

Bölüm : 8	Sayfa : 15
Yapılan İş : Haftalık Çalışmanın Raporlanması	Tarih : 10/04/2023

PDF formatındaki 1000 adet ticari faturaya ait bilgileri ayıklayan botun tasarımına başlandı. Proje için önceden hazırlanmış faturalar kullanıldı[3].

Invoice Number, Date, Company Name, Total Amount değerleri çekilecek, çekilen bilgiler excel dosyasına raporlanacaktır.

İşlenemeyen bir fatura olursa botun hata vermesi yerine süreci yönetmesi sağlanacaktır. Raporla ise neden işlenemediğine dair exception bilgisi yer alacaktır.

Ek olarak işlenemeyen ve işlenen faturalar, Output klasörü içine ayrı olarak taşınacaktır.

	A	B	C	D	E	F	
1	Invoice Number	Invoice Date	Customer Name	Total Amount	File Name	Status	Error Message

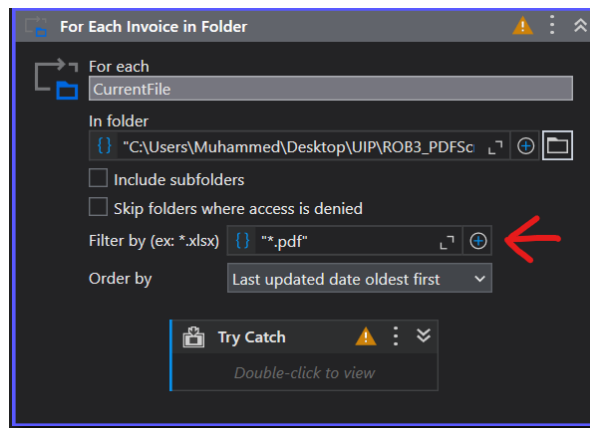
Şekil 30. Şablon Dosyasındaki Header'lar

Şablon olarak bu bilgileri içeren excel dosyasının ilk satırı aynen kalacak, 2. ve sonraki satırlara fatura bilgileri yazılacaktır.

İlk olarak bir veri tablosu oluşturulmuştur. Ayıklanan veriler veri tablosuna alınacak, sonrasında da satır olarak excel'e aktarılabilmesi için bu tablo kullanılacaktır.

Şekil 31. Build Data Table

Ardından, bir döngü yapısı ile bütün faturaları işlemek için For Each File in Folder aktivitesi kullanılmıştır. Girdi olarak faturaların bulunduğu klasör gösterilmiş, filtre olarak ise "*.pdf" ifadesi kullanılmıştır. Bu sayede sadece pdf dosyaları bot tarafından işlenilmeye çalışılacaktır.



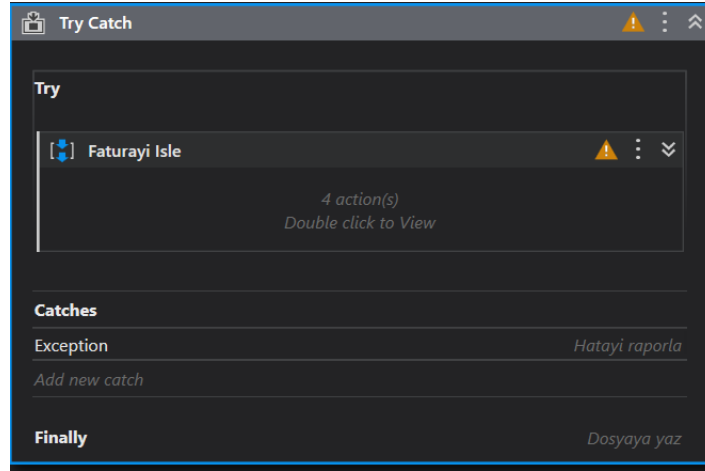
Şekil 32. For Each File

Kontrol Eden:

Bölüm : 8	Sayfa : 16
Yapılan İş : Haftalık Çalışmanın Raporlanması	Tarih : 11/04/2023

Her dosya için botun exception'a düşme riski bulunmaktadır. İstenilen bilgi çekilemediğinde, dosya açılmadığında vs. gibi durumlarda botun çökmemesi için bu hataların handle edilmesi gereklidir.

Try Catch yapısı kullanılarak bu işlem gerçekleştirilir.



