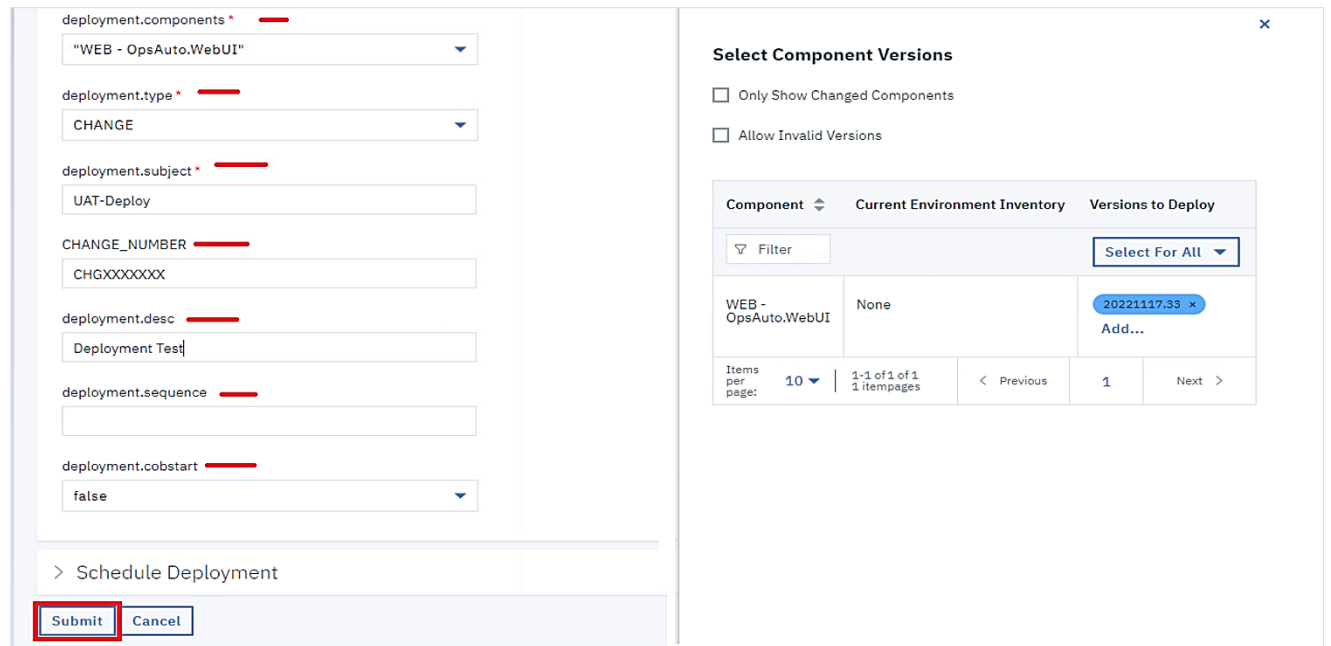
## 4.6. Çalıştırma

Dağıtım işleminden hemen önce uygulama sürüm bilgisi ve özellik tanımları Şekil 20’de yer alan uygulama ekranı ara yüzü üzerinden gerçekleştirilir. Bileşen sürüm seçimi (‘Choose Versions’) düğmesine tıklandıktan sonra ekranın sağ tarafında yer alan bileşen sürümü üzerinden gerçekleştirilir.



Şekil 20. Dağıtım süreci ön tanımlama

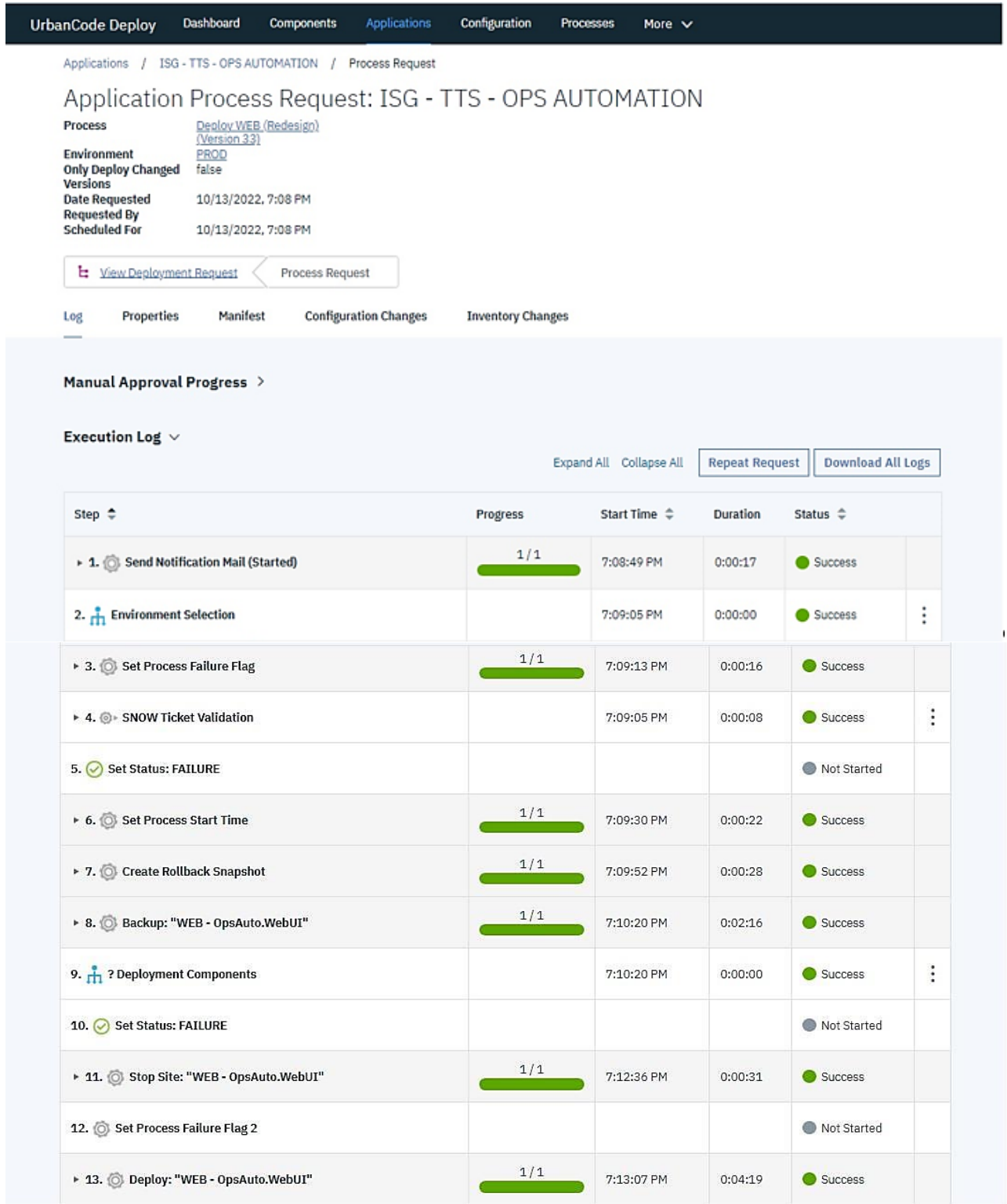
Sürüm seçiminden sonra dağıtım işleminde kullanılacak diğer değişkenlerin değerleri Şekil 21’de yer alan uygulama ekranı ara yüzü üzerinden girilir. Ardından dağıtım süreci ‘Submit’ komutu verilerek çalıştırılır.



Şekil 21. Dağıtım işlemine ait değişken girişleri

## 4.7. İzleme

Dağıtım süreci çalıştırılmaya başlatıldıktan sonra çalışan her bir adımda gerçekleşen işlemlere ait iz kaydı('log') bilgileri uygulama ekranları aracılığıyla anlık olarak takip edilmektedir. Çalıştırılan her bir adımın ne zaman başladığı ne kadar sürdüğü, durumu gibi bilgilerin yer aldığı izleme ekranına ait görüntü Şekil 22'de yer almaktadır.



UrbanCode Deploy Dashboard Components Applications Configuration Processes More

Applications / ISG - TTS - OPS AUTOMATION / Process Request

### Application Process Request: ISG - TTS - OPS AUTOMATION

Process [Deploy WEB \(Redesign\) \(Version 3.3\)](#)  
Environment [PROD](#)  
Only Deploy Changed false  
Versions  
Date Requested 10/13/2022, 7:08 PM  
Requested By  
Scheduled For 10/13/2022, 7:08 PM

[View Deployment Request](#) Process Request

Log Properties Manifest Configuration Changes Inventory Changes

#### Manual Approval Progress >

#### Execution Log

Expand All Collapse All Repeat Request Download All Logs

Step	Progress	Start Time	Duration	Status
1. Send Notification Mail (Started)	1/1	7:08:49 PM	0:00:17	Success
2. Environment Selection		7:09:05 PM	0:00:00	Success
3. Set Process Failure Flag	1/1	7:09:13 PM	0:00:16	Success
4. SNOW Ticket Validation		7:09:05 PM	0:00:08	Success
5. Set Status: FAILURE				Not Started
6. Set Process Start Time	1/1	7:09:30 PM	0:00:22	Success
7. Create Rollback Snapshot	1/1	7:09:52 PM	0:00:28	Success
8. Backup: "WEB - OpsAuto.WebUI"	1/1	7:10:20 PM	0:02:16	Success
9. Deployment Components		7:10:20 PM	0:00:00	Success
10. Set Status: FAILURE				Not Started
11. Stop Site: "WEB - OpsAuto.WebUI"	1/1	7:12:36 PM	0:00:31	Success
12. Set Process Failure Flag 2				Not Started
13. Deploy: "WEB - OpsAuto.WebUI"	1/1	7:13:07 PM	0:04:19	Success



14. 🔄 Restore: "WEB - OpsAuto.WebUI" (1)					● Not Started	
▶ 15. 🔄 Add: "WEB - OpsAuto.WebUI"	1 / 1	7:17:26 PM	0:01:33	● Success		
▶ 16. 🔄 Create: WEB - OpsAuto.WebUI	1 / 1	7:18:59 PM	0:00:02	● Success		
17. 🔄 Rollback Add: "WEB - OpsAuto.WebUI"					● Not Started	
▶ 18. 🔄 Check Process Failure	1 / 1	7:19:01 PM	0:00:02	● Success		
19. 🔄 Rollback Stop WEB - OpsAuto.WebUI					● Not Started	
20. 🔄 Deployment Components		7:19:02 PM	0:00:00	● Success		⋮
21. 🔄 ? Start COB					● Not Started	
22. 🔄 Environment Selection COB		7:19:03 PM	0:00:00	● Success		⋮
23. 🔄 Stop Site: WEB - OpsAuto.WebUI					● Not Started	
24. 🟢 Set Status: FAILURE					● Not Started	
▶ 25. 🔄 Create Snapshot	1 / 1	7:19:44 PM	0:00:27	● Success		
▶ 26. 🔄 Start Site	1 / 1	7:19:03 PM	0:00:41	● Success		
27. 🔄 Set Process Failure Flag 3					● Not Started	
28. 🟢 Set Status: FAILURE					● Not Started	
▶ 29. 🔄 Check Process Failure: "UD - OpsAutomation"	1 / 1	7:20:11 PM	0:00:01	● Success		
▶ 30. 🔄 Send Notification Mail (Succeeded)	1 / 1	7:20:12 PM	0:00:15	● Success		
31. 🔄 Send Notification Mail (Failed)					● Not Started	
32. 🔄 Set Process Failure Flag					● Not Started	
33. 🟡 Release Lock		7:20:27 PM	0:00:00	● Success		⋮
Total Execution	15 / 15	7:08:49 PM	0:11:39	● Success		

Şekil 22. Dağıtım işlemlerine ait izleme ekranı

#### 4.8. Kontrol

Dağıtımı tamamlanan bir uygulamaya yönelik erişim yetkileri, güvenlik gibi kurum ihtiyaç ve beklentilerine dair ek kontroller uygulanır. İzleme ekranından aynı zamanda çalıştırılan adımların yanlarında bulunan ok işaretleriyle o adıma dair çalıştırılan alt süreçler var ise onlara ait detaylar görüntülenebilir. İzleme ekranında karar verme mekanizması işletildiği için her bir adımın çalıştırılması ya da 'Success' olarak tamamlanması beklenmez. Örneğin 'Send Notification (Failed)' adımı sadece süreç hata aldığı anda tetiklenecek şekilde tasarlanmıştır.