Şekil 33. Try Catch

Try kısmında fatura pdf'sinden bilgilerin veri tablosuna çekilmesi için gerekli işlemler yer almaktadır. Bir hata oluşursa, işlem kesilir ve Catch bloğunun içindeki akış gerçekleştirilir. Finally kısmında ise, işlem başarılı veya başarısız farketmeksizin çalışması gereken işlemler yer alır.

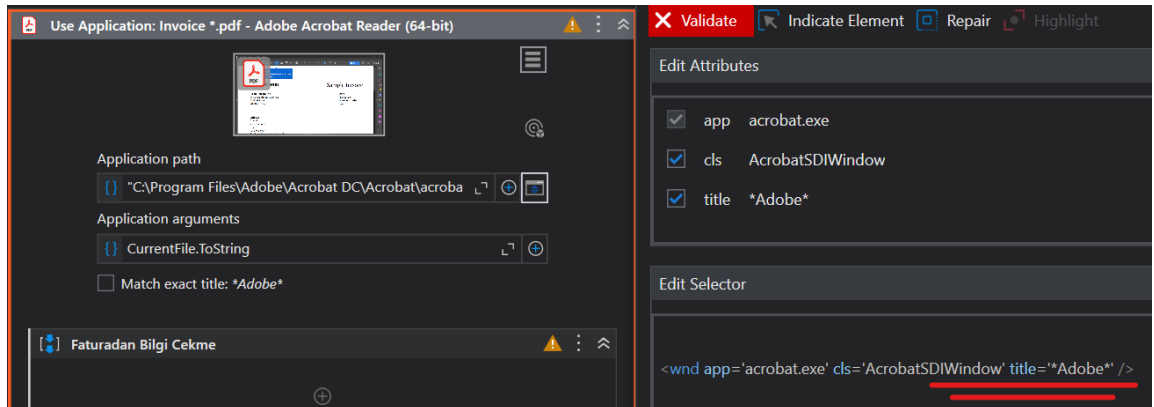
İlk olarak Fatura İşle adlı akış oluşturulmuştur. Bu kısımda gerçekleştirilen işlemler açıklanacaktır. Başlangıç olarak, işlenmek üzere seçilmiş olan dosyanın bilgisi Log aktivitesi ile konsola yazdırılır.

("İsleniyor: " + CurrentFile.Name.ToString ifadesi, sadece dosyanın ismini yazdıracaktır.

CurrentFile.ToString ifadesi ise dosyanın yolunu ve ismini beraber yazdırır.)

Ardından, Use Application aktivitesi oluşturulur ve uygulama olarak Adobe Reader seçilir.

Açılacak dosya argümanına ise CurrentFile.ToString parametresi verilir.



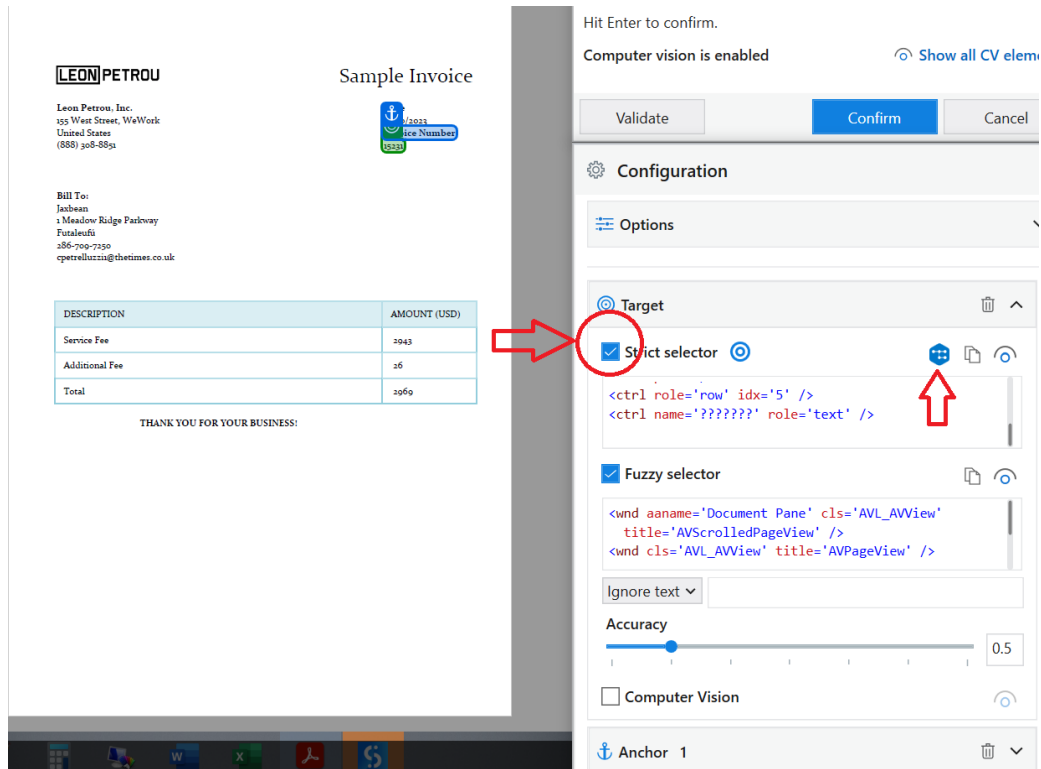
Şekil 34. App Selector

Selector kısmındaki title, açılan her dosyada değişmektedir. Açılan her pencerenin bot tarafından düzgünce algılanabilmesi için title kısmı “*Adobe*” olarak genelleştirilmiştir. Aktivitenin ayarlarından pencere maksimize edilmiş ve input mode olarak simulate seçilmiştir.

Artık, bot tarafından ilgili fatura açılmıştır. Verileri çekmeye başlamadan önce, faturanın ekrana tam olarak sığması için zoom ayarı yapılabilir. OCR ile tanımlama yapılıyorsa bu işlem gereklidir. Bunun için Keyboard Shortcuts aktivitesinde Ctrl + Y kısayolu kullanılır. Ardından Type Into aktivitesi ile istenen oran girilir. Örneğin, “45.9”. Ardından Enter tuşuna basılır.

Keyboard Shortcuts aktivitesi ayarlarında, input mode olarak Window Message (Ctrl + Y için) ve Hardware Events (Enter tuşu için) kullanılmıştır. İlk tercih olarak Simulate denenmiş, ancak istenen sonuç alınamadığı için Simulate>Chromium API>Window Message>Hardware öncelik sırası göze alınarak diğer modlar denenmiştir. Adobe için Chromium API modu çalışmayacağından diğer modlar kullanılmıştır.

Zoom ayarı yapıldıktan sonra, Get Text aktivitesi ile invoiceNumber alanındaki bilgi alınmıştır. Bunun için indicate target opsiyonuna tıkladıktan sonra target olarak fatura numarası, anchor olarak “Invoice Number” alanı seçilir. Fuzzy selector tamamlanır.



Şekil 35. Indicate Target

Fuzzy Selector'un yanında, botun hızının ve kararlılığının artması için Strict Selector ve Strict Anchor seçimi yapılmıştır.

Kontrol Eden:

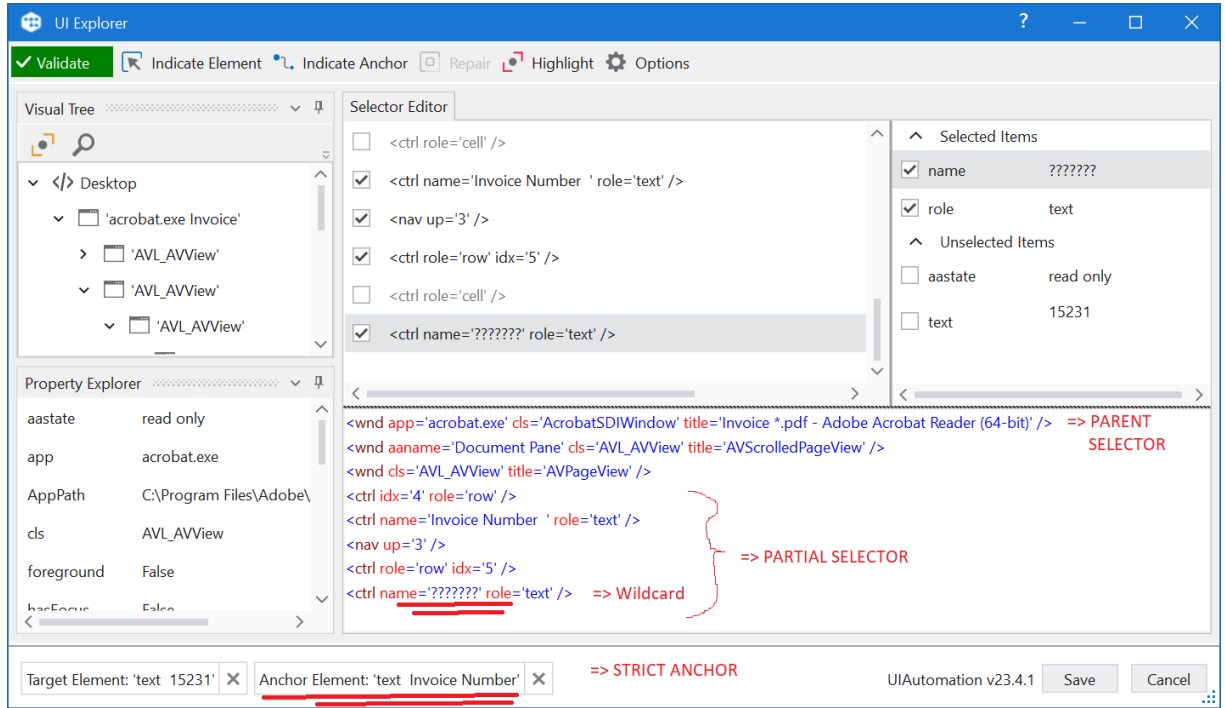
Bölüm : 8	Sayfa : 17
Yapılan İş : Haftalık Çalışmanın Raporlanması	Tarih : 12/04/2023

Strict Selector oluşturmak için UI Explorer açılır (bkz. Şekil 36). Öncelikle Parent Selector kısmında “Invoice 2.pdf” gibi yer alan sabit isim, “Invoice *.pdf” haline getirilerek genelleştirilir. Böylece açılan her fatura penceresi tanımlanabilecektir. Ardından invoice number’ı tanımlayan Partial Selector oluşturulacaktır.

Strict Anchor oluşturmak için indicate anchor opsiyonu seçilir ve fatura üzerinden “invoice number” alanı işaretlenir.

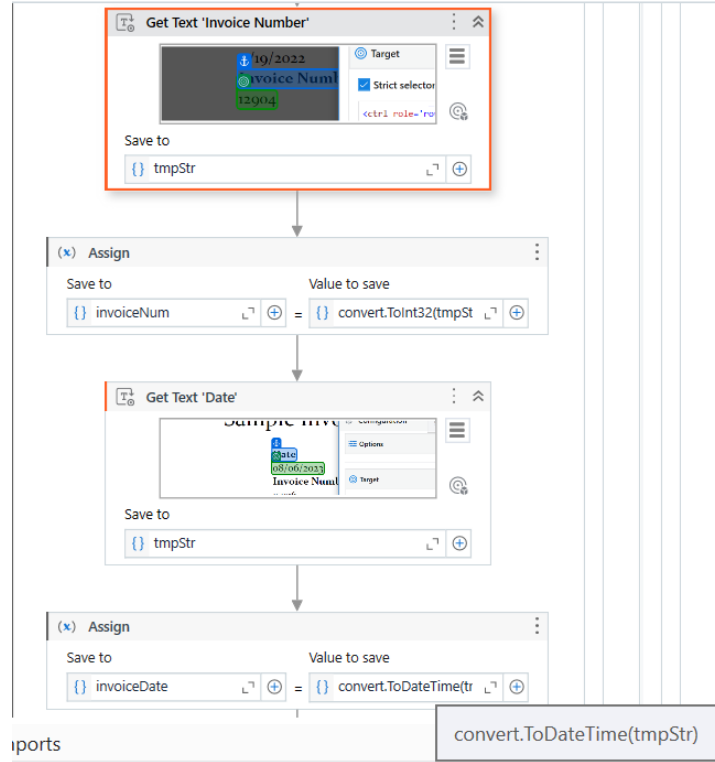
UiPath’ın kendi oluşturduğu Strict Selector yeterli değildir. Örneğin, 15231 fatura numarası olmak üzere, otomatik oluşturulan strict selector şu şekildedir: <ctrl name='15231 ' role='text' />

Bu ifade, yalnızca ilgili fatura için tanımlama yapabilir. Farklı numaralı faturalarda başarısız olacaktır. Bunu düzeltmek için wildcard ifadeleri ile genelleştirme yapılmıştır. “?” karakteri sadece bir karakterin yerine geçmektedir. <ctrl name='???????' role='text' /> olarak seçilen strict selector bütün fatura numaralarını tanımlayabilir.