## 5. BULGU VE TARTIŞMA

Manuel dağıtım ele alındığında insan hatası ve dağıtım süresi bakımından iki temel sorunla karşılaşılmaktadır. Manuel dağıtım işlemi her ne kadar işinde bu yetkinliğe sahip kişiler tarafından yapılıyor olsa da insanı etkileyen iş yoğunluğu, stres, duygusal faktörler ve çeşitli çevresel faktörler sebebiyle defalarca tekrarlanan işlerde bile insan hataları kaçınılmaz bir gerçektir. Manuel dağıtım süresi bakımından otomatik dağıtımlara göre göreceli olarak uzundur. Telekomünikasyon, banka, hastane gibi çok farklı sistem ve uygulamaların çevrimiçi bir arada çalıştığı bilişim altyapılarında manuel bir dağıtım ve güncelleme otomatik dağıtıma göre kıyaslanamayacak kadar uzun ve karmaşıktır. Manuel dağıtım, otomatik dağıtımdan çok daha fazla sorunlu ve kesintilere daha yatkındır. Otomatik sistemlerin yapabileceği görevleri tam olarak yapamazlar. Azami özen ve gayret gösterilse bile insan hatalarının olması kaçınılmazdır. Manuel dağıtımda işler yavaş ilerlediği ve hata yapmaya daha açık olduğundan dağıtım işi saatler ve hatta günlerce sürebilir. İlgili ekiplerin dağıtım süresi boyunca görevde kalması gerekir, bu da çalışanların ofis ortamında stres altında daha uzun zaman geçirmesi anlamına gelmektedir. Üstelik bu alanda yetkin bir uzmana sahip olmak da her zaman mümkün olmayabilir. Diğer taraftan bir uygulamanın dağıtımının uzun ve problemlili geçme ihtimalinin varlığı uygulama geliştiricilerinde kariyerini kısaltabilecek bir soruna neden olacağı endişesine neden olmaktadır. Ayrıca manuel dağıtım sırasında bir kesinti ya da hata gerçekleştiğinde, daha önceki başarılı olan bir sürüme geri dönmek son derece zor ve riskli bir taşımadır. Bu tür nedenler manuel dağıtım yetersiz kılmaktadır. Dağıtım sürecinin daha hızlı ve hatasız gerçekleştirilmesinde otomatik dağıtım araçlarının kullanımı bir çözüm sunmaktadır.

Otomatik dağıtım araçlarının kullanılması dağıtım sürecinde yetkin kişinin sürekli olarak bulundurulması ihtiyacını ortadan kaldıracaktır. Otomatik dağıtım süreci, tasarım kısmı hariç, bunların çalıştırılması için yetkin insan kaynağına ihtiyaç duymamaktadır. Böylece yetkin insan kaynağının kurum içerisinde daha verimli olarak değerlendirilmesi, kuruma sağladığı faydanın yanı sıra çalışan uzman kişinin çalışma motivasyonunu da arttıracaktır. Karmaşık gibi görünen dağıtım süreç tasarımları ile bu çalışmada olduğu gibi başta şablonlar oluşturulup sonraki kullanımlar basitleştirebilir. Özellikle sürekli tekrarlanan uygulama dağıtımlarında dağıtım sürecinin baştan bir defa tasarlandıktan sonra istenildiği kadar kullanılması imkânı vardır. Aynı zamanda bu kullanım modeli çok fazla sayıda uygulama barındıran bir kurumda merkezi yönetim ve daha hızlı süreç tasarımı ortaya çıkarma gibi yararlar da sağlamaktadır. Kurum kendi içerisinde şablonlar ve kullanım için eğitim materyalleri hazırlayarak işe yeni başlayan çalışanlar için sunulan oryantasyon sürecini kısaltabilir. Böylece bu iş için gerekli yetişmiş teknik insan kaynağına olan talebi de azaltabilir.

Güvenlik bakımından, bir sunucuya her bir erişim bir risk teşkil etmektedir. Erişim yetkisi olan kullanıcı her ne kadar bu yetkiye kontrollü olarak sahip olsa da bilerek ya da bilmeyerek uygulamaya zarar verebilir, iş sürekliliği kesintiye uğrayabilir. Otomatik dağıtım sürecinde bu yetkinlikte bir erişime ihtiyaç olmadığı için bu risk bütünüyle ortadan kalkmaktadır.

Telekomünikasyon, sigortacılık ve bankacılık gibi birçok sektörde yazılımların dağıtım ve güncellenmesine yönelik yasal düzenlemeler yanında COBIT, ITIL ve CMMI gibi kalite standart gereksinimlerine dair sıkı kurallar söz konusudur. Bağımsız otoriteler tarafından bu kurallara uyuma ilişkin kontrol ve denetimler gerçekleştirilmektedir. Bu kontrol ve denetimlerde sürece ilişkin detaylı veri ve iz kayıtlarına ihtiyaç duyulmaktadır. Bu isteklerin karşılanması otomatik dağıtım aracı kullanılarak gerçekleştirilen dağıtımlarda oldukça kolaydır. Dağıtıma ilişkin iz kayıtlarının sistem üzerinde uçtan uca görülebilir olması denetleyen taraf için de kolaylık ve güvence sağlamaktadır. Manuel gerçekleşen süreçler ise genellikle kimin neyi, ne zaman, nerede ve neden yaptığını açıklayan, merkezi belgelerden ve delillerden yoksundur. Bu süreçte herhangi bir kayıt yapısı bulunmamaktadır. Bu nedenle bazı kurumlarda yapılan aktivitenin kayıt altında tutulması için manuel olarak ekran görüntüleri alınıp saklanması istenmektedir. Bu durumda manuel olarak dağıtım yapan kişinin bu konu için de çalışması, zaman ayırması gerekir ve süreç daha da uzar. Diğer taraftan bir bankada iç denetim, bağımsız dış denetim, kalite denetimi gibi birçok kontrol ve denetim faaliyeti yürütülmektedir. Sık gerçekleşen denetimler bilgi teknolojileri tarafında ciddi iş yüküne sebep

olmaktadır. Bilgi teknolojileri yönetimleri otomatik dağıtım araçlarını kullanarak denetim için harcanan süreyi kısaltarak elindeki kaynakları daha verimli kullanabilme imkânına sahip olacaktır. Özetle otomatik dağıtım araçlarının kullanımı orta ve büyük ölçekli kurumsal firmalarda bilgi teknolojilerinin kontrol ve yönetimine katkı sunmaktadır.

## 6. SONUÇ

Bu çalışmada yazılım dağıtım süreci, manuel dağıtımdan kaynaklı sorunlar, bu sorunların kurum ve çalışanlara olan olumsuz etkileri, sorunların çözümünde kullanılan otomatik dağıtım araçları, otomatik dağıtım çözümünün sunduğu katkılar araştırılmıştır. Bu amaçla sorunun çözümünde kullanılan otomatik dağıtım araçları incelenmiş, yazılım dağıtım sürecinin otomatikleştirilmesine yönelik örnek bir WEB uygulaması ele alınarak IBM UrbanCode Deploy ürünü üzerinde uygulamalı olarak çalışılmıştır. Otomatik dağıtımda yer alan kavramlar açıklanmış, dağıtım tasarımının yapılabildiği ekranların kullanımına değinilmiş, hazırlanan şablonlar kullanılmıştır. Uygulamalı çalışmanın tüm sayfaları örnek ekran görüntüleriyle ortaya konulmaya çalışılmıştır. Benzer çalışma ve uygulamalar için öğretici olmasına özen gösterilmiştir.

Manuel dağıtım süreci oldukça hataya açıktır. Tanımlanmış bir süreçteki her bir işlem adımının her zaman aynı standartta insan eliyle yapılması çok zordur. Bir dağıtım sürecinin sürekli tekrarı, yazılı olması ve süreci yürüten kişilerin son derece yetkin olması bile manuel olarak yürütülen süreçlerde hata olasılığını ortadan kaldıramaz. Dağıtım sürecinin önemli bir adımı yanlışlıkla atlanabilir, sürüm sırasında meydana gelen bazı hatalar tespit edilemeyebilir, yanlış yazılım sürümleri devreye alınabilir, dağıtım yapılacak ortam ve uygulamalar yeterince analiz edilmeden dağıtıma başlanmış olabilir. Bu ve benzeri birçok nedenden dolayı hatalı, eksik ya da iş kesintisine sebep olan bir dağıtım pek çok sorun ve kayıplara neden olmaktadır. Ayrıca dağıtım işlemi yapan kişinin dağıtım yapılacak uygulamanın fonksiyonu hakkında yeterli bir bilgi sahibi olması beklenmez. Bu nedenle ciddi etkilere sahip bir uygulamanın dağıtımda bir sorun ile karşılaşma ihtimaline karşı yazılım geliştiricisi, proje yöneticisi, iş analisti ve ilgili diğer teknik personelin bir kriz masası etrafında bir arada çalışmaları gerekir. Uzun süren bir manuel dağıtım sürecinde çok sayıda uzman stres altında çalışır. Dağıtımda herhangi bir hata geliştiğinde iş kesintilerinin uzun sürme ihtimali de yüksektir. Çok fazla yazılım ürünü bulunan ve sürüm güncellemelerinin sıklıkla yapıldığı bir kurumda bu tür hata ve problemlerin yaşanma ihtimali daha da yüksektir. Diğer taraftan sıkı denetim ve kontrollere tabi olan kurumlarda dağıtım sürecinin işletilmesi yanında bunun kontrolü de oldukça zordur. Manuel dağıtım sürecinden kaynaklı bu ve benzeri birçok problemin çözümünde otomatik dağıtım araçlarının kullanımı bir çözüm sunmaktadır.

Günümüzde orta ve büyük ölçekli kurumların bilgi teknolojileri kuruluşlarında uygulama dağıtım süreçlerinin otomasyonu, yasal uyum yükümlülüğünün ötesinde sunduğu birçok imkân ve katkı nedeniyle önemli bir ihtiyaç haline gelmiştir. Bu çalışmada henüz otomatik dağıtım sürecini kullanmayan ve manuel işlemden dolayı birçok sorun yaşayan kurum veya kuruluşlar için yol gösterici öneriler bulunmaktadır. Manuel dağıtım sürecinden kaynaklanan birçok soruna karşı otomatik sürecin sunduğu çözüm ve katkılara dikkat çekilerek belli bir farkındalık oluşturulması hedeflenmiştir. Hali hazırda otomatik dağıtım ürünü kullanan kurumlar için ise dağıtım sürecinde tekrarlanacak standart operasyonları şablon haline getirmenin merkezi yönetim için sağladığı katkılara dikkat çekilmiştir. Tekrar eden yeni uygulamalarda bu yöntem kullanılarak dağıtım süreç tasarımlarının çok daha kısa sürede nasıl yapılabildiği anlatılmıştır. Yazılım yaşam döngüsünde önemli bir yere sahip olan yazılım uygulamalarının dağıtım süreci ve bu sürecin otomatikleştirilmesine yönelik uygulamalı çalışma literatüre kazandırılmıştır. İlgili araştırmacılara ve teknik kullanıcılara bir referans kaynağı olarak sunulmaktadır.