Şekil 36. UI Explorer Arayüzü

Değişiklikler kaydedildikten sonra string olarak alınan veri int değişkenine çevrilir. Bu çevirme işlemi diğer kolonlar için de kendi tiplerine göre düzenlenerek aynen uygulanır. Örneğin invoiceDate ifadesi Date tipine, invoiceTotal ifadesi Double tipine çevrilir. Şekil 37’yi inceleyiniz.



Şekil 37. Explicit Conversion

invoiceDate, invoiceCompanyName, invoiceTotal alanları için de benzer şekilde strict selector, strict anchor ve fuzzy selector & anchor seçimi yapılır. Tek fark, strict seletor için kullanılan wildcardların içeriğidir:

invoiceDate için <ctrl name='*//*/*' role='text' />,

invoiceCompanyName için <ctrl name='*' role='text' />,

invoiceTotal için <ctrl name='*' role='text' /> ifadeleri, ilgili alanları tanımlar.

Son olarak, ayıklanan bilgiler konsola loglanır.

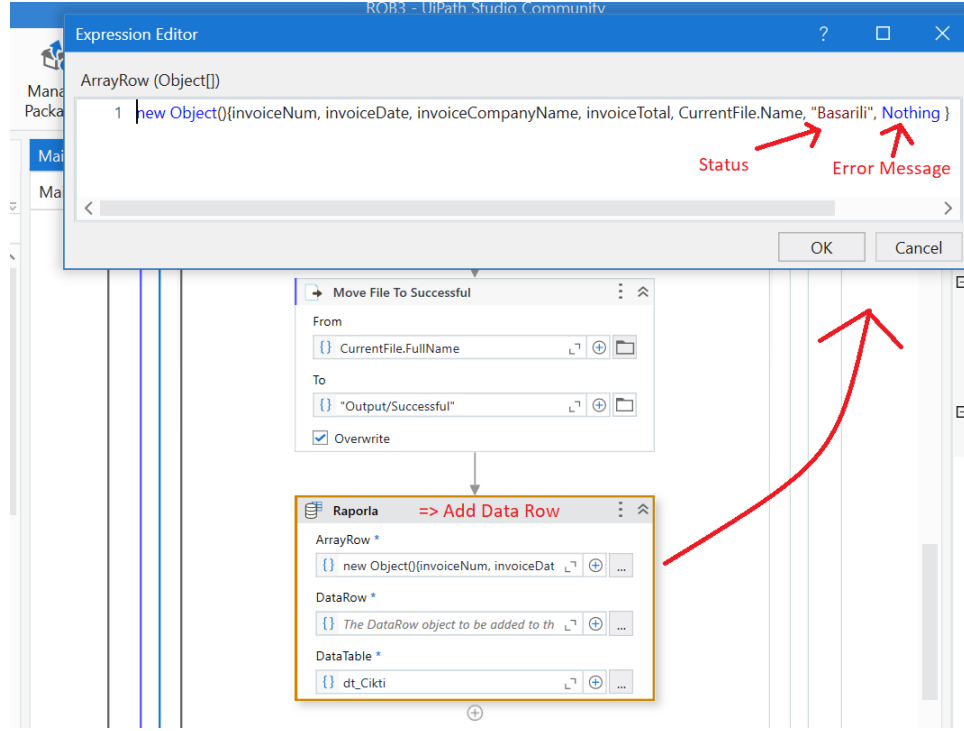
```
1 Isyeri Adı: " + invoiceCompanyName + Environment.NewLine + "Fatura no: " + invoiceNum.ToString + Environment.NewLine + "Tarih: " + invoiceDate.ToString("dd/MM/yyyy") + Environment.NewLine +  
2 Total: " + invoiceTotal.ToString
```

Şekil 38. Logging

Kontrol Eden:

Bölüm : 8	Sayfa : 18
Yapılan İş : Haftalık Çalışmanın Raporlanması	Tarih : 13/04/2023

Use Application aktivitesi içinde yapılacaklar bitmiştir. Bu, fatura dosyasının başarıyla işlendiği anlamına gelir. Bu durumda ilgili pdf dosyasının Output/Successful lokasyonuna taşınması ve bilgilerinin data tablosuna eklenmesi gereklidir. Şekil 39’u inceleyiniz.



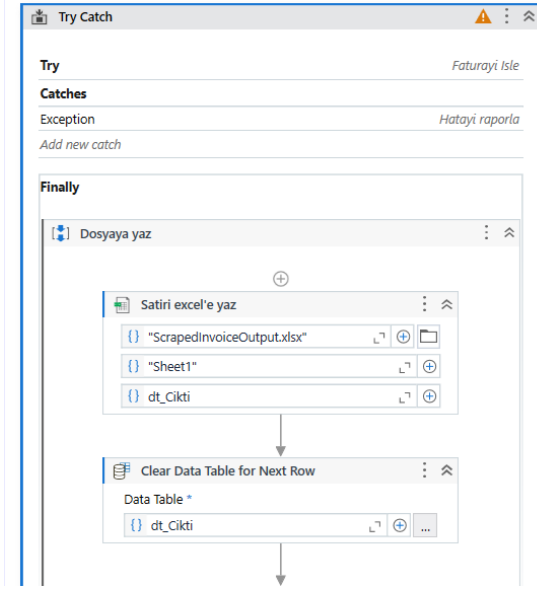
Şekil 39. Başarılı Ayıklama Sonrası

Oluşturulan ArrayRow objesi veri tablosuna eklenir. Try Catch bloğunun Try kısmı bitmiştir. Hata alınmadığı için Finally bölümüne geçilir. Bu kısım hata alınsa da alınmasa da her türlü çalışacaktır.

Şekil 40’da görülebileceği üzere, dt_Cikti tablosu bir satır veri içermektedir. Bu veri tablosu excel dosyasına yazdırıldığında, bir faturaya ait bilgi satır olarak excel’e yazılmış olur. Son olarak veri tablosunun içeriği temizlenir, böylece döngü başa döndüğünde bir sonraki dosyadan ayıklanacak veriler yeniden boş tabloya yazılır ve excel’e aktarılır.

Eğer bu sıfırlama yapılmazsa, excel satırları aşağıdaki örnekteki gibi tekrarlanan ifadeler içerecektir.

Invoice 1 ...
Invoice 1 ...
Invoice 2 ...
Invoice 1 ...
Invoice 2 ...
Invoice 3 ...
...



Şekil 40. Dosyaya Yaz

Clear Data Table aktivitesinin arkasından Invoke Code aktivitesi içerisinde “GC.Collect()” kodu çalıştırılabilir. Bu ifade, newObject() ile oluşturulan ve artık kullanılmayan arrayRow değişkeninin ve diğer değişkenlerin belleği şişirmesini engellemektedir. Bu projede ilgili ifade kullanılmadan yapılan çalıştırmalarda belirli bir iterasyondan sonra botun yanıt vermediği görülmüştür.

İlgili kod, her döngü ardından çalıştırılmış ve gereksiz değişkenleri boşalttığından bu sorun çözülmüş ve tek seferde 1000 dosya başarı ile işlenmiştir.