## KAYNAKÇA

- Ali, M., Aftab, A. & Buttt, W.H. (2020). "Automatic Release Notes Generation". 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), pp. 76-81, doi: 10.1109/ICSESS49938.2020.9237671.
- Ali, M. Tarar, M.I.N. & Butt, W.H. (2020). "Automatic Release Notes Generation: A Systematic Literature Review," 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1-5, doi: 10.1109/INMIC50486.2020.9318191.
- Arcangeli, J-P., Boujbel, R., & Leriche, S. (2015). "Automatic Deployment of Distributed Software Systems: Definitions and State of the Art". Journal of Systems and Software. Vol.103, pp. 198-218, <https://doi.org/10.1016/j.jss.2015.01.040>
- AWS Code Deploy (2022). <https://aws.amazon.com/tr/codedeploy/> (14.10.2022).
- Bamboo (2022). Continuous Delivery, from Code to Deployment, <https://www.atlassian.com/software/bamboo> (14.10.2022).
- Borandag, E. & Yücalar, F. (2020). "Artırılmış Gerçeklik ile Scrum Task Board Uygulaması". Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi, 4 (1) , 1-12. Doi:10.33461/uybisbbd.652366
- Capistrano (2022). What is Capistrano?, <https://capistranorb.com/documentation/overview/what-is-capistrano/> (14.10.2022).
- CircleCI (2022). Product Overview, <https://circleci.com/product/>, (14.10.2022).
- Chen, M., Zhang, S., Deng, H., Chen, B., Xing, C., & Xu, B. (2020). "Automatic deployment and control of network services in NFV environments". Journal of Network and Computer Applications. <https://doi.org/10.1016/j.jnca.2020.102677>
- CloudBees (2022). <https://www.cloudbees.com/>, (14.10.2022).
- Codar (2022). Deploy and redeploy packages, <https://docs.microfocus.com/Codar/1.80/Content/Concepts/devopsUseCase-DeployRedeploy.htm> (14.10.2022).
- DeployBot (2022). DeployBot Code Deployment Guides, <https://deploybot.com/> (14.10.2022).
- DergiPark (2022). DergiPark Akademik, <https://dergipark.org.tr/tr/> (20.03.2023).
- DevOps (2022). Azure DevOps Services , <https://learn.microsoft.com/tr-tr/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops>, (14.10.2022).
- Digital (2022). Deploy Description, <https://digital.ai/products/deploy/> (18.10.2022).
- Eloranta, M. (2018). "Continuous development and release automation of web applications", Master Thesis, School of Electrical Engineering, Aalto University.
- Farley, D., & Humble, J. (2010). "Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation", Addison-Wesley Professional, ISBN: 9780321670250
- GoCD (2022). GoCD Features, <https://www.gocd.org/why-gocd/> (16.11.2022).
- Gradle (2022). Gradle Build Tool, <https://gradle.org/> (16.11.2022).
- Highsmith, J. & Cockburn, A. (2001). "Agile Software Development: The Business of Innovation", in Computer, vol. 34, no. 9, pp. 120-127, IEEE, doi: 10.1109/2.947100.
- Hofmann, P., Samp, C., & Urbach, N. (2019). Robotic process automation. Electronic Markets. <https://doi.org/10.1007/s12525-019-00365-8>.



- IBM (2022). What is IBM® UrbanCode® Deploy?, <https://www.ibm.com/cloud/urbancode/deploy> (25.08.2022).
- IBM\_Urbancode (2022). Systems and topology overview, [https://www.ibm.com/docs/en/urbancode-deploy/7.0.5?topic=deploy-systems-topology-overview#ov\\_systems\\_\\_core](https://www.ibm.com/docs/en/urbancode-deploy/7.0.5?topic=deploy-systems-topology-overview#ov_systems__core) (12.08.2022).
- Jenkins (2022), What is Jenkins?, <https://www.jenkins.io/doc/> (16.09.2022).
- Lascu, T., Mauro, J., & Zavattaro, G. (2015). “Automatic Deployment of Component-Based Applications”. *Science of Computer Programming*. <https://doi.org/10.1016/j.scico.2015.07.006>
- Mohd Nordin, A.A., Latih, R. and Ali, N. M., “Software Development Productivity Model: Validation through Expert Review”. 2021 International Conference on Electrical Engineering and Informatics (ICEEI), Kuala Terengganu, Malaysia, 2021, pp. 1-6, doi: 10.1109/ICEEI52609.2021.9611151
- Morris, K. (2016). “Infrastructure as Code: Managing Servers in the Cloud”, O'Reilly Media.
- Octopus (2022), Deployment automation, <https://octopus.com/>, (16.09.2022).
- Pai, W. (2002). “A Quality-Enhancing Software Function Deployment Model”. *Information Systems Management*, 19:3, 20-24, doi: 10.1201/1078/43201.19.3.20020601/37166.3
- PiriKesif (2022). Piri Keşif Aracı, <https://kesifaraci.com/>, (20.03.2023)
- Poppendieck, M., & Poppendieck, T. (2007). “Implementing Lean Software Development: From Concept to Cash”, Addison-Wesley Professional, ISBN: 0321437381.
- Ruiz-Rube, I., ManuelDoderó, J., & Colomo-Palacios, R. (2015). A framework for software process deployment and evaluation. *Information and Software Technology*. <https://doi.org/10.1016/j.infsof.2014.12.001>
- Şahinaslan, E. (1998). Yazılımda Kalite Modellerinin Değerlendirilmesi, Bilgisayar Mühendisliği YL Tezi, Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü, Gebze Teknik Üniversitesi, İzmit, Türkiye
- TeamCity, (2022). Powerful continuous integration for DevOps-centric teams, <https://www.jetbrains.com/teamcity/> (16.10.2022).
- Travis (2022). About Us, <https://www.travis-ci.com/about-us/> (16.10.2022).
- UrbanCode(2022). Accelerate Software Delivery with IBM UrbanCode, <https://www.royalcyber.com/technologies/urbancode/> (20.10.2022).
- Zhai, H., & Wang, J. (2021). “Automatic deployment system of computer program application based on cloud computing”. *International Journal of System Assurance Engineering and Management*. <https://doi.org/10.1007/s13198-021-01068-0>
- Zhang, C., Sun, E., Cheng, X., Zang, D., & Chen, Z. (2021). A Scheme and Implementation of Automatic Deployment of Multilingual Industrial Mechanism Model Based on OpenStack. *2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*. <https://doi.org/10.1109/aeeeca52519.2021.9574124>

# Bulanık Çok Kriterli Karar Verme Yöntemleri Kullanılarak Karar Destek Sistemiyle Çalışan Ergonomik Kiosk Cihazı Tasarımı ve Yer Seçimi

\*\*\*

## Working with Decision Support System Ergonomic Kiosk Device Design and Location Selection Using Fuzzy Multi-Criteria Decision-Making Methods

Yusuf YAZAR<sup>1</sup> 

Buse BAYRAM<sup>2</sup> 

Şeyma Gül COŞKUN<sup>3</sup> 

Tamer EREN<sup>4</sup> 

DOI:10.33461/uybisbbd.1240463

### Öz

#### Makale Bilgileri

**Makale Türü:**  
Araştırma Makalesi

**Geliş Tarihi:**  
22.01.2023

**Kabul Tarihi:**  
30.05.2023

©2023 UYBISBBD  
Tüm hakları saklıdır.



Bilgi ve iletişim teknolojilerinin gelişmesiyle birlikte günümüzde bilgilendirmeler için teknolojik altyapılı elektronik cihazlar kullanılmaya başlanmıştır. Bu çalışmada üniversite öğrenci ve ziyaretçilerine hizmet etmek amacıyla karar destek sistemiyle çalışan bir kiosk cihazı tasarlanmıştır. Bu cihazın tasarlanma amacı mevcut yerleşim planının bilinmemesi, ulaşmak istenilen bilgilere ulaşamaması ve yönlendirici levhaların anlaşılabilmesi gibi problemlerin çözülmesidir. Kiosk cihazının buton yerleşimleri, işleyiş şekli, içerikleri düşünülmüş ve ihtiyaçları karşılayacak şekilde belirlenmiştir. Sistem işleyiş akış şeması üzerinde gösterilmiş, benzer sistemler tasarlanırken kullanılacak bir algoritma öne sürülmüştür. Kiosk cihazı tasarlanırken yazılım ergonomisi kriterleri kullanılmıştır. Cihaz maliyetlerini azaltmak ve kullanımı arttırmak için cihaz yer seçimi problemi de çalışmada incelenmiştir. Cihaz yer seçimi için Pisagor bulanık AHP ve TOPSIS yöntemleri kullanılmıştır. Yapılan araştırma birçok sektörde yönlendirmelerin kiosk cihazı aracılığıyla yapılabileceğini öngörmektedir. Mevcut problemin de tasarlanan karar destek sistemiyle çalışan kiosk cihazı ile çözüleceği düşünülmektedir.

**Anahtar Kelimeler:** Karar destek sistemi, kiosk cihazı tasarımı, Pisagor bulanık AHP ve TOPSIS, yazılım ergonomisi.

### Abstract

#### Article Info

**Paper Type:**  
Research Paper

**Received:**  
22.01.2023

**Accepted:**  
30.05.2023

©2023 UYBISBBD  
All rights reserved.



With the development of information and communication technologies, electronic devices with technological infrastructure have started to be used for information purposes. In this research, a kiosk device working with a decision support system was designed to serve university students and visitors. The purpose of the design of this device is to solve problems such as not knowing the current settlement plan, not being able to reach the desired information and not being able to understand the guiding signs. The button placements, functioning and contents of the kiosk device were considered and determined to meet the needs. The operation of the system is shown on the flow chart and an algorithm that can be used when designing similar systems has been proposed. While designing the kiosk device, software ergonomics criteria were used. In order to reduce device costs and increase usage, the device location problem was also investigated in the research. Pythagorean fuzzy AHP and TOPSIS methods were used for device location selection. Research show that in many sectors, routing can be done through the kiosk device. It is thought that the current problem will be solved with the kiosk device working with the designed decision support system.

**Keywords:** Decision support system, kiosk device design, Pythagorean fuzzy AHP and TOPSIS, software ergonomics.

**Atıf/ to Cite (APA):** Yazar, Y., Bayram, B., Coşkun, Ş. G. & Eren, T. (2023). Bulanık Çok Kriterli Karar Verme Yöntemleri Kullanılarak Karar Destek Sistemiyle Çalışan Ergonomik Kiosk Cihazı Tasarımı ve Yer Seçimi. Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi, 7(1), 68-84.

<sup>1</sup> Kırıkkale Üniversitesi, Mimarlık ve Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü yusuf.yazar@hotmail.com

<sup>2</sup> Kırıkkale Üniversitesi, Mimarlık ve Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü busebayram9@gmail.com

<sup>3</sup> Kırıkkale Üniversitesi, Mimarlık ve Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü seymagulcoskun3@gmail.com

<sup>4</sup> Prof. Dr. Kırıkkale Üniversitesi, Mimarlık ve Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü tamereren@gmail.com

## 1. GİRİŞ

Karar Destek Sistemleri veri, bilgi ve iletişim teknolojileri, belgeler vb. aracılığıyla ilgili karar vericilerin, sorunlarını çözüp kararlarını sağlıklı şekilde vermelerine yardımcı olan ve destek sağlayan sistemlerdir. Karar destek sistemleri ile çalışan kiosk cihazları; havalimanları, devlet daireleri içerisine konulan hizmet edeceği amaca göre kurgulanmış, vatandaşa ücretsiz hizmet eden dokunmatik ekranlara denir. Bir yazılımın kullanıcının çalışma performansını mümkün olduğu kadar olumlu etkilemesi için sahip olması gereken özellikler yazılım ergonomisi kavramı ile açıklanmaktadır (Başar ve Aslay, 2011).

Yazılım ergonomisi projelerin önce role sonra kişiye özel tasarlanmasını önermektedir (Pekcan ve Oğulata, 2008). Yazılım ergonomisine uygun olarak tasarımı tamamlanan kiosk cihazının konumlandırılması aşamasına geçilmelidir. Cihaz, hitap edeceği kişilere uygun ve ulaşılabilir konumda olmalıdır.

Üniversite yerleşim planının öğrenciler tarafından bilinmemesi, ulaşmak istediği yer ve kişilere erişimlerinde tabela ve yönlendiricilerin işlevini yerine getirememesi durumlarında levhaların etkisiz kalması, misafirlerin fakülte içi mekân arayışlarında zorlanmaları Kırıkkale Üniversitesi Mühendislik ve Mimarlık Fakültesi içerisine kiosk cihazı yerleştirilmesi düşüncesini ortaya çıkarmaktadır. Kiosk İngilizcede büfe anlamına gelen ve bilgilendirme amacıyla kullanılan teknoloji cihazıdır. Öğrencilerin fakülte içerisinde bölüm başkanlıkları, derslikler, konferans salonları, laboratuvarlar, ortak alanlar ve çalışma alanları gibi mevcut yerleşim alanlarını bulmakta zorlanması, yönlendirici tabelaların konum ve yön bilgisi verme durumundaki etkisizliği, mevcut yerleşimin karışık olması veya algılanması durumunda ulaşım zorlukları çekmesi ve bu nedenle gecikmelerin olması kaçınılmazdır. Ek olarak öğrencilerin yazılı afiş duyurularını görmemesi, üniversite internet sayfasının duyurular kısmını takip etmemesi, fakültede misafir olarak bulunanların kapalı veya açık alanlardan habersiz olması, yabancı öğrencilerin Türkçe bilgilendirmelerden faydalanamaması nedenleriyle fakülte içerisinde bilgilendirmelerin etkisinin az olduğu düşünülmektedir. Bu çalışmada tüm bu bahsedilen problemler göz önünde bulundurularak mevcut yerleşim düzenine bilgilendirmeler aracılığıyla daha hızlı uyum sağlamak ve gecikmeleri ortadan kaldırmak amacıyla mevcut yerleşim planını, bölüm ders ve sınav programlarını, derslik ve konferans salonlarının mevcut durumunu (boş, dolu, kullanılamaz vb.), ulaşmak istenen öğretim görevlisinin odasının konumu, özel gereksinimli kişilerin fakülte içerisinde kendilerine tahsis edilen alanların gösterilmesi, yabancı dil desteğiyle kullanılabilirliğin artırılması, öğrenci kimlik kartının okutulmasıyla kendi alanıyla ilgili bilgilere ulaşması vb. durumları içeren, yazılım ergonomisine uygun kiosk cihaz tasarımı ve Pisagor bulanık AHP ve TOPSIS yöntemleri kullanılarak yer seçimi üzerine çalışılmıştır.

Çalışmanın ikinci bölümünde cihaz yeri seçimi anlatılmıştır. Üçüncü bölümde Pisagor bulanık AHP, Pisagor bulanık TOPSIS ve yazılım ergonomisi yöntemlerinden bahsedilmiştir. Projede kullanılan yöntem ve sistemler üzerine yapılmış literatürdeki çalışmalara dördüncü bölümde yer verilmiştir. Beşinci bölümde uygulama örneği ayrıntılı olarak anlatılmıştır. Son bölüm olan altıncı bölümde ise, çalışmanın sonuçlarına ve literatüre katkısına yer verilmiş, yapılacak çalışmalara örnek teşkil edecek noktalara değinilmiştir.

Bu çalışmada zamandan tasarruf etmek ve bilginin etkin kullanılması amacıyla üniversitelerde kullanılabilecek karar destek sistemiyle çalışan kiosk cihazı tasarlanmıştır. Çalışmanın asıl amacı çoklu hizmet olmakla birlikte öğrencilerin dışında ziyaretçi ve engelli bireyleri de kullanıcı olarak dahil etmek, güncellenebilir bilgi ile yönlendirmelerin eksikliğini ortadan kaldırmak ve çok dilli bir yazılım ile kullanılabilirliğini artırmaktır. Ayrıca yazılım ergonomisi yönünden incelenilerek kullanımı etkin ve kolay hale getirmek amaçlanmıştır. Çalışmada sırasıyla problem tanımlanmış, yöntemler belirlenmiş, belirlenen yöntemler uygulanmış ve sonuç kısmında değerlendirilmiştir. Literatürde üniversitelerde kullanılabilecek karar destek sistemiyle çalışan kiosk cihazı veya benzer bir yönlendirme çalışmasına rastlanmamıştır. Ayriyeten kiosk cihazı tasarım çalışmalarının yazılım ergonomisi açısından değerlendirilmediği görülmüştür. Bu çalışma bahsi geçen özelliği neticesinde literatürde öncü olma niteliğindedir.

Cihaz seçimi problemleri teknolojinin gelişmesiyle birlikte birçok yöntemle çözülebilmektedir. Uygun ve etkili bir sonuca ulaşmak için karar vericinin belli yöntemlere başvurması gerekebilir. Teknolojinin insan yaşamına çeşitli şekillerde uyarlanması sayesinde, tasarım alanında yeni nesil ihtiyaçlar ve beklentiler ortaya çıkmıştır (Yıldırım, 2018). Kiosk cihazları fiş yazıcıları, dokunmatik panelleri, barkod okuyucuları, çelik klavyeleri, ödeme kabul ve ödeme iade cihazları, kesintisiz güç kaynağı (UPS) sistemleri, kart alıcı ve kart verici cihazları, telefon ahizesi ve otomatik aramalı telefon devrelerini içerisinde barındırabilmektedir (Keyosk, 2019). Uygulanacağı mekâna ve amaca bağlı olarak, yazılım ergonomisi göz önünde tutulduğunda çeşitli kriterler altında tasarımının yapılması mümkündür. Tasarımı yapılan kioskun konumlandırılması doğru şekilde yapılmalıdır. Kullanıcıların dikkatini çekebilecek ve onların kiosku kullanmasını tercih edebilecek bir yerde olmalıdır (Zedeli, 2021). Toplum yararına etki edebilecek etkileşimli bilgi ekranı da denilen kioskların yerleşiminde karar destek sistemleri kullanarak etkin sonuçlar elde edilebilir. Kiosk cihazları kullanım amaçlarına bağlı olarak çeşitli yerlerde kullanılabilir. Bu amaçlar; borç ödeme, bilet sistemi vb. satın alım işlemleri, kart dolularının yapılması, bilgi ve reklam içeriklerinin paylaşılması olabilir. Kuyruğu azaltma amaçlı devlet binalarında sıra veren kioskları da görmek mümkündür.

Literatürde benzer çalışmalar mevcuttur. Bilen (2019) yapmış olduğu yüksek lisans tezinde İzmir Tepecik Eğitim ve Araştırma Hastanesi Kan Alma Birimi sıra alma sisteminde meydana gelen kuyruğu azaltma amaçlı kiosk tasarımı yapmıştır. Özden (2022) yaptığı inceleme çalışmasında Anadolu Medeniyetleri Müzesi ile Frankfurt Tarih Müzesi'nin bilgilendirme tasarım uygulamalarını karşılaştırmalı olarak değerlendirmiştir. Maguire (1999) makalesinde kiosk sistemlerini kullanıcı tabanlı değerlendirmiştir. Rehğ vd. (1997) kioskların görüntü tabanlı insan algılaması yapması için bir tasarım örneği sunmuşlardır. Gülmez vd. (2008) yaptıkları çalışmada kioskların doğrudan pazarlama üzerindeki gelişim ve etkilerini incelemiştir. Cihan vd. (2016) yaptığı çalışmada AHP ve TOPSIS yöntemlerini kullanarak bir devlet hastanesi için ekokardiyografi cihazı seçimi üzerine çalışmışlardır. Christian ve Avery (1998) dijital akıllı kiosk projesiyle halkın kullanabileceği etkileşimli bir kiosk tasarımı geliştirmişlerdir. Porter vd. (2004) algoritmik mantık kullanarak acil departmanı için hasta odaklı işbirlikçi karar destek sağlayan astım kiosku tasarlamışlardır. Akçay (2009) hastaneler için etkileşimli elektronik bilgi ekranı (kiosk) tasarımı sayesinde hastaların hangi polikliniğe gideceklerine karar vermelerine yardımcı karar destek sistemi geliştirmiştir. Newman ve Salmon (2010) yer sincabı kontrol eğitimi ve sertifikasyonu için etkileşimli elektronik bilgi ekranı (kiosk) tasarımını geliştirmişlerdir. Bu sayede görevli birimlerin bilgilendirilmesi sağlanmıştır. Başar ve Aslay (2011) Atatürk Üniversitesi öğrenci bilgi sisteminin ergonomi kriterlerine uyup uymadığını incelemiştir. İki farklı anket çalışması yaparak yazılım kriterlerine göre değerlendirmişlerdir. Anketin güvenilirliğini SPSS programı yardımıyla analiz etmişlerdir. Salaki vd. (2015) bulanık mantık kullanarak Android tabanlı bir karar destek sistemi oluşturmuşlardır. Bu sistemle amaç lisede öğrencilerin yetenekli oldukları alanlara karar verilirken öğretmenlere yardımcı olmaktır. Bilen (2019) hastane kuyruklarında daha etkin bir kullanım için kiosk arayüz tasarımı ve gövde tasarımını uygulamıştır. Zedeli (2021) kent meydanlarında bulunan kiosk cihazının interaktif tasarımını uygulamıştır. Aksüt vd. (2020) yaptığı literatür taramasında ergonomik riskler ve sınıflandırılması konusuna yer vermiştir. Bu çalışmada, literatürdeki benzer çalışmalardan farklı olarak arayüz tasarımı yazılım ergonomisi kriterleri göz önünde bulundurularak tasarlanmıştır. Başar ve Aslay (2011)'in çalışmasına benzer olarak tasarım yazılım ergonomisi kriterlerine göre değerlendirilmiştir. Cihazın kullanım amacı da literatürden farklı olarak üniversitelerde bilgilendirme amaçlıdır. Cihaz yeri seçimi probleminde ise yöntem olarak Pisagor bulanık AHP ve TOPSIS yöntemleri kullanılmıştır. Karar destek sistemiyle çalışan kiosk cihazı tasarımıyla kullanıcılara tek bir noktadan çoklu hizmet sağlanabilecektir.

## 2. MATERYAL VE METOT

### 2.1. Bulanık Pisagor AHP

1965 yılında Zadeh tarafından ortaya atılan bulanık kümeler yöntemi ile bulanık mantığın, bulanık küme ve alt kümelerden oluştuğu, her nesnenin bir üyelik derecesinin olduğu açıklanmıştır.

Üye olmama dereceleri 1'den büyük olabileceği için Yager (2013) yılında Pisagor bulanık yöntemini geliştirmiştir. Geliştirilen bu yöntem çok kriterli karar verme yöntemleri ile kullanılmıştır. Analitik hiyerarşi süreci (AHP) bir ÇKKV yöntemidir, karar verici tarafından belirlenen kriterler alternatiflere göre ikili karşılaştırılır (Çavdur, vd. 2019). Pisagor bulanık kümeleri ile AHP birleştirilerek karşılaştırma oranlarının karşılığında bir değere ulaşılır. Buna bağlı olarak karar vericinin hızlı karar vermesine ve belirsizliğin ortadan kalkmasına olanak sağlar.

### 2.1.1. Pisagor bulanık AHP modeli

Bu bölümde Pisagor bulanık AHP'nin adımlarına yer verilmiştir (İlbar vd., 2018).

**Adım 1:** Uzman görüşlere göre ikili karşılaştırma matrisi  $R = (r_{ik})_{m \times m}$  oluşturulur. Saaty ölçeğine göre dilsel terimler pisagor bulanık sayılarla eşleştirilerek Pisagor bulanık AHP değerlendirme ölçeği oluşturulur.

**Adım 2:**

$$d_{ik_L} = \mu_{ik_L}^2 - v_{ik_U}^2 \quad (1)$$

$$d_{ik_U} = \mu_{ik_U}^2 - v_{ik_L}^2 \quad (2)$$

denklemleri kullanılarak üye ve üye olmayan fonksiyonların alt ve üst noktaları arasındaki farklar matrisi  $R = (r_{ik})_{m \times m}$  hesaplanır.

**Adım 3:**

$$s_{ik_L} = \sqrt{1000d_L} \quad (3)$$

$$s_{ik_U} = \sqrt{1000d_U} \quad (4)$$

eşitlikleri kullanılarak  $S = (S_{ik})_{m \times m}$  aralık çarpım matrisi hesaplanır.

**Adım 4:**

$$\tau_{ik} = 1 - (\mu_{ik_U}^2 - v_{ik_L}^2) - (v_{ik_U}^2 - v_{ik_L}^2) \quad (5)$$

kullanılarak  $\tau = (\tau_{ik})_{m \times m}$  belirlilik değerin matrisi oluşturulur.

**Adım 5:**

$$\tau_{ik} = \left( \frac{s_{ik_L} + s_{ik_U}}{2} \right) \tau_{ik} \quad (6)$$

eşitliği kullanılarak normalizasyondan önce ağırlık matrisini elde etmek için  $T = (t_{ik})_{m \times m}$ , aralık çarpım matrisi  $S = (S_{ik})_{m \times m}$  ile çarpılır.

**Adım 6:**

$$w_i = \frac{\sum_{k=1}^m t_{ik}}{\sum_{i=1}^m \sum_{k=1}^m t_{ik}} \quad (7)$$

kullanılarak normalize edilmiş ağırlıklar ( $w_i$ ) elde edilir.

### 2.1.2. Pisagor bulanık AHP uygulama alanları

Bu yöntem daha çok seçim ve kriter problemlerinde uygulanmıştır. Demiralay vd. (2022) yapmış oldukları makalelerinde Pisagor bulanık AHP yöntemini kullanarak Türkiye için uzay istasyonu seçimi uygulaması yapmışlardır. Erol vd. (2021) Pisagor bulanık AHP yöntemini kullanarak elektrik üretim santrallerinde iş güvenliği uzmanı seçiminde hibrit bir karar modeli uygulaması yapmışlardır. Oruç (2021) yapmış olduğu yüksek lisans tezinde Pisagor bulanık AHP yöntemiyle sıfır atık uygulaması, uygulamada karşılaşılan engelleri değerlendirmiştir.

## 2.2. Pisagor bulanık TOPSIS

Günümüzde en yaygın olarak kullanılan çok amaçlı karar verme yöntemlerinden biri olan TOPSIS'e literatürde ilk yer veren Hwang ve Yoon (1981), bu yöntemle seçilen alternatifin pozitif çözümden en kısa ve negatif çözümden en uzak mesafede olan ideal çözüme uyan alternatiflerini sıralayarak en iyi alternatifi bulmayı amaçlamışlardır.

### 2.2.1. Pisagor bulanık TOPSIS modeli

Bu bölümde Pisagor bulanık TOPSIS adımlarına yer verilmiştir (Yucesan ve Gul, 2019).

**Adım 1:** Başlangıçta  $R = (C_j(x_i))_{m \times n}$  karar matrisi oluşturulur.  $C_j (j = 1, 2, \dots, n)$  ve  $x_i (i = 1, 2, \dots, m)$  kriterlerin ve alternatiflerin değerlerine atıfta bulunmaktadır. Matris formu aşağıdaki gibidir:

$$R = (C_j(x_i))_{m \times n}$$

$$R = \begin{pmatrix} P(u_{11}, v_{11}) & \dots & P(u_{1n}, v_{1n}) \\ \vdots & \ddots & \vdots \\ P(u_{m1}, v_{m1}) & \dots & P(u_{mn}, v_{mn}) \end{pmatrix} \quad (8)$$

**Adım 2:** Pisagor bulanık pozitif ideal ve negatif ideal çözümlerin hesaplama işlemi aşağıdaki denklemler kullanılarak hesaplanmıştır.

$$x^+ = \{C_j, \max_i \{s(C_j(x_i))\} | j = 1, 2, \dots, n\}$$

$$x^+ = \{\langle C_1, P(u_1^+, v_1^+) \rangle, \langle C_2, P(u_2^+, v_2^+) \rangle, \dots, \langle C_n, P(u_n^+, v_n^+) \rangle\} \quad (9)$$

$$x^- = \{C_j, \min_i \{s(C_j(x_i))\} | j = 1, 2, \dots, n\}$$

$$x^- = \{\langle C_1, P(u_1^-, v_1^-) \rangle, \langle C_2, P(u_2^-, v_2^-) \rangle, \dots, \langle C_n, P(u_n^-, v_n^-) \rangle\} \quad (10)$$

**Adım 3:** Pozitif ideal ve negatif ideal çözümlere olan mesafeler aşağıdaki eşitlikler kullanılarak hesaplanmıştır.

$$D(x_i, x^+) = \sum_{j=1}^n w_j d(C_j(x_i), C_j(x^+))$$

$$D(x_i, x^+) = \frac{1}{2} \sum_{j=1}^n w_j \left( |(\mu_{ij})^2 - (\mu_j^+)^2| + |(v_{ij})^2 - (v_j^+)^2| + |(\pi_{ij})^2 - (\pi_j^+)^2| \right) \quad (11)$$

$$D(x_i, x^-) = \sum_{j=1}^n w_j d(C_j(x_i), C_j(x^-))$$

$$D(x_i, x^-) = \frac{1}{2} \sum_{j=1}^n w_j \left( |(\mu_{ij})^2 - (\mu_j^-)^2| + |(v_{ij})^2 - (v_j^-)^2| + |(\pi_{ij})^2 - (\pi_j^-)^2| \right) \quad (12)$$

$i = 1, 2, \dots, n$  pozitif ideal değerinin  $D(x_i, x^+)$  küçük olması; alternatif olan  $x_i$ 'nin en optimum sonucu vermesini, negatif ideal değerinin büyük olması;  $D_{min}(x_i, x^-)$   $x_i$ 'nin en optimum olduğunu gösterir. Bu şekilde gösterimi yapılabilir;  $D_{min}(x_i, x^+) = \min_{1 \leq i \leq m} D(x_i, x^+)$  ve  $D_{max}(x_i, x^-) = \max_{1 \leq i \leq m} D(x_i, x^-)$ .

**Adım 4:** Tüm alternatifler için göreceli yakınlık katsayısı hesaplanır.

$$\xi(x_i) = \frac{D(x_i, x^-)}{D_{max}(x_i, x^-)} - \frac{D(x_i, x^+)}{D_{min}(x_i, x^+)} \quad (13)$$

**Adım 5:** Son olarak alternatiflerin en iyi sıralaması belirlendi. Göreceli yakınlık endeksleri en yüksek olan alternatif en iyi alternatiftir.

### 2.2.2. Pisagor bulanık TOPSIS uygulama alanları

Bulut (2021) yapmış olduğu yüksek lisans tezinde COPRAS, bulanık COPRAS, Pisagor bulanık analitik hiyerarşi prosesi, Pisagor bulanık TOPSIS yöntemlerini kullanarak batarya enerji depolama sistemlerinin teknolojik performanslarının ve doğal gaz kombine çevrim santrallerine entegrasyonunun değerlendirmiştir. Gedikli ve Ervural (2022) yapmış oldukları makalede Pisagor

bulanık kümeler, Pisagor bulanık TOPSIS, bulanık COPRAS ve TOPSIS yöntemlerini kullanarak en uygun tersine lojistik hizmet sağlayıcısının bulanık grup karar verme yaklaşımı altında belirlenmesini araştırmışlardır.

### 2.3. Yazılım Ergonomisi

Ergonomi kavramı, Uluslararası Ergonomi Kurumu (IEA) tarafından şu şekilde tanımlanmıştır: “Ergonomi ya da İnsan Faktörleri Mühendisliği, insanın refahını, mutluluğu ve genel sistem performansını geliştirecek bilgi ve teoriyi bulmayı, uygun yöntemlerin uygulanmasını ve bir sistemin diğer elementler ve insanlar arasındaki etkileşimlerini temelde anlamaya çalışan bilimsel bir disiplindir” (IEA, 1997). Başka bir deyişle, Ergonomik Tasarım ya da Mühendisliği, etkili insan işlevleri ile güvenli ve konforlu çevreler, ürünler, sistemler, işler, görevler, makineler ve aletlerin tasarımı bilgisinin kullanılmasıdır (BCPE, 1999).

Yazılım ergonomisi, tasarımların belirli kriterler doğrultusunda değerlendirilerek tasarlanmasını gerektirmektedir. Akova (2006) yaptığı çalışmada Tablo 1’deki kriterler ile bir kontrol listesi oluşturmuştur.

**Tablo 1** Yazılım Ergonomisi Kriterleri Değerlendirme Tablosu

<b>Kriterler</b>	<b>Uygunluk</b>
Kullanıcı Karakteristiğine Uygunluk	
Görev Karakteristiğine Uygunluk	
Tasarımın Tanıdık Görünüş ve His Uyandırması	
Tutarlılık	
İnsan Faktörleri Bulgularının Kullanımı	
Gereksinimlere Uyumluluk	
Kabul Edilebilirlik	
Öğrenilebilirlik	
Kavranılabilirlik	
Üretkenlik	
Estetiklik	
Kullanılabilirlik	

Aydın ve Kurt (2002) yapmış olduğu çalışmada Tablo 1’deki kriterlerin aşağıdaki gibi açıklanabileceğinden bahsetmiştir.

- Kullanıcı Karakteristikleri:** Sisteme ara sıra veri girişi yapan kullanıcılar için görüntü üzerinde direktifler sağlanmalıdır.
- Görev Karakteristikleri:** Kullanıcı tarafından hata yönetimi yapılması gerekiyorsa diyaloglar yardımıyla kullanıcının görevi tamamlaması için anlatımlar bulunmalıdır.
- Tasarımın Tanıdık Görünüş ve His Uyandırması:** Mevcut sistemlere benzer sistem tasarlanarak kullanıcı becerilerinin transfer edilmesi sağlanır. Eğitim maliyeti en aza indirgenmiş olur.
- Tutarlılık:** Mevcut uygulama diğer uygulamaların bir bileşeni olma özelliğinde olursa daha çok fayda sağlanır.

- e. **İnsan Faktörleri Bulgularının Kullanımı:** Uygulama standartları insan faktörü göz önünde bulundurarak geliştirilmelidir.
- f. **Gereksinimlere Uyumluluk:** Tasarlanan yazılım alıcı veya bazı kanunların belirlenmiş standartlarına uygun olma özelliğini taşımaktadır.
- g. **Kabul Edilebilirlik:** Kullanıcı ara yüzü kullanılabilir olmalıdır.
- h. **Öğrenilebilirlik:** Szabo ve Kanuka (1998), iyi tasarlanmış bir grafiğin daha az zamanda daha etkili bir öğrenim sağladığını tespit etmişlerdir.
- i. **Kavranılabilirlik:** Tasarlanan sistemin diğer sistemlere benzerliği kullanıcıların işleyişi daha kolay kavrayabilmesini sağlamaktadır.
- j. **Üretkenlik:** Tasarımların hata yönetiminin en az olacak şekilde yapılması üretkenliği arttırmaktadır.
- k. **Estetiklik:** Grafik tasarımların sade ve düzenli olması estetik görünüm sağlamaktadır.
- l. **Kullanılabilirlik:** Pekcan ve Bihter (2007) yaptığı çalışmada yazılımların ana kullanılabilirlik standardını şu şekilde özetlemiştir;

“Yazılımların ana kullanılabilirlik standardı olan ISO Standardı 9241’in 11. Bölümü’ne göre “Kullanılabilirliğin ana hatlarıdır ve bu bölüm kullanılabilirliğin ana hatlarını şu şekilde belirtmektedir:

- i- Kullanım Genel Durumu (kullanıcılar, ekipman, çevre, amaçlar, görevler),
- ii- Kullanılabilirlik Ölçütleri (etkinlik, verimli çalışma, tatminiyet),
- iii- Tasarım sürecinde kullanılabilirliğin tanımlaması ve değerlendirilmesi.

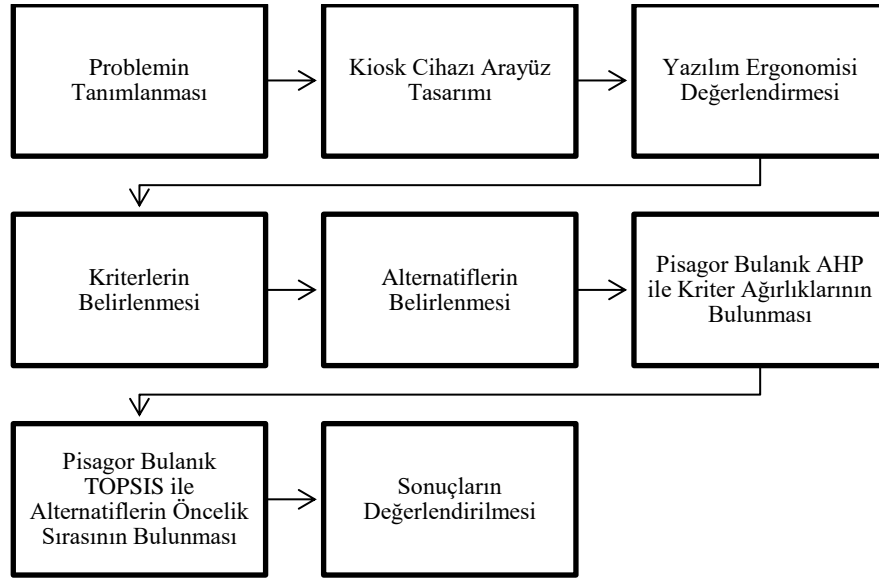
ISO 9241 bölüm 10’a göre, “Tasarım Prensipleri” yazılımın kullanılabilirliğini geliştirmek için kullanıcının ihtiyaçlarına tasarımcıyı yönlendirerek yedi genel tasarım prensibi ortaya koymaktadır:

- a) Göreve uygunluk,
- b) Kendi kendini tanımlayıcılık,
- c) Kontrol edilebilirlik,
- d) Kullanıcı beklentileri ile uyumluluk,
- e) Hata toleransı,
- f) Kişisel tercihlere uyumlandırabilirlik,
- g) Öğrenime uygunluk (s. 6).

### 3. UYGULAMA

Bu çalışmada karar destek sistemiyle çalışan kiosk cihazı tasarımı yapılmış, Pisagor bulanık AHP ve Pisagor bulanık TOPSIS yöntemleriyle Kırıkkale Üniversitesi Mühendislik ve Mimarlık Fakültesi için cihaz yeri seçimi üzerine çalışılmıştır. Çalışmada problemin tanımlanmasından sonra uzmanların görüşünden yararlanılarak kriterler belirlenmiş ve alternatifler seçilmiştir. Kriter ağırlıklarının bulunmasında Pisagor bulanık AHP yöntemi kullanılmış, alternatif öncelik sıralarının belirlenmesi aşamasında da Pisagor bulanık TOPSIS yöntemine başvurulmuştur. Şekil 1’de uygulama aşamaları verilmiştir.





Şekil 1 Uygulama Akış Şeması

### 3.1. Problemin Tanımlanması

Kırıkkale Üniversitesi Mühendislik ve Mimarlık fakültesi içerisinde bulunan tabelaların konum ve yer bilgisi verme görevini etkili şekilde yapamaması, mevcut yerleşimin karmaşık olması veya algılanması, öğrenci ve ziyaretçilerin ulaşmak istediği mekân ve kişilere erişmekte zorlanmaları, bunun sonucunda da gecikmelerin yaşanması kaçınılmazdır. Bu sorunların ortadan kaldırılması, mevcut yerleşim düzenine uyum sağlanması, bilgilendirmeler aracılığıyla daha hızlı uyum sağlanıp ve gecikmelerin ortadan kaldırılması amacıyla bir bilgi sistemi tasarlanması, tasarlanan bilgi sisteminin yer seçiminin yapılması ele alınmıştır.

### 3.2. Kiosk Cihazı Arayüz Tasarımı

Tasarlanan kiosk cihazı arayüzü 4 temel ekrandan oluşmaktadır. İlk olarak ana ekran kullanıcıları karşılamaktadır. Şekil 2’de görüleceği üzere orta üst kısımda üniversite logosu, sağ üst kısımda tarih ve saat, sol orta kısımda duyurular bölümü, orta kısımda öğrenci, ziyaretçi ve engelsiz ulaşım butonları bulunmaktadır. Sağ orta kısımda ise QR kod okut butonu bulunmaktadır. Sağ altta bulunan buton aracılığıyla kullanıcılar sistem dilini İngilizce olarak seçebilmektedir.



Şekil 2 Kiosk Ana Ekranı

Ana ekrandan öğrenci butonuna basıldığında Şekil 3’te görüleceği üzere “Lütfen öğrenci numaranızı giriniz...” bilgisi ekranın orta kısmında yönlendirme amacıyla çıkarak yazının altında bulunan alana öğrenci numarası girilmesi gerekmektedir. Bu sırada sol alt kısımda 30 saniye hareketsizlik sayacı aktif olmaktadır. Sayacın amacı yanlış tıklama veya işlemden vazgeçme durumlarında sistemin başlangıç durumuna dönmesini sağlamaktır. Sayaç (ana ekran hariç), dil seçimi, tarih ve logo sabit olarak tüm ekranlarda bulunmaktadır.



Şekil 3 Öğrenci Seçimi Ekranı

Öğrenci numarası girildikten sonra Şekil 4'te görüleceği üzere öğrenci bilgilendirme butonları aktif olmaktadır. Öğrenci burada derslikler, konferans salonları, çalışma alanları, laboratuvarlar, bölüm başkanlıkları, ortak alanlar, dekanlık gibi yerlere nasıl gidilebileceğini öğrenebilmektedir. Ayrıca bölüm ders programı, bölüm sınav takvimi ve bölüm duyurularına da erişebilmektedir. Sol üst kısımda öğrenci isim bilgisi, öğrenci numarası ve öğrenci bölümü bilgi amaçlı çıkmaktadır. Farklı seçimler yapabilmek için geri dön butonu ve ana menü butonları da ek olarak ekrana eklenmiştir.



Şekil 4 Öğrenci Bilgi Ekranı

Ana ekrandan ziyaretçi butonuna basıldığı takdirde Şekil 5'te görüleceği üzere ziyaretçi bilgilendirme butonları aktif olmaktadır. Ziyaretçiler konferans salonları, sınav salonları, bölüm başkanlıkları ve ortak alanlara nasıl gidileceği bilgisini alabilmektedir. Ayrıca mevcut etkinlikleri görüntüleyebilir, etkinlik hakkında detaylı bilgi alabilir, etkinlik alanına ulaşım yönlendirmesini alabilir, geliştirildiği takdirde etkinliklere kayıt işlemini de buradan yapabilir.



Şekil 5 Ziyaretçi Bilgi Ekranı

Ana ekrandan engelsiz ulaşım butonuna basıldığı takdirde Şekil 6'da görüleceği üzere fakülte içerisinde engelsiz ulaşım sağlayabileceği yönlendirmeleri içeren fakülte yerleşim planı

gösterilecektir. Şekil 6’da kullanılan engelsiz ulaşım yerleşim planı örnektir ve gerçeği yansıtmamaktadır. Sadece sistemin anlaşılması için örnek amacıyla kullanılmıştır.



Şekil 6 Engelsiz Ulaşım Bilgilendirme Ekranı

Ana ekrandan QR kod okut butonuna basıldığı takdirde Şekil 7’de görüleceği üzere “Lütfen QR kodunuzu cihazın alt bölümünde bulunan kameraya gösteriniz.” Uyarısı çıkmaktadır. Ekranın orta alanında kamera ön izlemesi bulunmaktadır. Kameraya QR kod okutulduktan sonra eğer okuma doğru bir şekilde yapıldıysa ön izleme ekranının altında “QR kod okuma başarılı...” şeklinde yeşil renkli uyarı yazısı çıkacaktır. Okutma başarılı olmadığı durumda “QR kod okuma başarısız, tekrar deneyiniz...” şeklinde kırmızı renkli uyarı yazısı çıkacaktır.



Şekil 7 QR Kod Okuma Ekranı

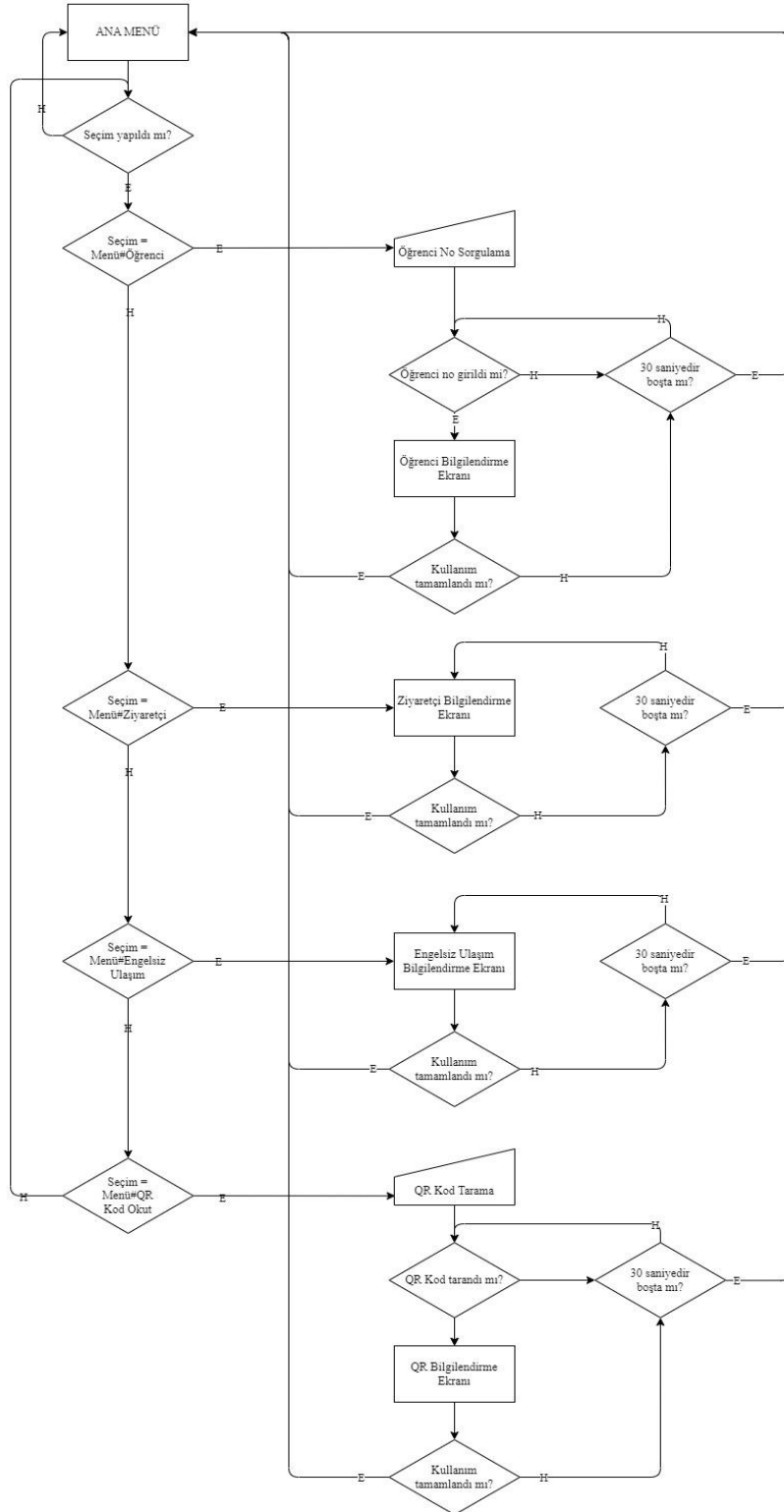
QR kod okutulduktan sonra QR kod içerisinde yerleştirilmiş yönlendirme kodu aracılığıyla ilgili sayfaya yönlendirme işlemi cihaz tarafından otomatik olarak yapılacaktır. Ayrıca sınav bilgisi gibi gömülü bilgiler okutulması durumunda da Şekil 8’de görüleceği üzere sınav hakkında bilgileri ve sınav salonuna nasıl gidileceği bilgisi ekranda gösterilecektir.



Şekil 8 QR Kod Okuma Sonuç Bilgileri

### 3.3. Kiosk Cihazı Akış Şeması

Kiosk cihazının genel yapısı 1 ana menü, 4 alt menü ve menülere ait bilgilendirme ekranından oluşmaktadır. Şekil 9’da görüleceği üzere karar destek sistemiyle çalışan kiosk cihazının akış şeması menü, alt menü, bilgilendirme ekranı, sorgu faaliyetleri, hareketsizlik kontrolü ve işlem bitirme kontrolünden oluşmaktadır.



Şekil 9 Kiosk Cihazı Akış Şeması

### 3.4. Yazılım Ergonomisi Değerlendirmesi

3.2. Kiosk Cihazı Arayüz Tasarımı başlığında görülen arayüz ekran görüntüleri 2.3. Yazılım Ergonomisi bölümünde verilen kriterler kullanılarak tasarımın yazılım ergonomisine uygunluğu Tablo 3’te incelenmiştir.

**Tablo 2** Tasarlanan Kiosk Cihazının Yazılım Ergonomisi İncelemesi

Kriterler	Uygunluk
Kullanıcı Karakteristiğine Uygunluk	Uygundur.
Görev Karakteristiğine Uygunluk	Uygundur.
Tasarımın Tanıdık Görünüş ve His Uyandırması	Uygundur.
Tutarlılık	Geliştirilebilir.
İnsan Faktörleri Bulgularının Kullanımı	Uygundur.
Gereksinimlere Uyumluluk	Uygundur.
Kabul Edilebilirlik	Uygundur.
Öğrenilebilirlik	Uygundur.
Kavranılabilirlik	Uygundur.
Üretkenlik	Uygundur.
Estetiklik	Uygundur.
Kullanılabilirlik	Uygundur.

Tablo 2’de uygunluk sütununda verilen cevaplar 2.3. Yazılım Ergonomisi bölümünde verilen kriterlerin karşılanması durumunda uygundur cevabı verilmiş, mevcut sistem dahilinde geliştirilerek karşılanması durumunda da geliştirilebilir cevabı verilmiştir. Kriterlerin değerlendirilmesi sonuçlarının doğruluğu amacıyla Türkiye’de kullanılan 5 adet kiosk cihazı ana ekranı da çalışmaya eklenmiştir. Şekil 10’da görüleceği üzere yıllardır aktif olarak kullanılan kiosk cihazları arayüzleriyle proje kapsamında geliştirilen arayüz ekranları yakın benzerlik göstermektedir.



Şekil 10 Kiosk Cihazı Örnekleri

### 3.5. Cihaz Yeri Seçimi

#### 3.5.1. Kriterlerin belirlenmesi

İncelemeler sonucunda kriterler Tablo 3’teki gibi karşılaştırılmıştır. 2 endüstri mühendisi ve 1 akademisyenden oluşan uzmanların görüşünün alınması, teknolojik aletlerin yüksek ısıya maruz

kalması sonucunda oluşabilecek zararların da göz önünde bulundurulması neticesinde bahsi geçen kriterlere karar verilmiştir.

**Tablo 3** Kriterler ve Açıklamalar

Kriter Numarası	Kriterler	Açıklamalar
K1	Uzaklık	Cihazdan yapılan yönlendirme gecikmeye sebep olmamalıdır. Yönlendirilecek alanlara en yakın noktaya yerleştirilmesi gerekmektedir.
K2	Yoğunluk	Cihaz en yoğun girişe sahip kapı yakınına yerleştirilerek kullanımı artırılmalıdır. Bu nedenle en yoğun girişin seçilmesi gerekmektedir.
K3	Erişilebilirlik	Cihaz ziyaretçi, öğrenci ve engelli bireylerin ulaşabileceği bir noktada olmalıdır. Engelli bireyler için rampa vb. bulunmalıdır.
K4	Kullanım Alanı	Cihaz kullanım alanı geniş olmalı, cihazın kullanımı sırasında alanı meşgul etmemelidir. Ayriyeten sıra oluşma durumlarında karmaşaya sebep olmamalıdır. Bu nedenle geniş kullanım alanı olan nokta seçilmelidir.
K5	Güvenlik	Cihaz kötüye kullanılmamalı, cihaza fiziksel zarar verilmemelidir. Bu nedenle en güvenli noktaya yerleştirilmelidir.
K6	Merkezi Yerleşim	Cihaz tüm kullanıcı tipleri için ana girişe sahip veya yakın noktaya yerleştirilmelidir. Bu sayede 2 bloktan da erişim sağlanabilir.
K7	Dış Faktörler	Cihaz elektronik bir görüntü paneline sahip olduğu için güneş, sıcaklık vb. etkenlerden uzak olmalıdır. Bu nedenle dış faktörlerden etkilenmeyeceği bir noktaya yerleştirilmelidir.
K8	Fark edilebilirlik	Cihazın kullanılabilmesi için fark edilmesi gerekmektedir. Bu nedenle cihazın etrafında kapatıcı büyüklükte eşyalar veya yapılar bulunmamalıdır.

### 3.5.2. Alternatiflerin Belirlenmesi

**Tablo 4** Alternatifler

Alternatifler	
A1	A Blok Ana Giriş
A2	A Blok Orta Giriş
A3	A Blok Üst Giriş
A4	A Blok Kantin Girişi
A5	B Blok Girişi

Fakülte yerleşimi; erişilebilirlik, uzaklık, yoğunluk, kullanım alanı, güvenlik, merkezi yerleşim, dış faktörler, fark edilebilirlik kriterleri açısından değerlendirilmiştir. 1 numaralı alternatif A blok ana girişini, 2 numaralı alternatif A blok orta girişini, 3 numaralı alternatif A blok üst girişini, 4 numaralı alternatif A bloktaki kantin girişini, 5. alternatif ise B blok girişini temsil etmektedir. Alternatifler belirlenen kriterler göz önünde bulundurularak değerlendirilmiştir. Alternatifler Tablo 4'te gösterilmiştir.

## 4. BULGULAR

#### 4.1. Kriterlerinin Ağırlıklandırılması

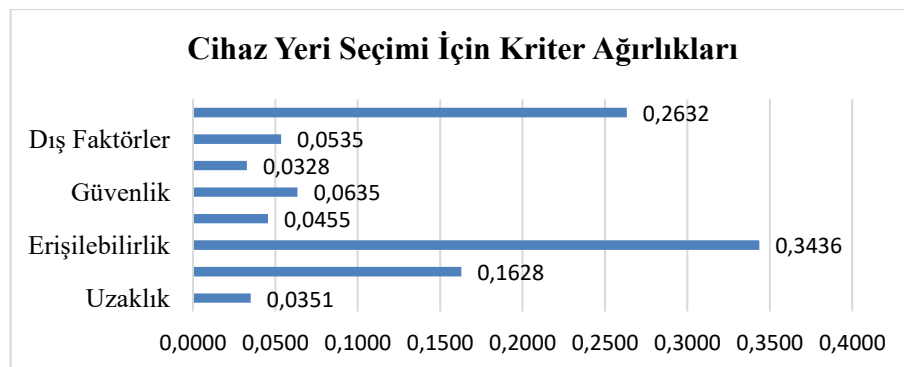
**Tablo 5** Pisagor Bulanık AHP Ağırlık Ölçeği

Dilsel Değişken	Aralıklı Pisagor Bulanık Sayılar			
	$\mu_L$	$\mu_U$	$\nu_L$	$\nu_U$
Kesinlikle Düşük Önemli (KD)	0	0	0,9	1
Çok Düşük Önemli (ÇD)	0,1	0,2	0,8	0,9
Düşük Önemli (D)	0,2	0,35	0,65	0,8
Ortalamanın Altında Önemli (OA)	0,35	0,45	0,55	0,65
Eşit (E)	0,1965	0,1965	0,1965	0,1965
Ortalama Önemli (O)	0,45	0,55	0,45	0,55
Ortalamanın Üstünde Önemli (OÜ)	0,55	0,65	0,35	0,45
Yüksek Önemli (Y)	0,65	0,8	0,2	0,35
Çok Yüksek Önemli (ÇY)	0,8	0,9	0,1	0,2
Kesinlikle Yüksek Önemli (KY)	0,9	1	0	0

8 adet kriterin ağırlıklandırılmasında Pisagor bulanık AHP yöntemi kullanılmıştır. Kriterlerin ikili karşılaştırılması dilsel değişkenler aracılığıyla yapılmış daha sonra bulanık sayılar ile ölçeklendirilmiştir. Ölçeklendirilmede Tablo 5'ten yararlanılmıştır. Değerlendirmelerin geometrik ortalaması karar matrisini vermiştir. Farklar matrisi oluşturulup tereddüt dereceleri hesaplanmıştır. Normalize edilmemiş ağırlıklara ulaşıp en sonunda Tablo 6'daki kriter ağırlıkları elde edilmiştir.

**Tablo 6** Kriter Ağırlıkları

Kriterler		$w_i$	Sıralama
<b>K1</b>	Uzaklık	0,0351	<b>7</b>
<b>K2</b>	Yoğunluk	0,1628	<b>3</b>
<b>K3</b>	Erişilebilirlik	0,3436	<b>1</b>
<b>K4</b>	Kullanım Alanı	0,0455	<b>6</b>
<b>K5</b>	Güvenlik	0,0635	<b>4</b>
<b>K6</b>	Merkezi Yerleşim	0,0328	<b>8</b>
<b>K7</b>	Dış Faktörler	0,0535	<b>5</b>
<b>K8</b>	Farkedilebilirlik	0,2632	<b>2</b>



**Şekil 11** Kriter Ağırlıkları Grafiği



Pisagor bulanık AHP yönteminin uygulanması neticesinde kriter ağırlıkları belirlenmiş olup 0,3436 ile en yüksek ağırlığa sahip kriter erişilebilirlik olduğu Şekil 11’de görülmektedir. Akabinde 0,2632 ile takip eden fark edilebilirlik kriterinin çıkması da erişilebilirliğe yakın bir kriter olduğundan normaldir. Cihazın erişilebilir ve fark edilebilir olması kullanımı açısından oldukça önem teşkil etmektedir. Değerlendirmeler sonucunda değerlerin tutarlı olduğu görülmüştür.

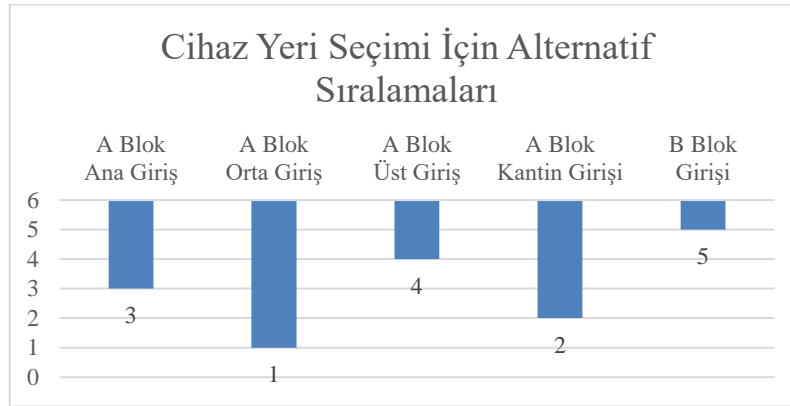
#### 4.2. Alternatiflerin Öncelik Sırasının Bulunması

Tasarlanan kiosk cihazının Kırıkkale Üniversitesi Mühendislik ve Mimarlık Fakültesi’nde konumlandırılması için 5 adet alternatif olduğu kanısına varılmıştır. Bu 5 adet alternatiflerin öncelik sıralamasının bulunabilmesi için Pisagor bulanık TOPSIS’ ten yararlanılmıştır. Alternatiflerin kriterlere göre karar matrisi geometrik ortalama alınarak oluşturulmuş olup Tablo 7’de gösterilmektedir.

**Tablo 7** Karar Matrisi

R	K1			K2			K3			K4			K5			K6			K7			K8		
	w <sub>1</sub>		0,035	w <sub>2</sub>		0,163	w <sub>3</sub>		0,344	w <sub>4</sub>		0,046	w <sub>5</sub>		0,063	w <sub>6</sub>		0,033	w <sub>7</sub>		0,053	w <sub>8</sub>		0,263
	u	v	π	u	v	π	u	v	π	u	v	π	u	v	π	u	v	π	u	v	π	u	v	π
A1	0,40	0,87	0,29	0,80	0,44	0,41	0,50	0,80	0,33	0,40	0,87	0,29	0,25	0,92	0,30	0,50	0,80	0,33	0,80	0,44	0,41	0,60	0,71	0,37
A2	0,80	0,44	0,41	0,50	0,80	0,33	0,80	0,44	0,41	0,10	0,00	0,99	0,10	0,00	0,99	0,80	0,44	0,41	0,70	0,60	0,39	0,80	0,44	0,41
A3	0,10	0,97	0,22	0,40	0,87	0,29	0,50	0,80	0,33	0,40	0,87	0,29	0,25	0,92	0,30	0,50	0,80	0,33	0,60	0,71	0,37	0,80	0,44	0,41
A4	0,10	0,99	0,10	0,60	0,71	0,37	0,80	0,44	0,41	0,70	0,60	0,39	0,40	0,87	0,29	0,70	0,60	0,39	0,80	0,44	0,41	0,50	0,80	0,33
A5	0,10	0,97	0,22	0,10	0,97	0,22	0,50	0,80	0,33	0,70	0,60	0,39	0,25	0,92	0,30	0,40	0,87	0,29	0,70	0,60	0,39	0,25	0,92	0,30

Ağırlıklı standart matrisi oluşturulup ideal ve negatif ideal çözümlerin oluşturulmasıyla yöntemin adımları uygulanmaya devam edilmiştir. İdeal çözüme göreli yakınlıklar belirlenip Şekil 12’deki alternatif sıralamaları elde edilmiştir.



**Şekil 12** Alternatif Sıralamaları

Kiosk cihazının fakülte içinde A blok orta girişinde konumlandırılmasına karar verilmiştir. Bu alternatif, erişilebilirlik, yoğunluk, uzunluk vb. saydığımız 8 adet kriter altında değerlendirilmiş ve en uygun bölge olduğu kararlaştırılmıştır.

#### 5. SONUÇ VE TARTIŞMA

Bu çalışmada Kırıkkale Üniversitesi Mühendislik ve Mimarlık Fakültesi öğrenci ve ziyaretçilerinin zamandan tasarruf etmesi, doğru bilgilendirilmesi ve ihtiyaçlarının karşılanması amacıyla karar destek sistemiyle çalışan kiosk cihazı tasarlanmıştır. Yer seçimi problemiyle Pisagor bulanık AHP ve Pisagor bulanık TOPSIS yöntemleri kullanılarak cihazın Kırıkkale Üniversitesi Mühendislik ve Mimarlık Fakültesi A Blok orta girişe yerleştirilmesine karar verilmiştir. Kiosk cihazı tasarlanırken yazılım ergonomisi kriterleri göz önünde bulundurulmuş ve bu kriterlere göre arayüz tasarımının değerlendirilmesi yapılmıştır. Değerlendirme sonrası yıllardır Türkiye’de kullanılan kiosk



cihazları ile kıyaslanarak arayüzün benzer bir tasarıma sahip olduğu görülmüştür. Kiosk cihazının kullanıcıları öğrenci, ziyaretçi ve engelli bireyler olarak düşünülmüştür. Kullanıcı tiplerinin ihtiyaçları doğrultusunda menüler oluşturulmuştur. Bu menülere kullanıcı tiplerine göre direkt veya öğrenci numarası ile erişim sağlanabilmektedir. Ayrıca QR kod teknolojisinin aktif kullanımından faydalanmak için tasarlanan kiosk cihazında QR kod okutma bölümüne de yer verilmiştir. QR kod okutarak hem zamandan tasarruf hem de bilginin etkin kullanımı amaçlanmıştır. Tasarlanan karar destek sistemiyle çalışan kiosk cihazı, güncelliğini yitirmiş yönlendirme levhalarının, belirli bir gruba hitap eden veya tarihi geçmiş duyuruların ve yeri kullanıcılar tarafından bilinmediği için kullanılmayan alanların (ortak alan, çalışma salonları, dinlenme alanları, laboratuvar gibi) etkisizliğini ortadan kaldıracak niteliktedir. Türkiye’de daha önce üniversitelerde yönlendirme amacıyla tasarlanmış bir kiosk cihazına rastlanamamıştır. Bu çalışma tüm üniversitelerde bilgi ve iletişim teknolojilerinin aktif kullanılması ve yönlendirmelerin kiosk cihazları aracılığıyla yapılması için örnek teşkil etmektedir. Ayrıca çalışma süresince tasarlanan sistemin birçok sektöre uygulanabileceği de görülmüştür. Literatürde kiosk cihazı tasarım çalışmalarının yazılım ergonomisi kriterlerine göre değerlendirilmediği görülmüştür. Bu çalışma literatürde yazılım ergonomisinin daha fazla kullanılmasına da katkı sağlayacaktır. Projede kullanılan yöntem, işleyiş ve düşünce yapısı yapılacak olan benzer çalışmalara örnek teşkil edecektir.

## KAYNAKÇA

- Akçay O., (2009). “Hastaneler için etkileşimli elektronik bilgi ekranı (Kiosk)”, Doctoral Dissertation DEÜ Sağlık Bilimleri Enstitüsü,
- Akova A., (2000). “Bilişim Toplumunda Bilişsel Ergonomi ve Önemi”, Kara Harp Okulu Bilgi Toplama ve Yayın Merkezi Yayınları,
- Aksüt G., Eren T., Tüfekçi M., (2020). “Ergonomik Risk Faktörlerinin Sınıflandırılması: Bir Literatür Taraması”, Ergonomi, 3(3), 169-192,
- Başar M. S., Aslay F., (2011). “Yazılım Ergonomisi: Atatürk Üniversitesi Öğrenci Bilgi Sisteminin Ergonomisinin İncelenmesi”, Journal of Graduate School of Social Sciences, 15(1), 25-42,
- BCPE, (1999). “Candidate Handbook: Policies, Practices&Procedures”, Board of,
- Bilen E., (2019). ”Design of the kiosk for more effective utilization in hospital queues”, Doctoral dissertation İzmir Institute of Technology Turkey,
- Bulut M., (2021). “Batarya enerji depolama sistemlerinin teknolojik performanslarının ve doğal gaz kombine çevrim santrallerine entegrasyonunun değerlendirilmesi”, Master's thesis Fen Bilimleri Enstitüsü/Endüstri Mühendisliği Ana Bilim Dalı,
- Christian A. D., Avery B. L., (1998). “Digital smart kiosk Project”, In Proceedings of the SIGCHI conference on Human factors in computing systems, 155-162,
- Çalık A., (2022). “Bulanık AHP-Bulanık ARAS Yöntemlerine Dayalı Dayanıklı Tedarikçi Seçimi”, İstanbul Gelişim Üniversitesi Sosyal Bilimler Dergisi, 9(2), 275-296,
- Çavdur, F., Sebatlı, A., & Küçük, M. K., (2019), Öğrenci-proje takımı oluşturma problemi için grup-karar verme ve hedef programlama temelli çözüm yaklaşımı. Journal of the Faculty of Engineering and Architecture of Gazi University, 34(1), 505-521.
- Demiralay E., Çopur E. H., & Paksoy T., (2022). “Spaceport Selection Using a Novel Hybrid Pythagorean Fuzzy AHP & TOPSIS Based Methodology: A Case Study of Turkey” Journal of Aeronautics & Space Technologies/Havacılık ve Uzay Teknolojileri Dergisi, 15(1), 1-17,
- Eren T., Cihan Ş., Ayan E., Topal T., Yıldırım E.K., (2017). “Çok ölçütlü karar verme yöntemleri ile ekokardiyografi cihazı seçiminin yapılması”, Sağlık Bilimleri ve Meslekleri Dergisi, 4(1), 41-49,

- Erol E., Özcan E., & Eren T., (2021). “Elektrik üretim santrallerinde iş güvenliği uzmanı seçiminde hibrit bir karar modeli” *Journal of Turkish Operations Management*, 5(1), 615-629,
- Gülmez M., Koçkaya A. F., (2008). “Doğrudan pazarlama aracı olarak kioskların tüketiciler tarafından kullanım etkinliği üzerine bir pilot çalışma”, *Verimlilik Dergisi*, (2), 157-175,
- IEA., (1997). “Facts and Background”, Zurich: International Ergonomics Association,
- İlbahar, E., Karaşan, A., Cebi, S., & Kahraman, C., (2018). “A novel approach to risk assessment for occupational health and safety using Pythagorean fuzzy AHP & fuzzy inference system”, *Safety science*, 103, 124-136,
- Maguire, M. C., (1999). “A review of user-interface design guidelines for public information kiosk systems”. *International journal of human-computer studies*, 50(3), 263-286,
- Newman P., Salmon T. P., (2010). “Ground Squirrel Control Training and Certification using an Interactive Computer Kiosk System”, *Proceedings of the Vertebrate Pest Conference*, 24, 24,
- Oruç, K. Ş., (2021). “Sıfır Atık Uygulaması, Uygulamada Karşılaşılan Engellerin Pisagor Bulanık ahp Yöntemi ile Değerlendirilmesi”, Doctoral dissertation, Ankara Yıldırım Beyazıt Üniversitesi Fen Bilimleri Enstitüsü,
- Özden K., (2022). “Müzelerde Etkileşimli Bilgilendirme Tasarımı: Anadolu Medeniyetleri Müzesi ile Frankfurt Tarih Müzesi’nin Karşılaştırılması”, *ARTS: Artuklu Sanat ve Beşeri Bilimler Dergisi*, (7), 151-178,
- Pekcan, B., (2007). “Yazılım Ergonomisi ve Bir İşletme Yazılımı Üzerine Uygulanması, Yüksek Lisans Tezi”, Çukurova Üniversitesi Fen Bilimleri Enstitüsü,
- Porter S. C., Cai Z., Gribbons W., Goldmann D. A., Kohane I. S., (2004). “The Asthma Kiosk: A Patient-centered Technology for Collaborative Decision Support in the Emergency Department” *Journal of the American Medical Informatics Association*, 11(6), 458–467,
- Rehg, J. M., Loughlin, M., & Waters, K., 1997. “Vision for a smart kiosk. In *Proceedings of IEEE Computer Society Conference*” on Computer Vision and Pattern Recognition, 690-696,
- Salaki R. J., Kawet R., Manoppo C., Tumimomor F., (2015). “Decision Support Systems Major Selection Vocational High School in Using Fuzzy Logic Android-Based” *International Conference on Electrical Engineering, Informatics, and Its Education*,
- Szabo M., Kanuka H., (1998). “Effects of Violating Screen Design Principles of Balance, Unity and Focus on Recall Learning, Study Time, and Completion Rates”. *ED-Media/ ED-Telecom 98 Conference Proceedings*, Association for the Advancement of Computing in Education, Charlottesville, VA, 8(1), 23-42,
- Yager, R. R., (2013). Pythagorean membership grades in multicriteria decision making. *IEEE Transactions on Fuzzy Systems*, 22(4), 958-965,
- Yıldırım, Y., (2018). “Exploring the significance of collaboration in the early design stages of interactive kiosks”,
- Yucesan, M., & Gul, M., (2020). Hospital service quality evaluation: an integrated model based on Pythagorean fuzzy AHP and fuzzy TOPSIS. *Soft Computing*, 24(5), 3237-3255,
- Zadeh L. A., (1965). “Fuzzy Sets”, *Information And Control*, 8, 338-353,
- Zedeli A. R., (2021). “Günümüzde kent meydanlarında bir etkileşim yöntemi olarak kiosk: İnteraktif tasarım uygulaması”, Doctoral dissertation Yıldız Teknik Üniversitesi Sosyal Bilimler Enstitüsü,