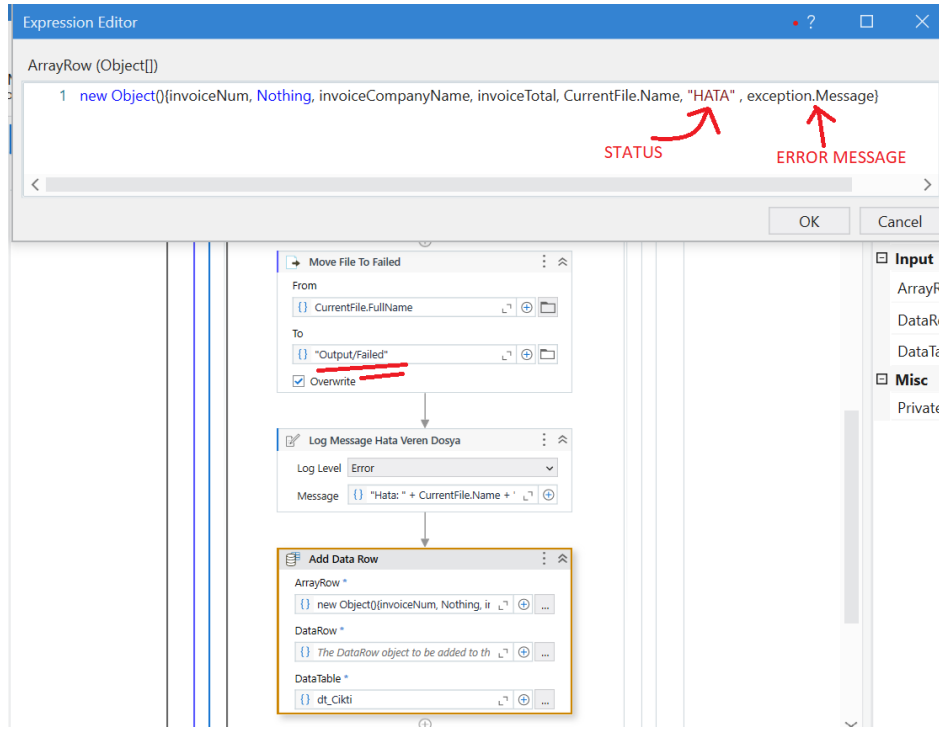
16232	07/06/23	Baopieopia	4062 Invoice 985.pdf	Basarili	
15415	28/12/23	Kazu	5535 Invoice 986.pdf	Basarili	
12316	16/10/23	Thoughtstorm	1478 Invoice 987.pdf	Basarili	
18138	22/02/23	Vimbo	4256 Invoice 988.pdf	Basarili	
10541	31/03/23	Fadeo	5372 Invoice 989.pdf	Basarili	
13981	21/06/23	Yambee	1005 Invoice 990.pdf	Basarili	
10122	01/11/23	Jabbersphere	9716 Invoice 991.pdf	Basarili	
15692	04/11/22	Aimbo	1080 Invoice 992.pdf	Basarili	
19910	06/07/23	Devbug	9866 Invoice 993.pdf	Basarili	
14285	21/12/23	Skidoo	3705 Invoice 994.pdf	Basarili	
16174	23/12/23	Fivechat	7545 Invoice 995.pdf	Basarili	
19558	14/03/23	Skibox	1124 Invoice 996.pdf	Basarili	
11564	02/01/23	Skimia	7100 Invoice 997.pdf	Basarili	
12489	05/03/23	Avavee	3712 Invoice 998.pdf	Basarili	
11491	28/11/22	Midel	8208 Invoice 999.pdf	Basarili	
10130	07/03/23	Gabtune	7956 Invoice 1000.pdf	Basarili	

Şekil 41. Excel Raporu

Bir sonraki sayfada, Try Catch bloğunun Try kısmında bir hata ile karşılaşılması durumunda düşülen Catch kısmında, yapılan işlemler incelenecektir.

Kontrol Eden:

Try bloğunda herhangi bir sebepten dolayı hata alınırsa Catch bloğuna atlanır. Hata veren dosya Output/Failed lokasyonuna taşınır. Konsola ismi yazdırılır. Yeni bir arrayRow objesi oluşturulur ve hata mesajı bu objeye eklenir. Sonrasında ise bu obje veri tablosuna aktarılır. Catch bloğu bittikten sonra Finally bloğuna geçilir ve ilgili veri tablosu excel'e yazılır.



Şekil 42. Hatayı Yönet

Hata mesajı örneği için Şekil 43'ü inceleyiniz. Invoice 2.pdf penceresi açıldığında kullanıcı tarafından kapatılmış ve bot bu dosyayı işleyememiştir. Bu nedenle hatayı handle edip sonraki dosyaya geçmiştir.

Invoice Number	Invoice Date	Customer Name	Total Amount	File Name	Status	Error Message
19070	28/12/23	Leenti	2210	Invoice 1.pdf	Basarili	
0			0	Invoice 2.pdf	HATA	Could not find the user-interface (UI) element for this action. Possible solutions: • Ensure application is opened and the UI element is visible on the screen at execution time • Edit the Target of the UI activity and use Validation to debug the issue. • If needed, re-indicate the element as its properties might have changed • Use "Check state" activity to check the application state before executing the action • Increase the "Delay before" value to allow time to the application to render entirely and become responsive
11898	13/07/23	Rhybox	2173	Invoice 3.pdf	Basarili	

Şekil 43. Hata Örneği

Kontrol Eden: