Serco Work Experience

22/07/24 - 16/08/24

Executive Summary

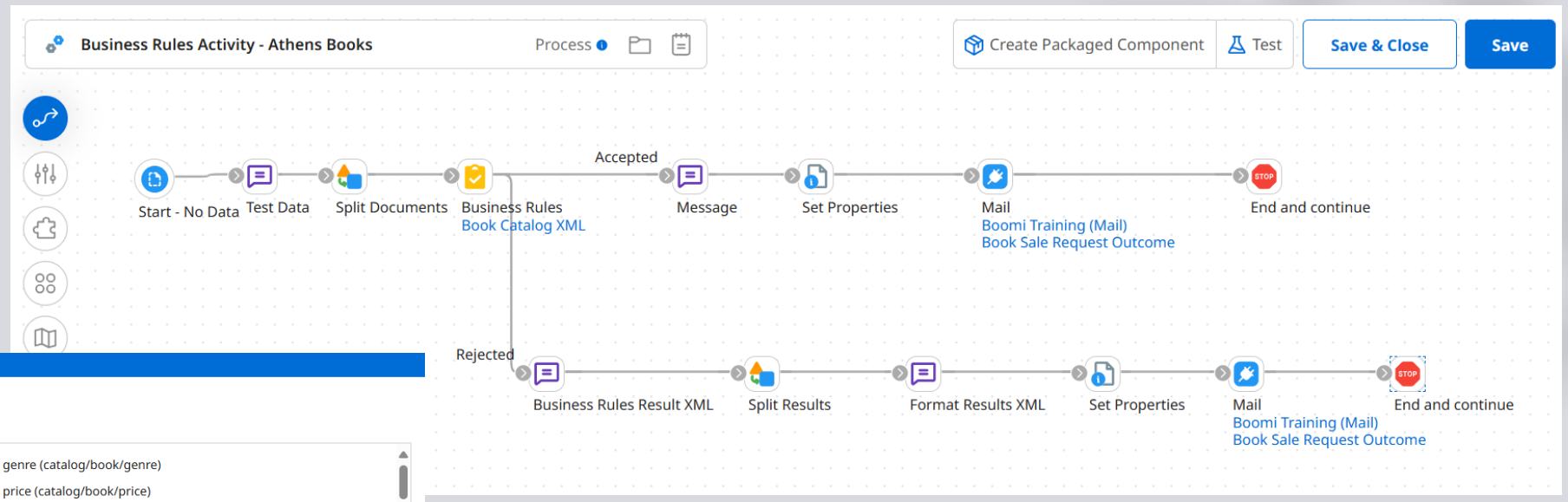
Suggest a short summary up front.

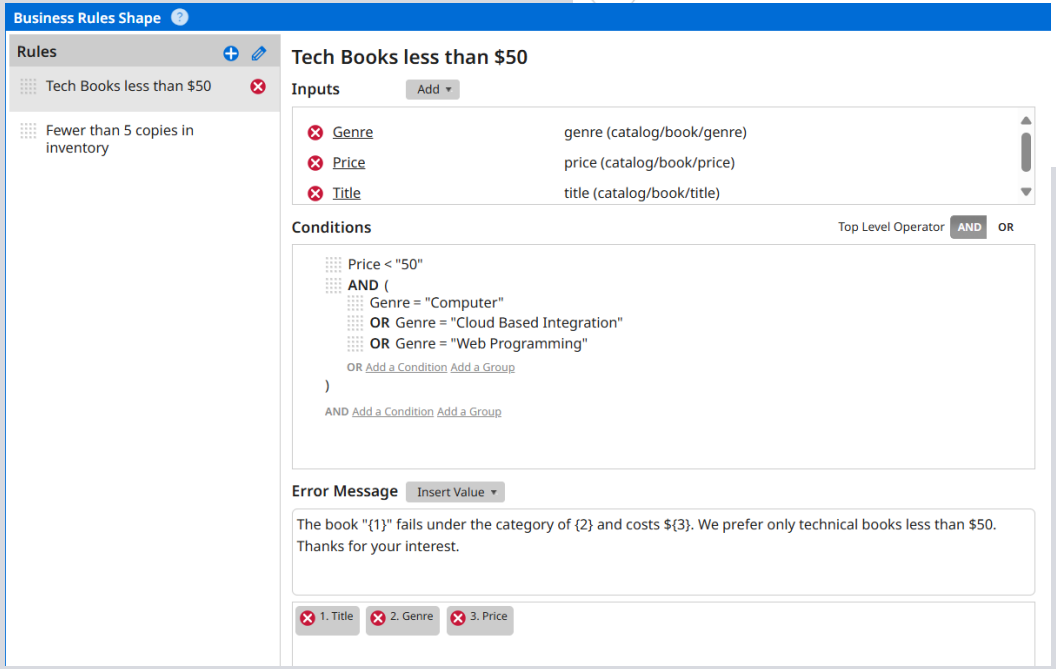
Week 1 – Boomi Training Modules

Associate Integration Developer Certification

Professional Integration Developer Certification

Examples of programs I developed during the Boomi Training Courses

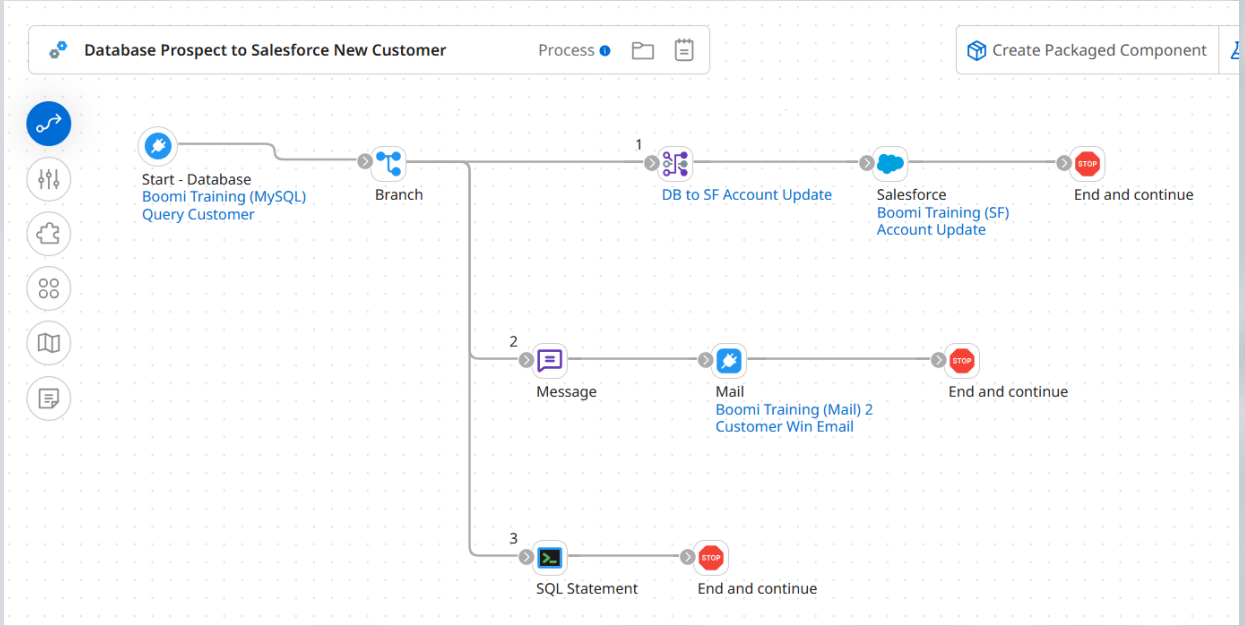




This program worked by getting a few books that a user wanted to sell to a bookstore. The program would then check the details of that book and see if it matched the criteria the store was looking for. If it did, the store would buy the book and email the user to say they are happy to purchase their book. If the book did not meet the criteria, an appropriate error message based on the failing criteria was formatted and this message was emailed to the user explaining why the store did not want to purchase their book.

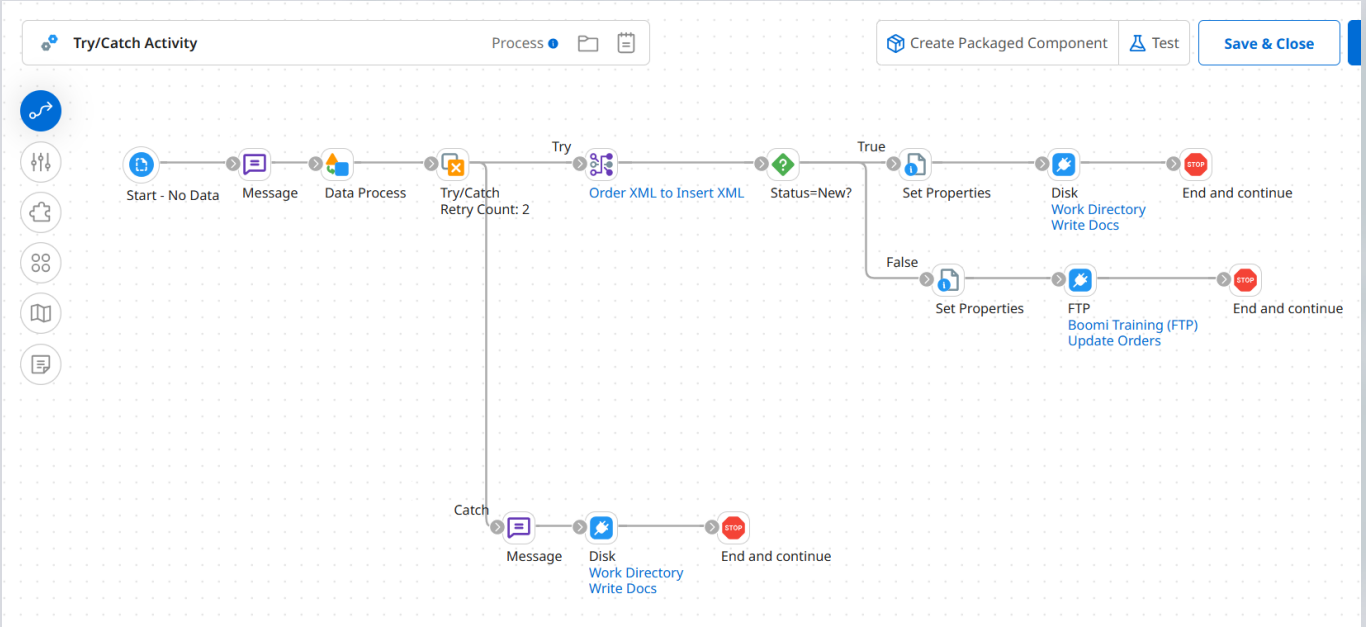
Examples of programs I developed during the Boomi Training Courses

- This program worked to take a database (DB) prospect and convert that to a new salesforce (SF) account.
- The program read in from the DB and followed 3 paths.
- First the DB was mapped to a SF account using a map
- Then a notification was sent out via email to say that the DB prospect has successfully been dealt with
- Then the DB record was updated using an SQL statement



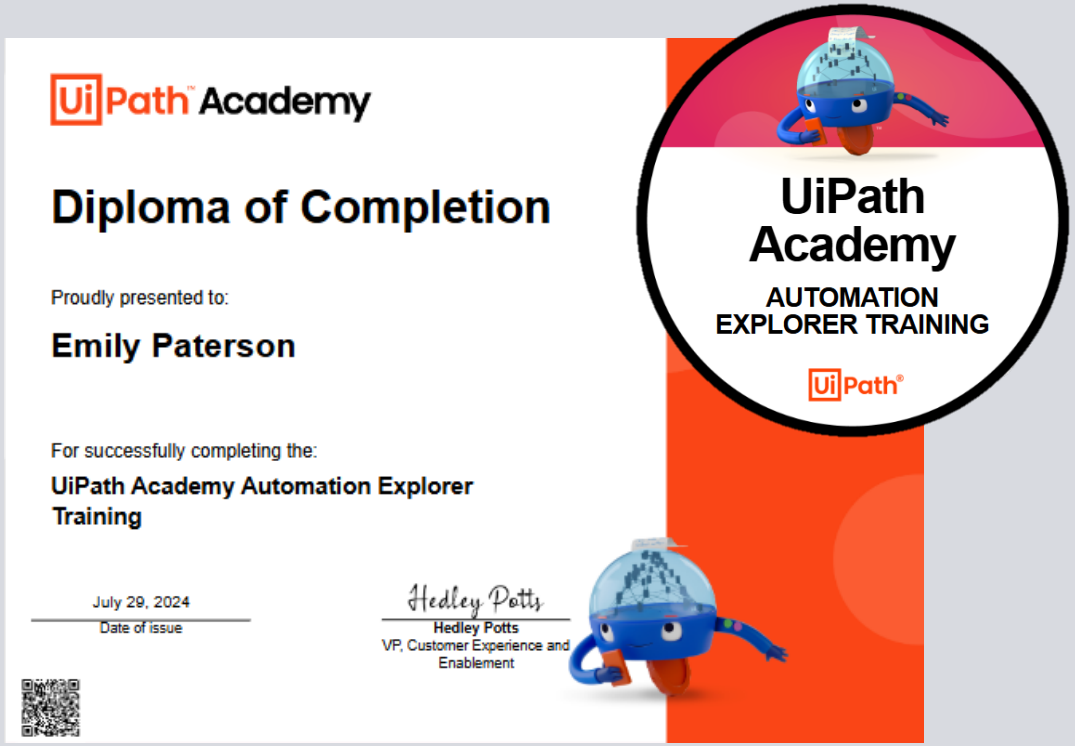
Examples of programs I developed during the Boomi Training Courses

- This program took in a number of orders and tried to either create a new order in the directory or update the current order.
- The order XML is mapped to the Insert XML
- If there are any issues with the mapping process, the file will go down the catch route where an error message is formatted. The order is them saved as it is so a human developer can go and look into the error.
- Otherwise, the order is now checked to see if it's a new order or it is updating a current order. Based on this check, the order is either added to the directory or the order is pulled from the directory and updated.



Week 2 – UiPath Training

Automation Explorer Training Course

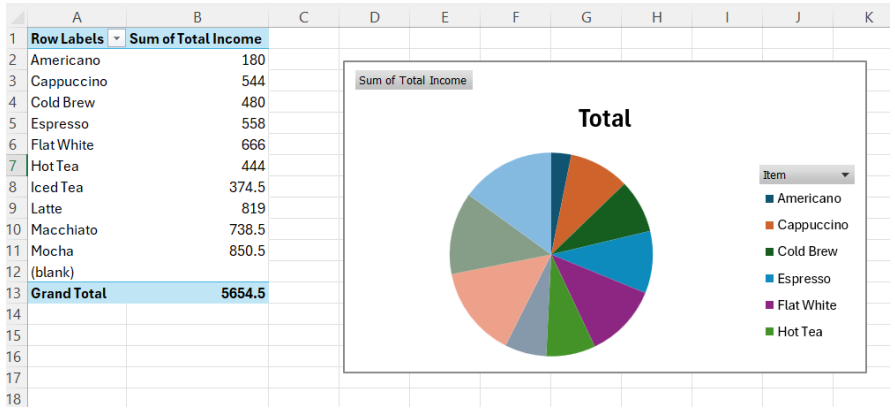


2 main projects

- Scraping Data from a website into an excel file and formatting that data
- Sorting an outlook inbox by getting the sender name and email address and creating folders/subfolders & moving the emails to the appropriate location

Example automation made during the UiPath Automation Explorer Course

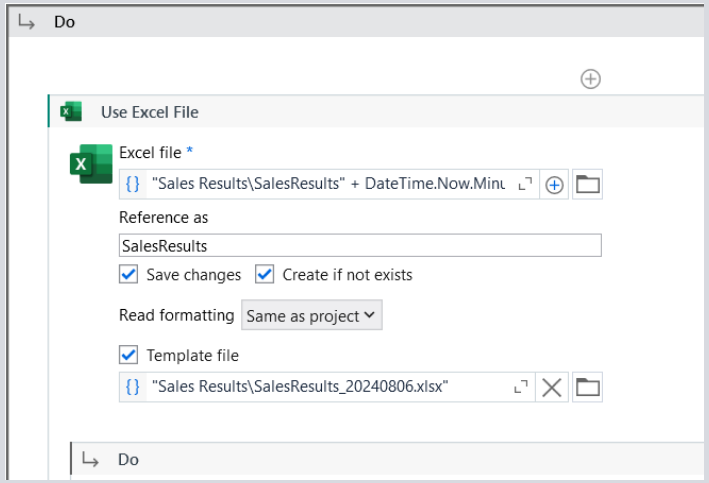
- This project involved taking multiple excel sheets and combining them into one single file to compare the data for multiple days in a coffee shop
- Then (using a pivot table) a chart was created so the data is easy to read and understand
- It includes error checking for when the sheet name is not what we would expect causing us to miss data



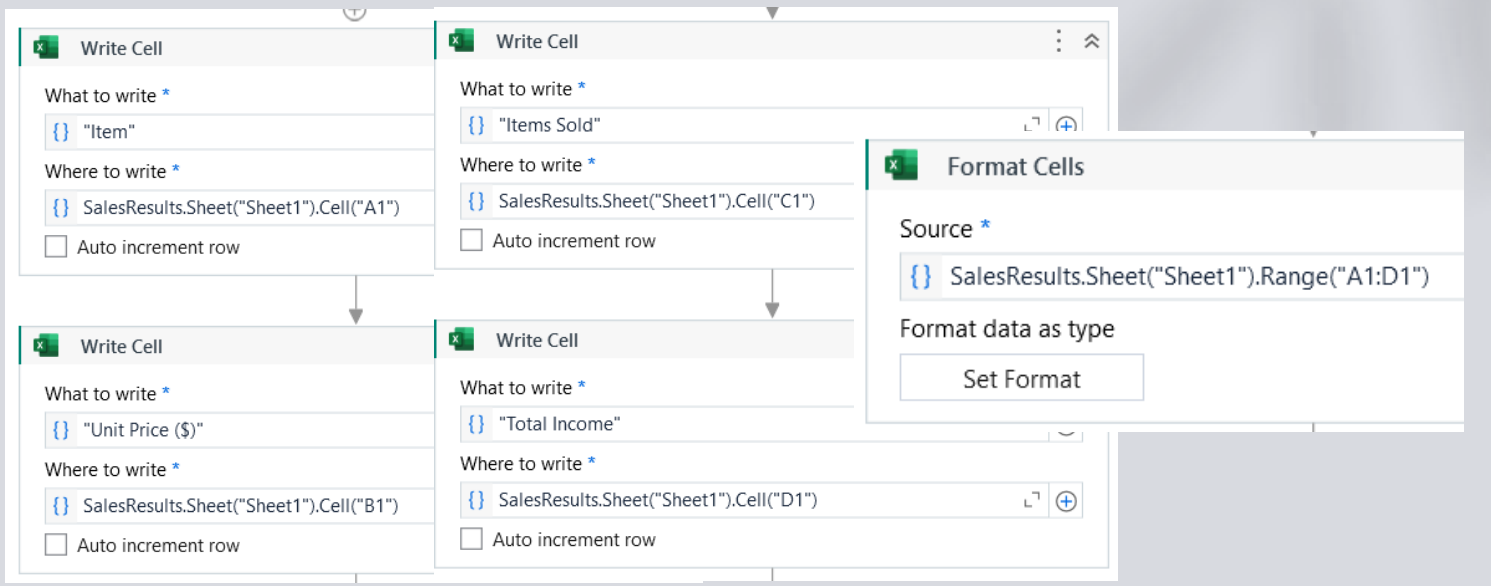
Row Labels	Sum of Total Income
Americano	180
Cappuccino	544
Cold Brew	480
Espresso	558
Flat White	666
Hot Tea	444
Iced Tea	374.5
Latte	819
Macchiato	738.5
Mocha	850.5
(blank)	
Grand Total	5654.5

Step 1: Create the Sales Results file & header columns

Dynamically creates a sales results file using the current time

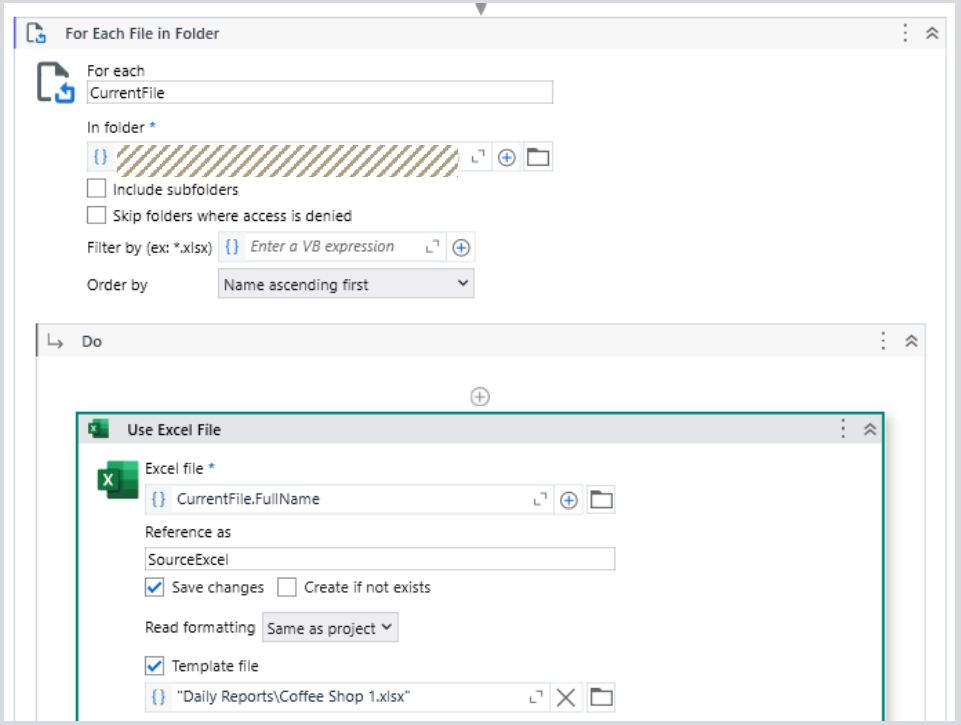


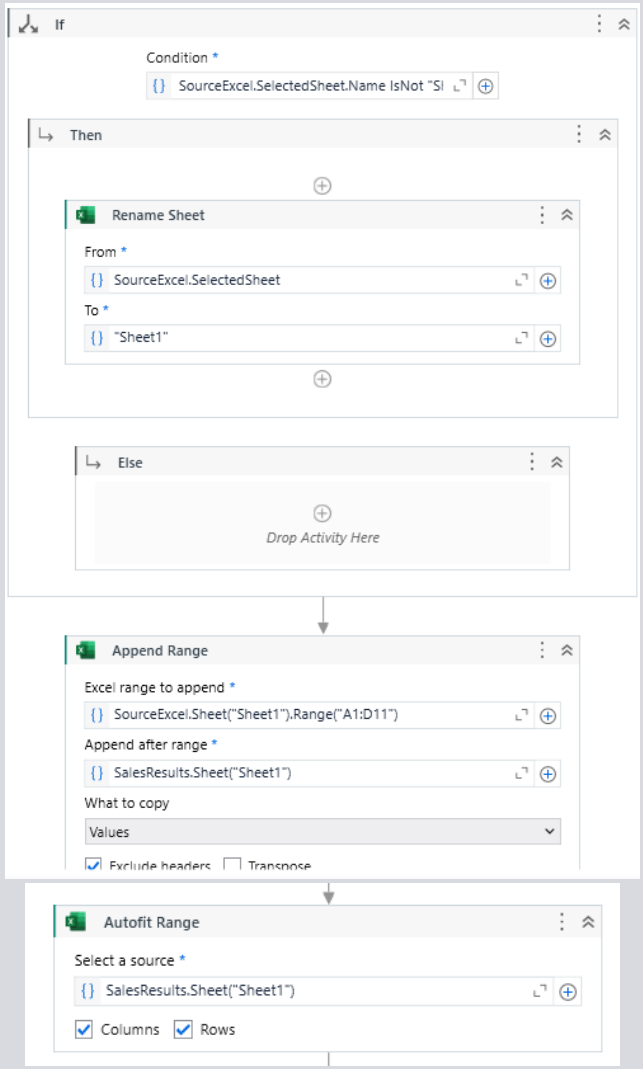
Create the header columns in the excel sheet & format them



Step 2: Go through each of the excel files in the daily reports folder

Iterating through each excel file in the folder

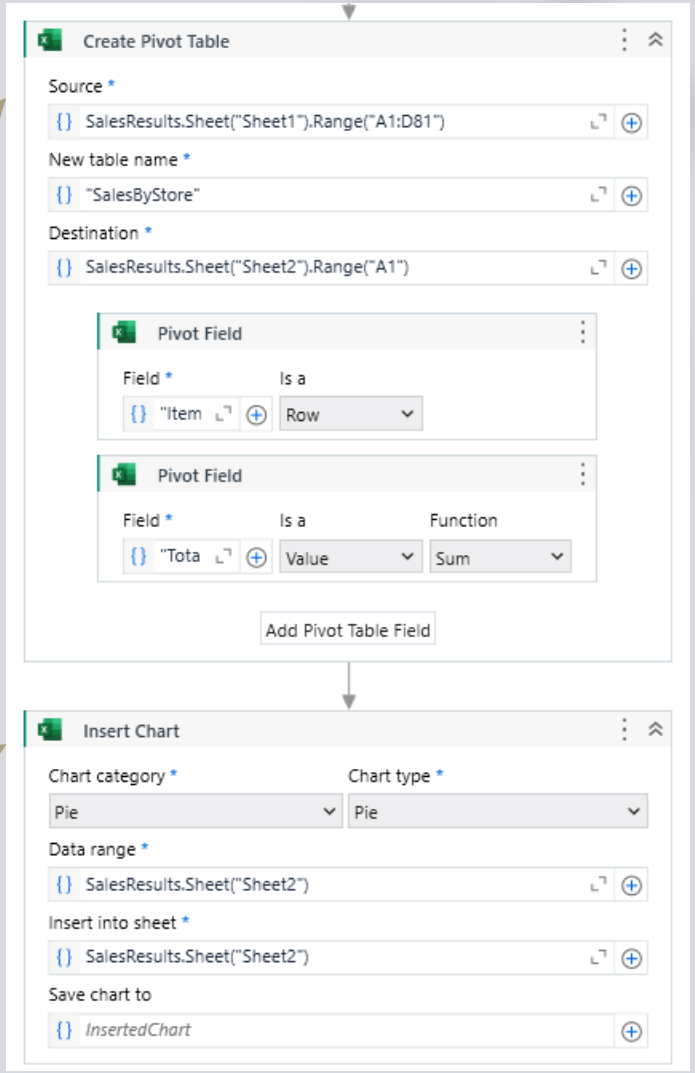




- *Then we have an error checking IF statement to make sure that we collect all the data from the files*
- *We check that the current sheet is called "Sheet1" otherwise its not picked up by Append Range. If it is not "Sheet1" update the title*
- *Then get the data & append to the results sheet & autofit*

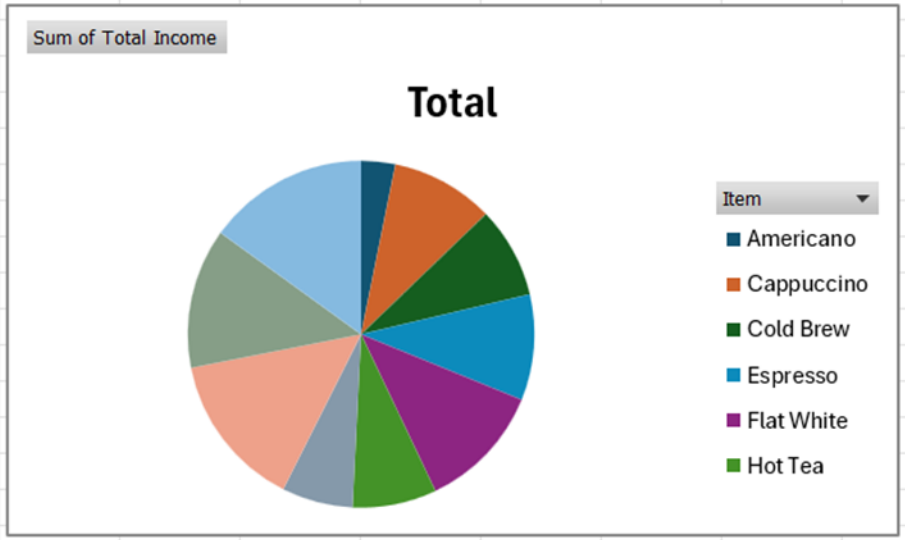
Step 3: Create a pivot table that can be used to make a pivot chart

- *Get the full range of data from the results sheet & create the table with this data*
- *The 2 fields we want displaying on the chart are item name and their total income so we set these as pivot fields*
- *Then we can create a pie chart using the pivot table stored in "Sheet2" & add it in*



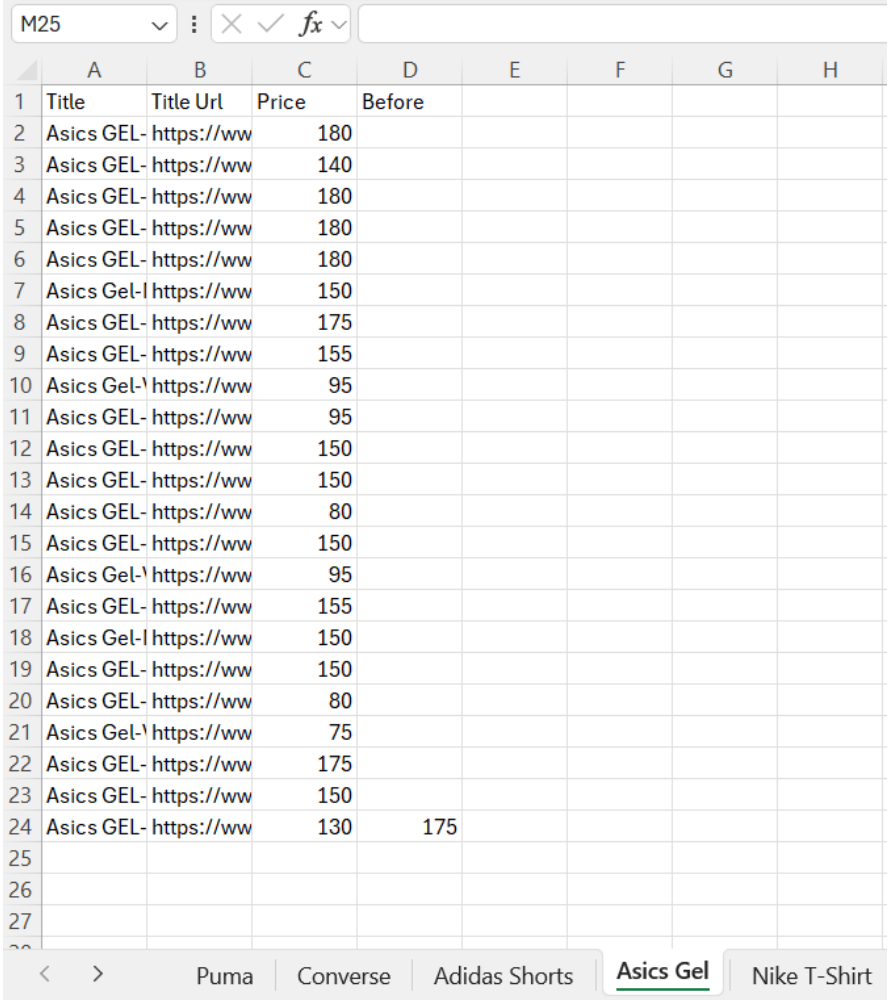
Step 4: View the data in excel

Row Labels	Sum of Total Income
Americano	180
Cappuccino	544
Cold Brew	480
Espresso	558
Flat White	666
Hot Tea	444
Iced Tea	374.5
Latte	819
Macchiato	738.5
Mocha	850.5
(blank)	
Grand Total	**5654.5**



Scaping Data Project

- This project loaded up a webpage (JD Sports) and searched for various items. The process used an excel sheet for the input and that sheet contained a list of items the user wanted information about. The project took in the item name, URL, Standard price and (if applicable) a discounted price if a sale was currently running.

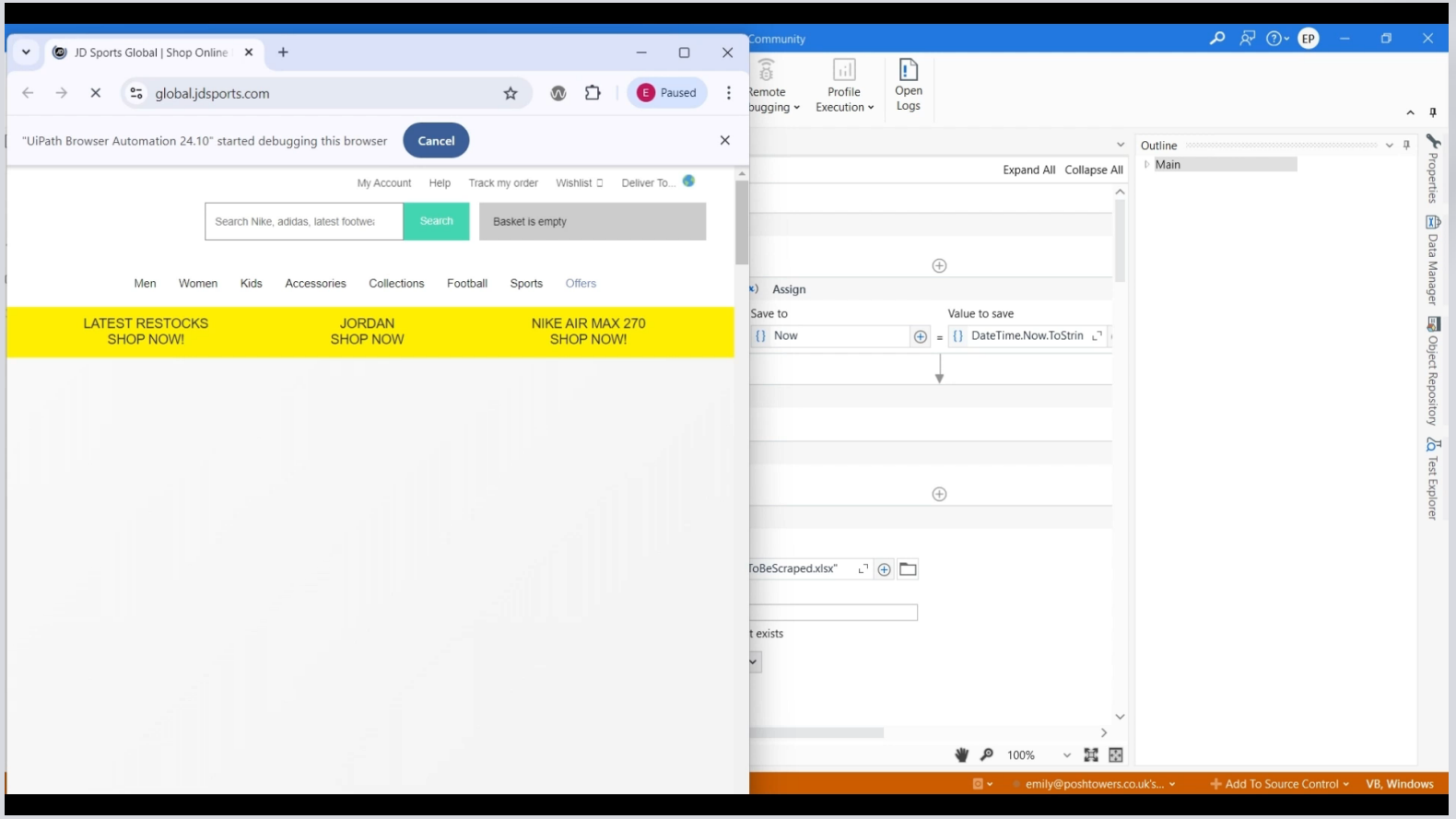


M25

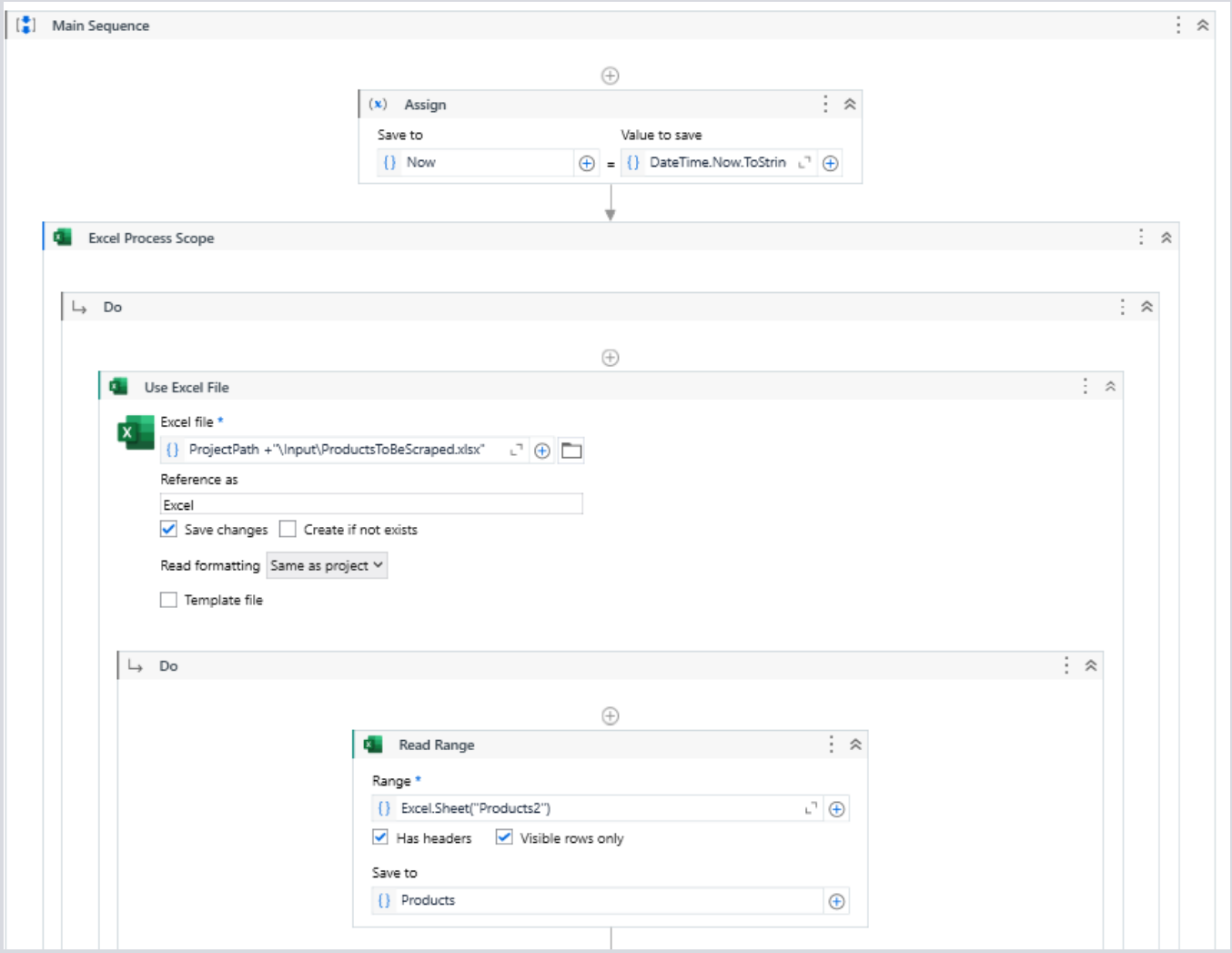
	A	B	C	D	E	F	G	H
1	Title	Title Url	Price	Before				
2	Asics GEL-	https://ww	180					
3	Asics GEL-	https://ww	140					
4	Asics GEL-	https://ww	180					
5	Asics GEL-	https://ww	180					
6	Asics GEL-	https://ww	180					
7	Asics Gel-I	https://ww	150					
8	Asics GEL-	https://ww	175					
9	Asics GEL-	https://ww	155					
10	Asics Gel-\	https://ww	95					
11	Asics GEL-	https://ww	95					
12	Asics GEL-	https://ww	150					
13	Asics GEL-	https://ww	150					
14	Asics GEL-	https://ww	80					
15	Asics GEL-	https://ww	150					
16	Asics Gel-\	https://ww	95					
17	Asics GEL-	https://ww	155					
18	Asics Gel-I	https://ww	150					
19	Asics GEL-	https://ww	150					
20	Asics GEL-	https://ww	80					
21	Asics Gel-\	https://ww	75					
22	Asics GEL-	https://ww	175					
23	Asics GEL-	https://ww	150					
24	Asics GEL-	https://ww	130	175				
25								
26								
27								

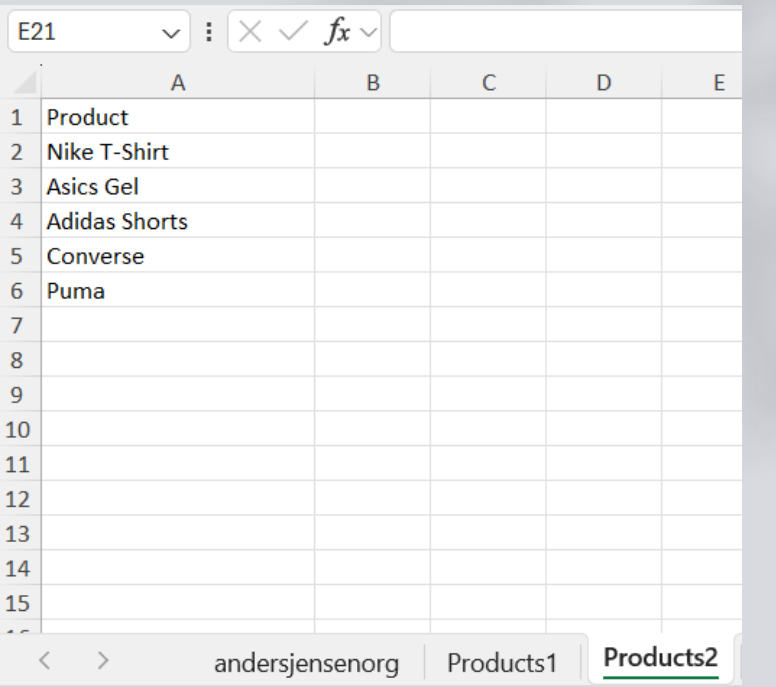
Puma | Converse | Adidas Shorts | Asics Gel | Nike T-Shirt

Video Of Process: (Actions in video are being completed by the bot)



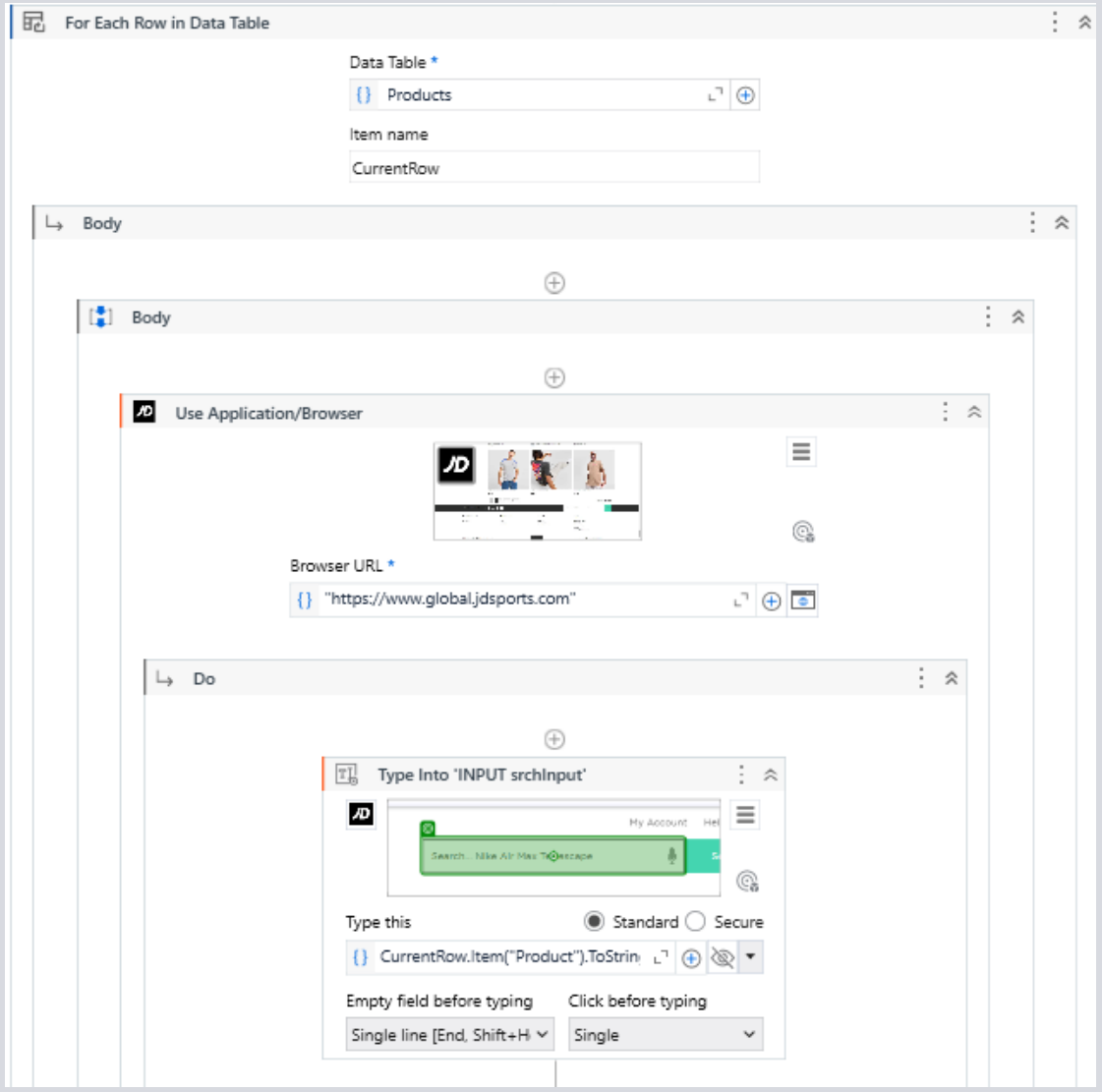
Step 1: Take in input from Excel File





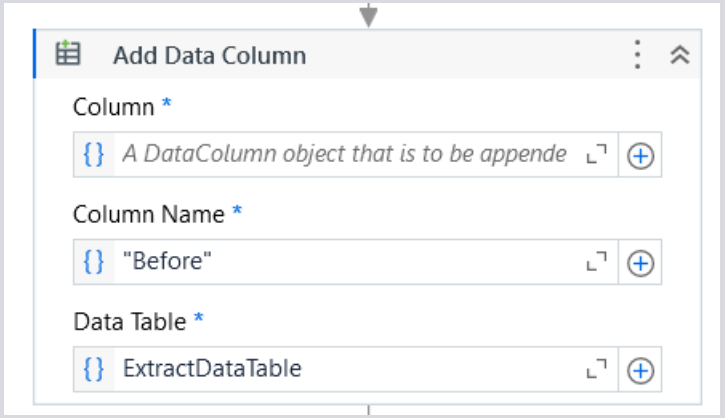
- The Project reads in from the excel file (ProductsToBeScraped.xlsx)
- Then it reads the range of values stored in the sheet titled “Products2”

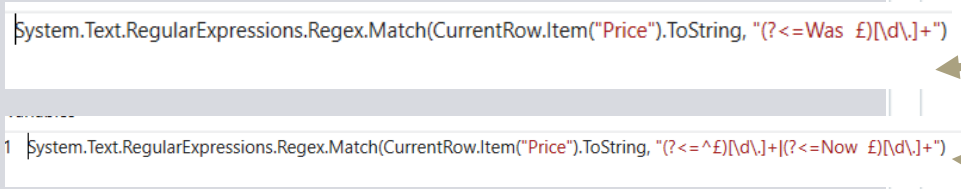
Step 2: Load the webpage and in turn search for the items, then extract the data

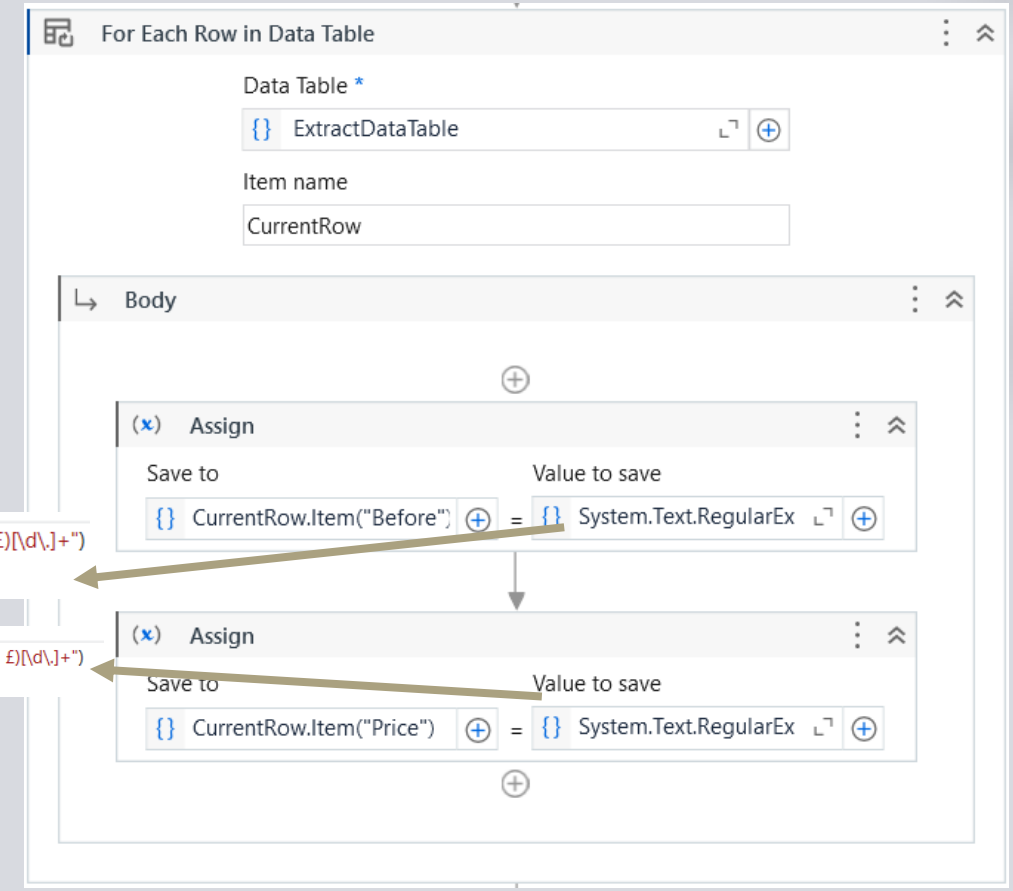


- The project will load up the JD website using the browser and type into the search bar the name of the product
- Then using Extract Data Table, the title, URL and price are extracted from the webpage for each product that comes up after the search

Step 3: Format the price so that the regular and discount prices are in different columns

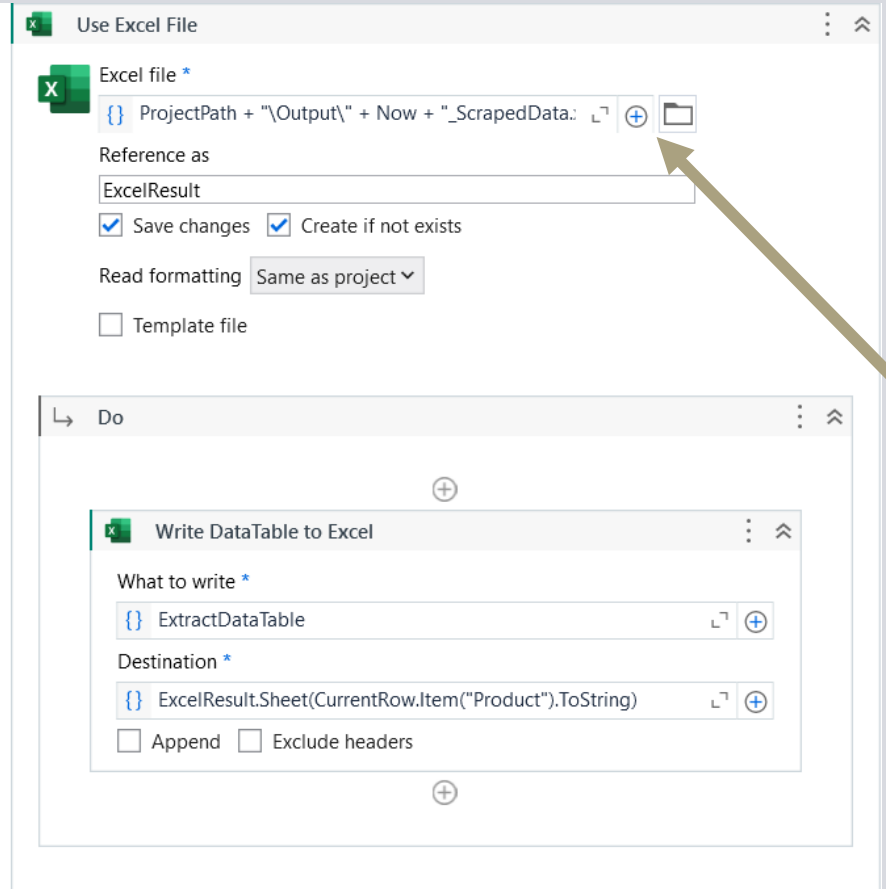




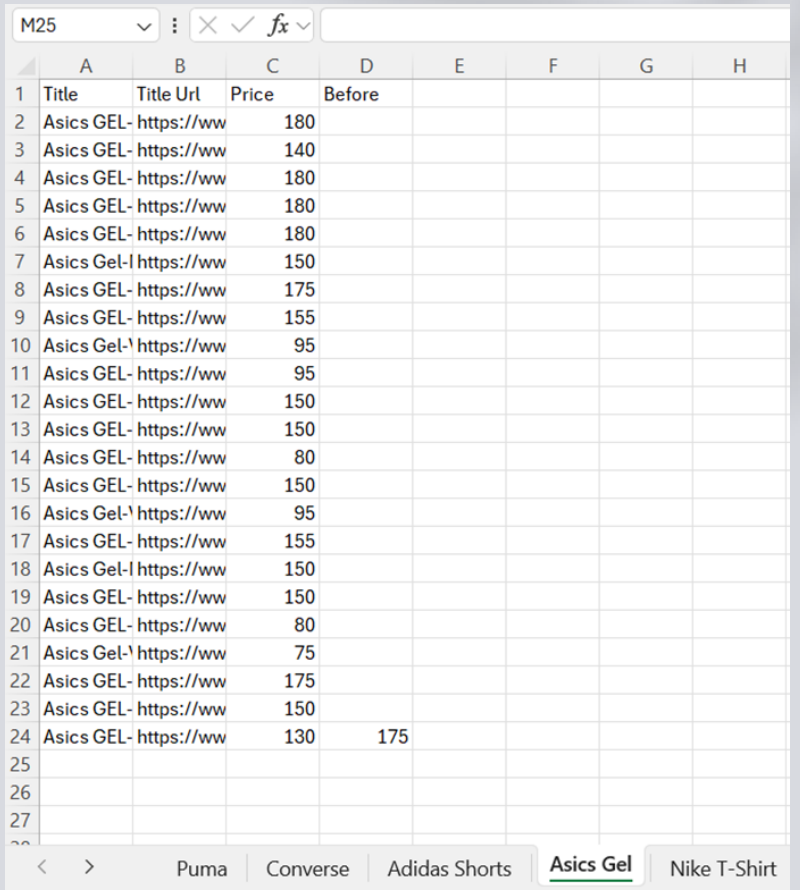


- Adds a new column called before price
- Extracts the before price using Regex and adds that to the ‘Before’ column
- Then extracts the after price using Regex and adds that to the ‘Price’ column

Step 4: Export the table as an Excel File



- Finally, take the data table and write that to an excel file
- Dynamically name the excel file so that it has the current date and time



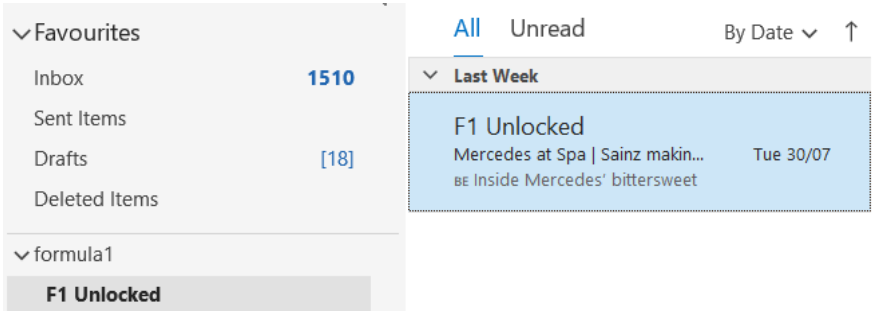
M25

	A	B	C	D	E	F	G	H
1	Title	Title Url	Price	Before				
2	Asics GEL-	https://ww	180					
3	Asics GEL-	https://ww	140					
4	Asics GEL-	https://ww	180					
5	Asics GEL-	https://ww	180					
6	Asics GEL-	https://ww	180					
7	Asics Gel-I	https://ww	150					
8	Asics GEL-	https://ww	175					
9	Asics GEL-	https://ww	155					
10	Asics Gel-\	https://ww	95					
11	Asics GEL-	https://ww	95					
12	Asics GEL-	https://ww	150					
13	Asics GEL-	https://ww	150					
14	Asics GEL-	https://ww	80					
15	Asics GEL-	https://ww	150					
16	Asics Gel-\	https://ww	95					
17	Asics GEL-	https://ww	155					
18	Asics Gel-I	https://ww	150					
19	Asics GEL-	https://ww	150					
20	Asics GEL-	https://ww	80					
21	Asics Gel-\	https://ww	75					
22	Asics GEL-	https://ww	175					
23	Asics GEL-	https://ww	150					
24	Asics GEL-	https://ww	130	175				
25								
26								
27								

Puma | Converse | Adidas Shorts | Asics Gel | Nike T-Shirt

Outlook Sorting Project

- This project will go through each email in my inbox and extract the name of the email sender. It will then check if I have requested that sender have a parent folder via an excel sheet. If I have, a parent folder is created first then the subfolder. If at any point the program runs into an error and the folder has already been created, then this is noted in the excel file and then the program can check if the folder has already been made rather than having to encounter the file already made error several times. After the folders are made, the email is moved to the corresponding folder.

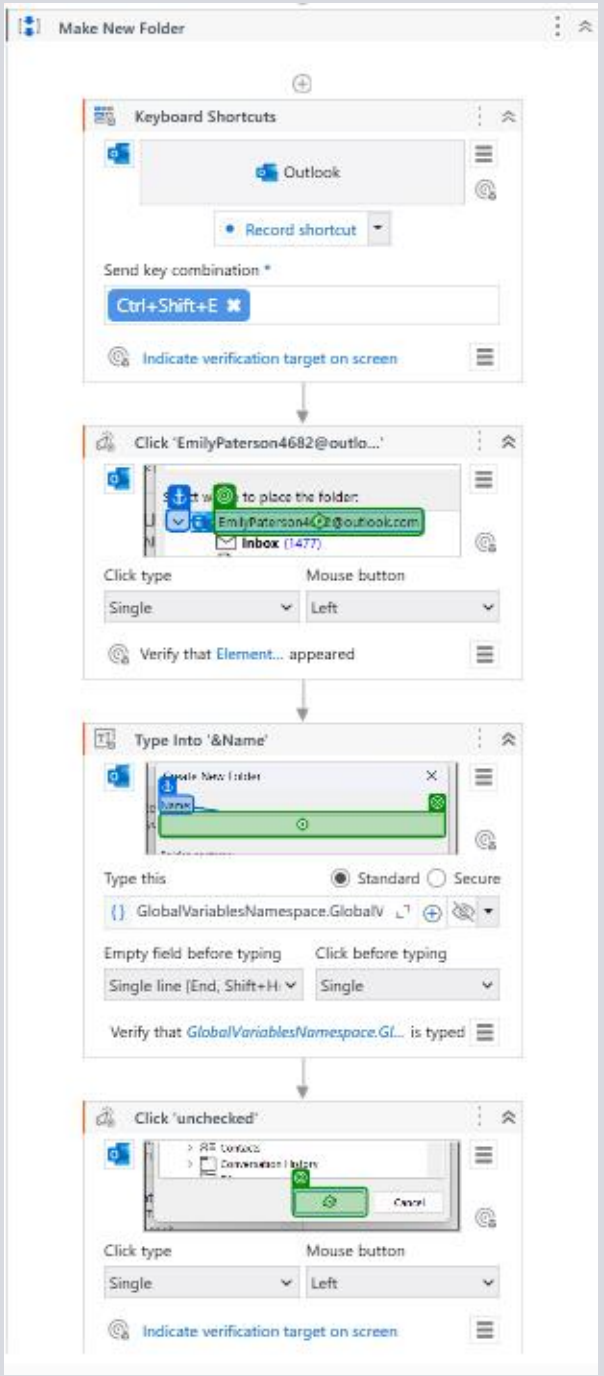


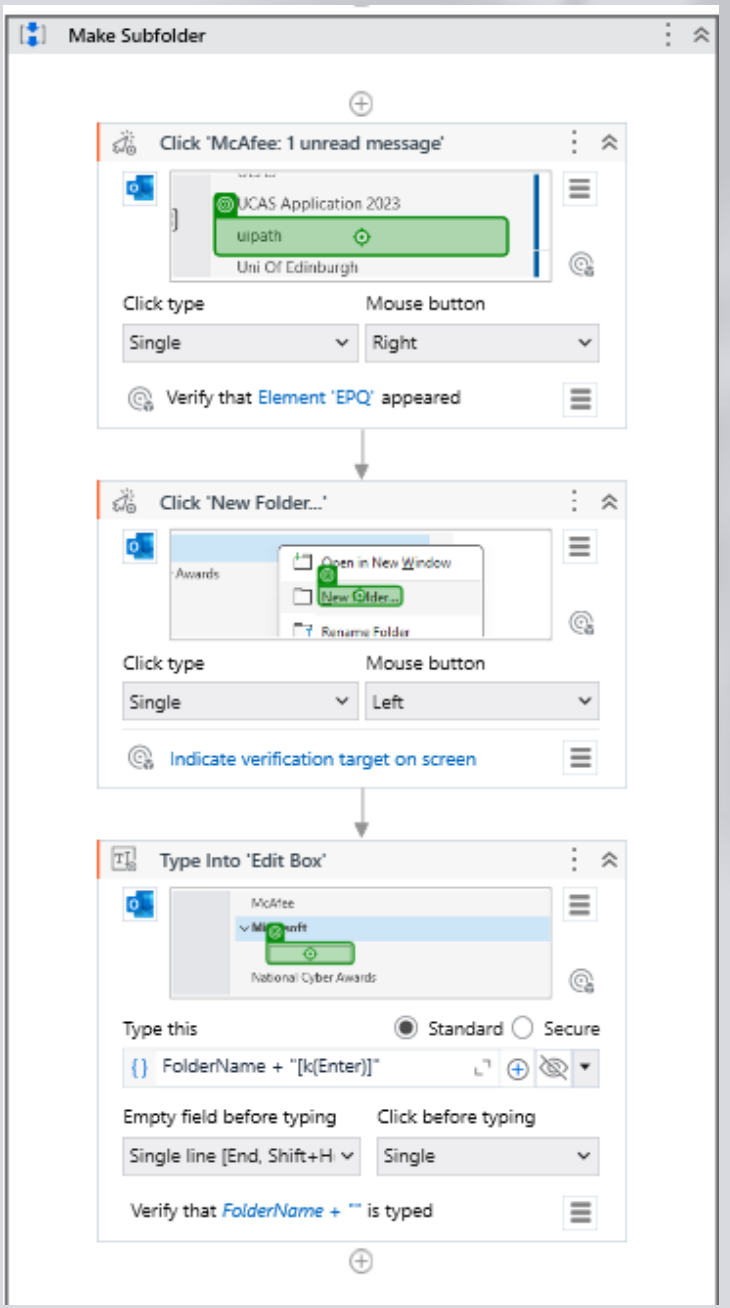
Video of process working: (Actions in video are being completed by the bot)

An example of a couple of standard subprocesses used within the program

1. *Making a new folder using the CTRL + Shift + E shortcut. Selects main folder for destination, types the folder name in and then presses ok*

2. *Making a subfolder, right clicks on parent folder to make a subfolder, types in the subfolder name and presses enter*

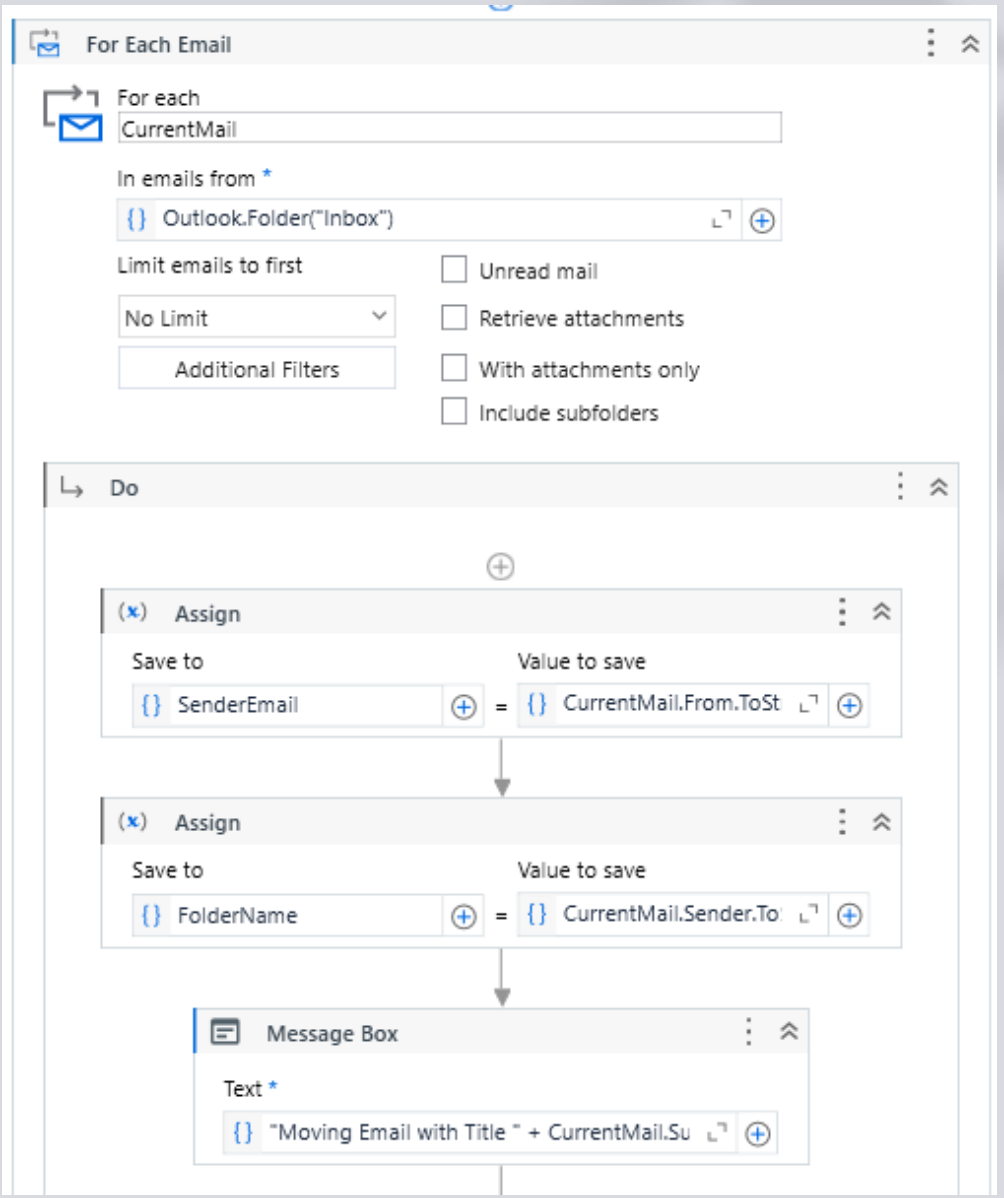




Step 1: Get Email & Get Email Sender

- *Goes through each email in my outlook Inbox and repeats the whole process*

- *Gets the email and extracts the senders Email*
- *It also creates the folder name by extracting the sender's name*
- *It then alerts the user to tell them the email is being moved*

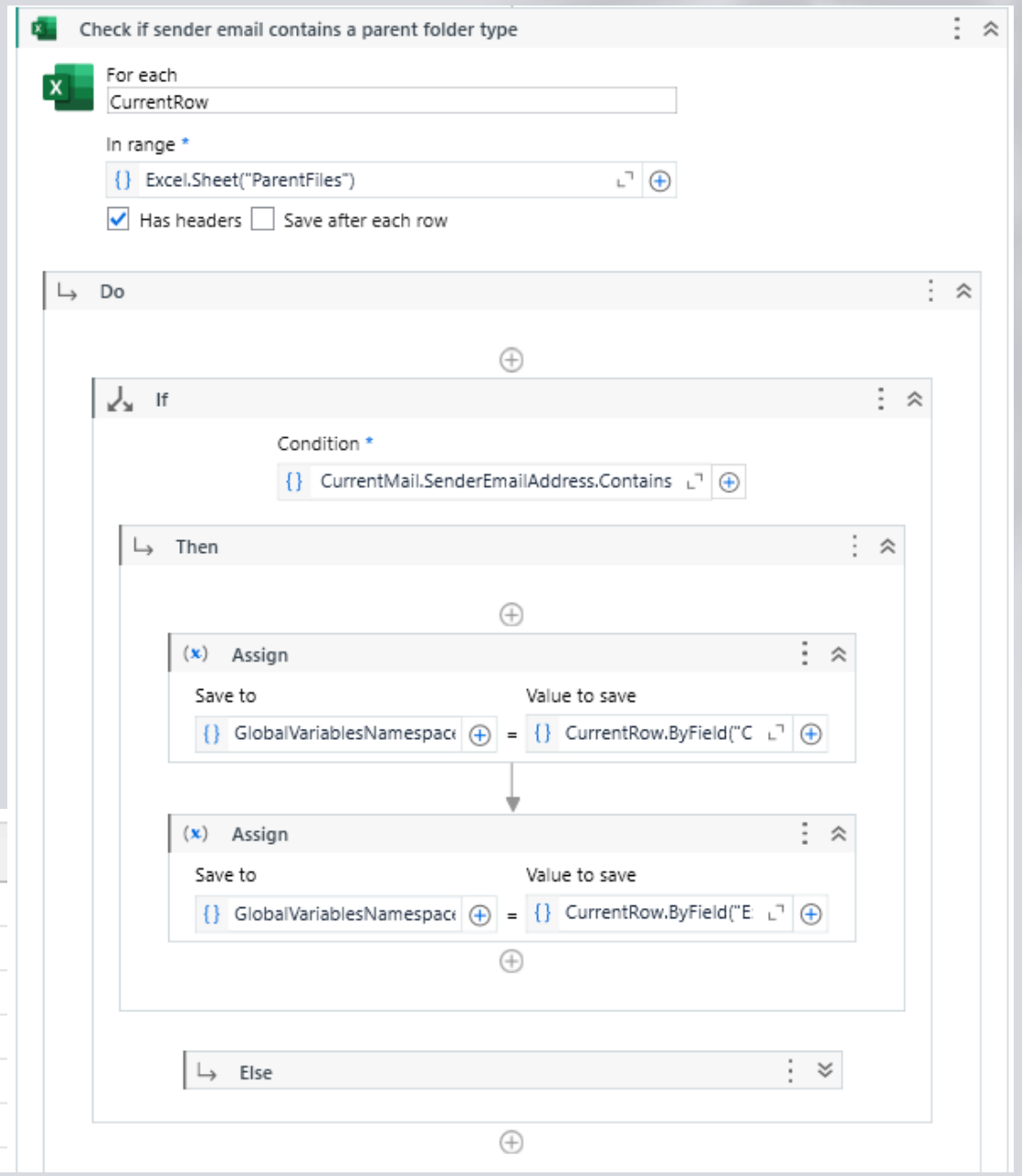


Step 2: Looks through an excel file to determine if I want a parent folder made for that sender

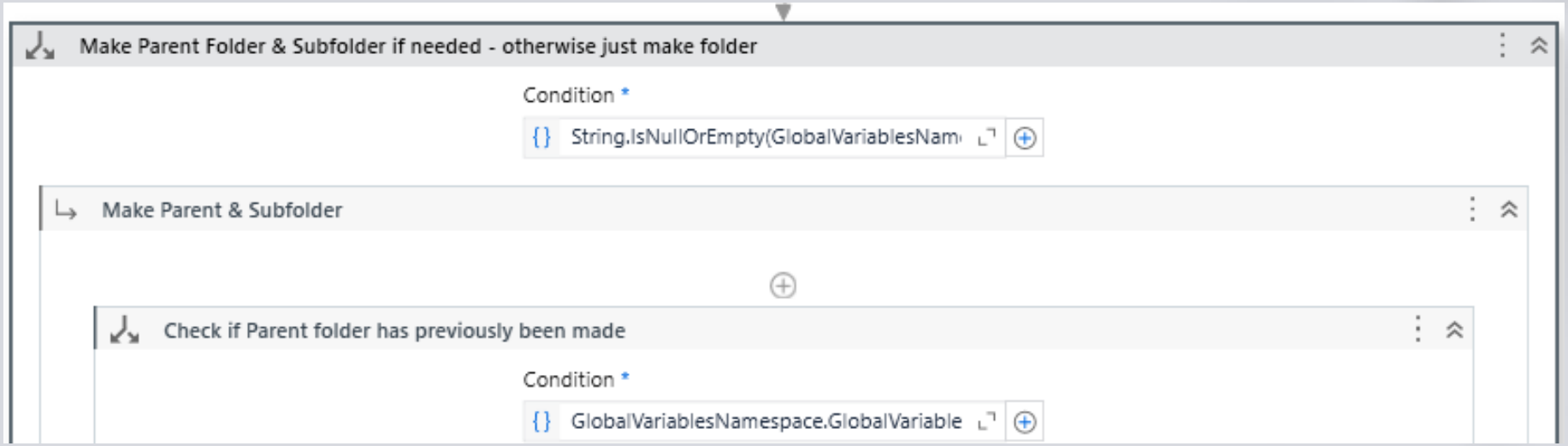
- *In an excel file, the ParentFiles sheet contains a list of companies I would like to have a parent file*
- *This means the Folder name generated in the previous step will be the name of the subfolder*
- *For example, if I had multiple people in a company sending me emails, I want all these emails under 1 parent folder named with the company name*
- *The excel file also keeps track of if the Parent Folder already exists in my outlook account*

- *To check if the current email needs a parent folder, we check if the sender email contains ant of the parent company names*
- *If it does, we save the parent folder name and if it exists or not*

	A	B
1	Company	Exists?
2	Serco	FALSE
3	Microsoft	FALSE
4	formula1	TRUE
5	uipath	TRUE
6		

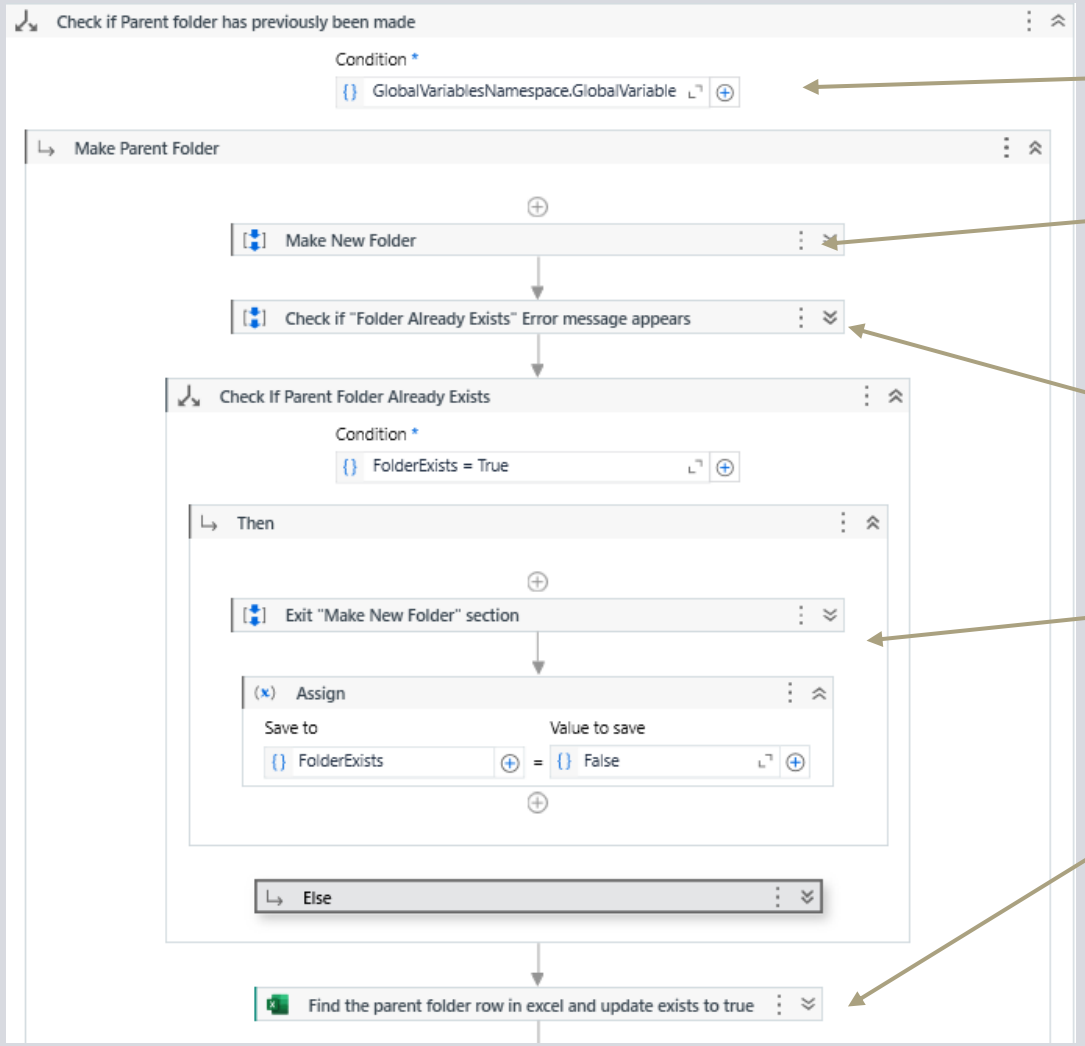


Step 3: Check if we need to make a parent & subfolder or just a folder



- *Checks If the variable 'ParentFolderName' is empty or null*
- *If it is, then no Parent folder is needed and the email sender's name can just be used to make a regular folder*
- *If it is not null or empty, then a parent folder needs to be made*
- *For Next Slides:*
  - **Parent & Subfolder: Steps 4 – 6**
  - **Just Regular Folder: Steps 7 -**

Step 4 (Parent & Subfolder): Attempt to make Parent Folder



Check if the Exists variable set earlier is equal to FALSE

If it is, then the parent folder doesn't exist so we make it via the Make New Folder subprocess. Otherwise skip to step 5

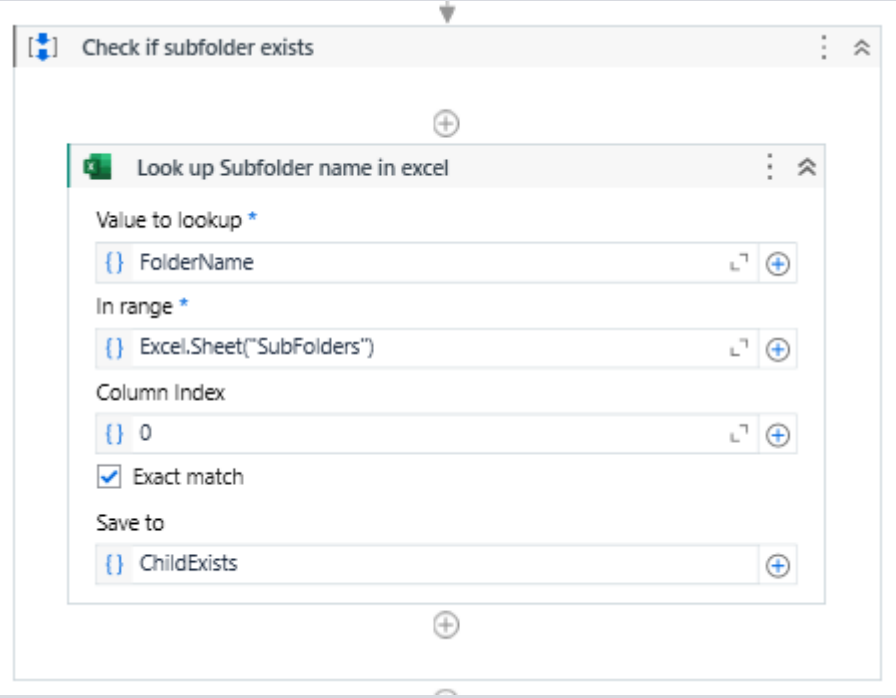
Then we check for an error message of (Folder Already Exists) by trying to find that element on the screen

If that error message does exist then close the make new folder window, reset the value of folder exists for the next email

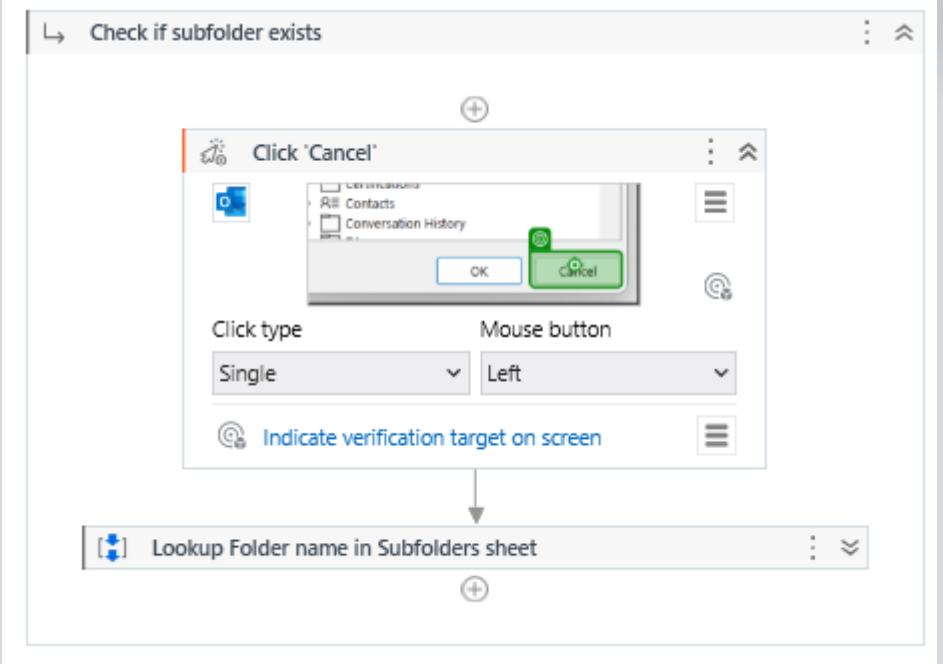
Then regardless if the error appears, update the excel file to show that the parent folder does exist by updating the Exists? Column to TRUE

Step 5 (Parent & Subfolder): Check if subfolder exists

- *First check if the subfolder name is listed in the excel file*
- *If it is listed, the subfolder already exists so we can skip through this step*

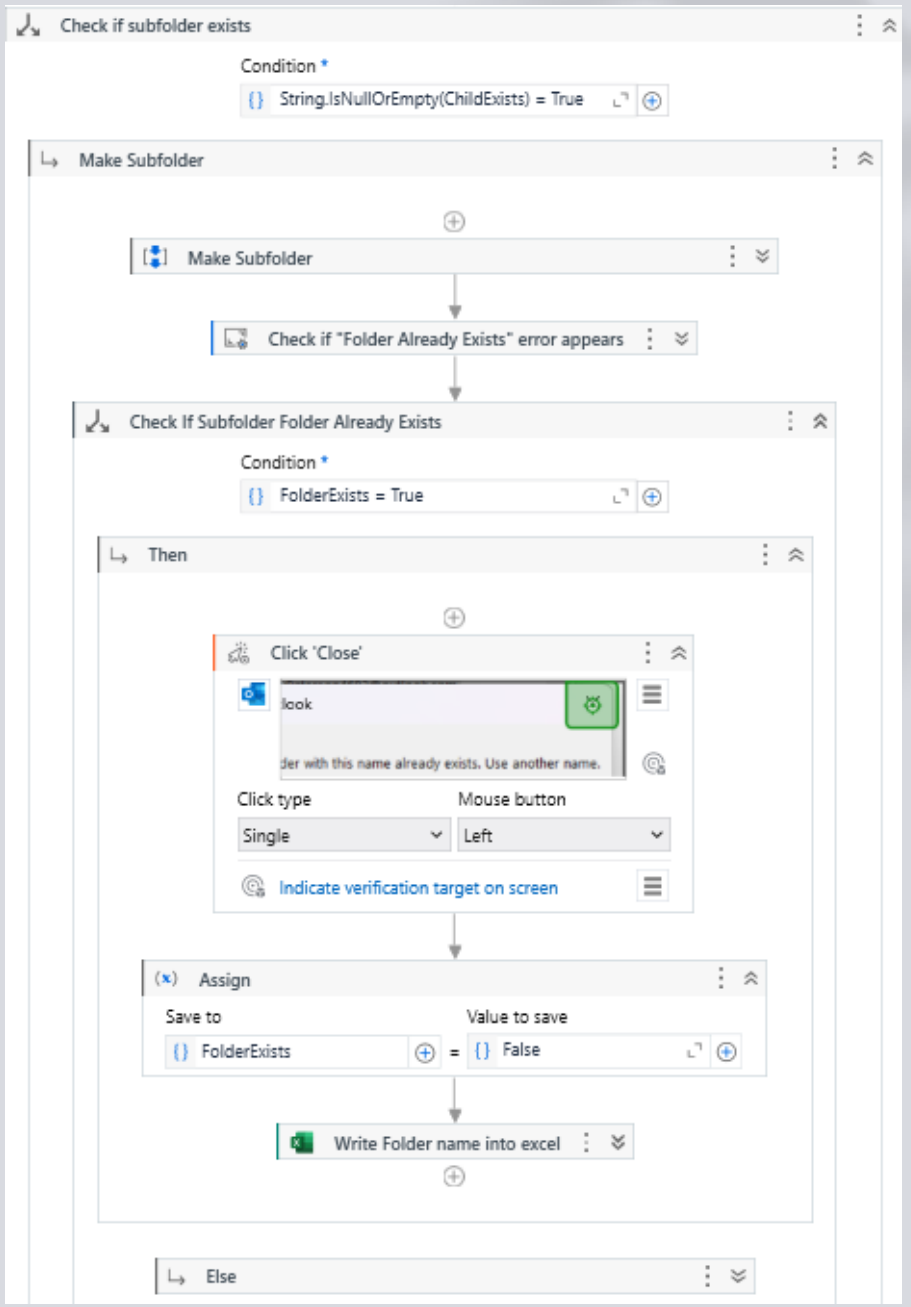


- *If we knew the parent folder already existed, we would have immediately moved to this step*
- *We can cancel the make new folder as we know the parent folder exits and then go to the same step of checking if the subfolder name exists in the excel folder*

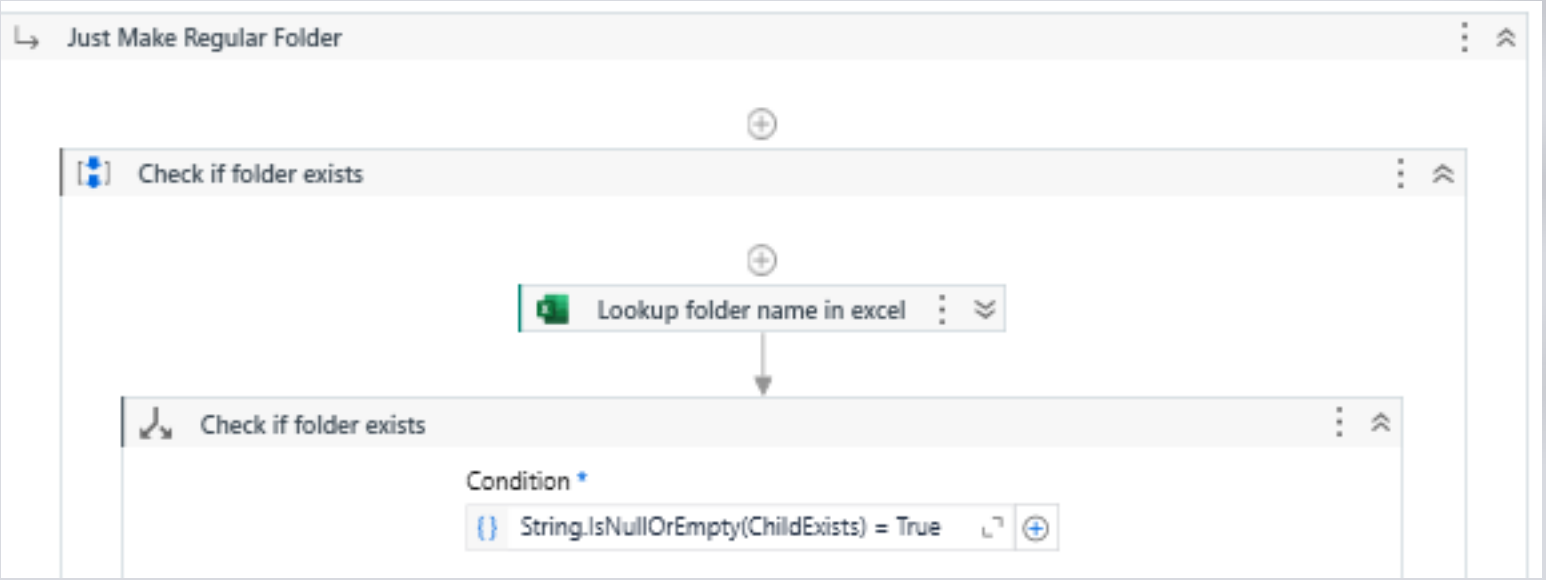


Step 6 (Parent & Subfolder): Make Subfolder

- *Checks if the 'ChildExists' variable is Null Or Empty*
- *If it is Null/Empty then the subfolder doesn't exist so we need to make the subfolder. Otherwise skip straight to step 9*
- *Use the Make Subfolder Subprocess to make the subfolder and again check if the folder already exists error appears*
- *If it does cancel the make new folder operation, reset FolderExists Variable to false*
- *Then write the subfolder name into excel*



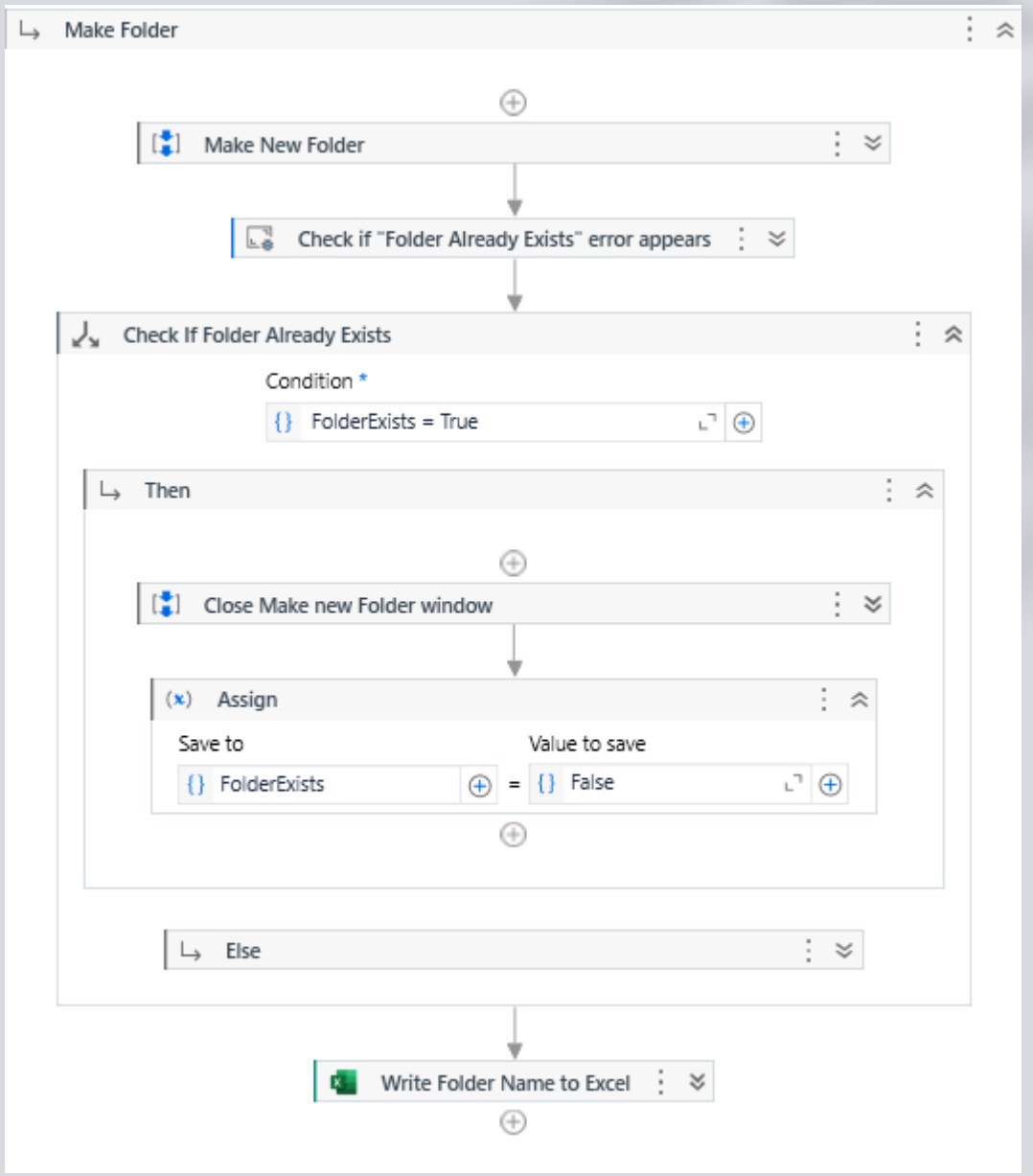
Step 8 (Regular Folder): Check if folder already exists



- *Using the same lookup process as making the subfolders, this checks if the folder name is already listed in the excel file*
- *If the folder already exists, then skip to step 9*
- *Otherwise, make the folder*

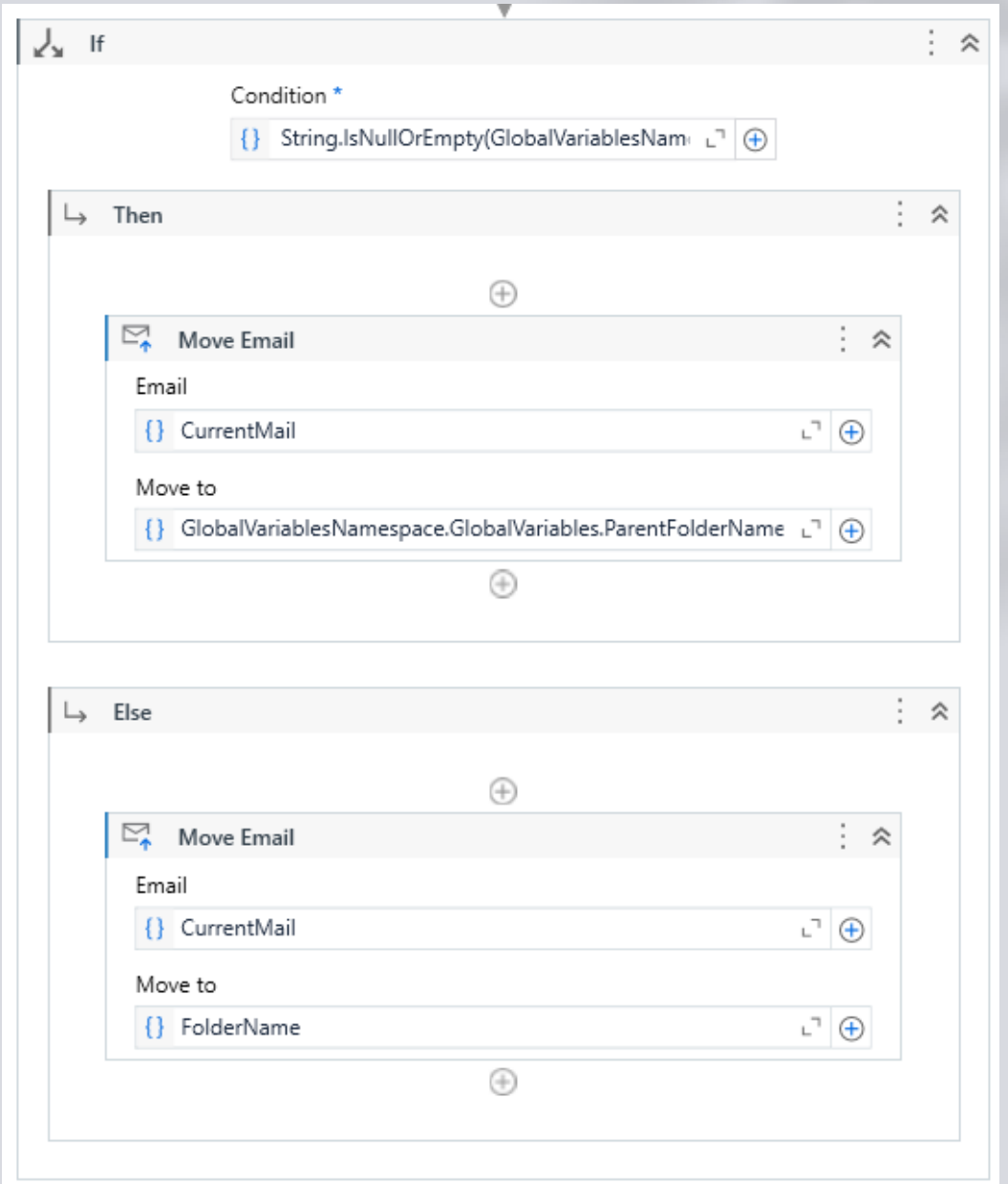
Step 8 (Regular Folder): Attempt to make new folder

- *Use the Make New folder subprocess again to make new folder*
- *Check for "Folder Already Exists" error and deal with it in the same way as previous step (Step 4)*
- *Then write the folder name into excel*



Step 9: Move the email

- *Checks if the current email has a parent folder again by checking if the 'ParentFolderName' is null or empty*
- *If the email does have a parent folder:*
  - **Move to ParentFolderName/FolderName**
- *If the email doesn't have a parent folder:*
  - **Move to FolderName**



Step 10: See the Results

Example: Moving this F1 Unlocked email to the F1 Unlocked folder under the parent folder formula1

Week 3 – Further Work On UiPath

1 personal project

- Involved taking employee data from an excel file, performing calculations on that data in order to format and email a payslip to that employee.

Started working on/overshadowing the development of a healthcare app system

- Development of the dispatcher & performer processes
- Started work on the dispatcher process in order to get a list of all the current reports in the Safety Observations Power App
- Then started work on the performer to analyse the data in those reports and collect the relevant data

Payslip Generator Project

- This project takes in data about employees from an excel file. It reads all the data and converts this into a data table within UiPath so the data can be easily accessed from within the program. The program can process 3 types of pay slips (UK, USA and Europe (Belgium)) based on the employee's home office.
- The program will then calculate the employee's monthly income based on their completed contract hours and (if applicable) overtime ours and holiday days (holiday day is assumed to be 8 hours). The program then calculates all relevant taxes based on their location. The program will then take all this data and format it into a pay slip.
- The pay slip can then be saved as a PDF and emailed to the employee (Currently my email address is used instead of the employees to avoid sending people random emails).

SERCO PAYSLIP

EMPLOYEE INFORMATION	PAY DATE	HOME OFFICE ADDRESS	EMPLOYEE ADDRESS	NI NUMBER	EMAIL	JOB TITLE
SADIE OWENS	08/05/2024 22:10:18	16, Bartley Wood Business Park, Hook, RG27 9UY	1 Church Lane,Portsmouth,PO57 7XK	FJ375937O	sadie.owens4@serco.com	Cleaner

EARNINGS	HOURS	RATE	TOTAL
STANDARD PAY	120	14	1680
OVERTIME PAY	8	21	168
HOLIDAY PAY (3 DAYS)	24	10	252

DEDUCTIONS	TOTAL
NATIONAL INSURANCE	84
INCOME TAX	210
PENSION CONTRIBUTION	
STUDENT DEBT REPAYMENT	

UK Pay slip

SERCO PAYSLIP

EMPLOYEE INFORMATION	PAY DATE	HOME OFFICE ADDRESS	EMPLOYEE ADDRESS	SOCIAL SECURITY NUMBER	EMAIL	JOB TITLE
LIBERTY FLETCHER	08/05/2024 23:13:31	12930 Worldgate Drive, Suite 6000, Herndon, VA 20170	Suite 108 7129 Thiel Meadows,New Latesha,KY 54619-6353	573-34-4535	liberty.fletcher3@serco.com	Intern

EARNINGS	HOURS	RATE	TOTAL
STANDARD PAY	160	10	1600
OVERTIME PAY	40	15	600
HOLIDAY PAY (0 DAYS)	0	8	0

DEDUCTIONS	TOTAL
MEDICARE TAX	175
FEDERAL INCOME TAX	246
SOCIAL SECURITY TAX	136
STUDENT DEBT REPAYMENT	0

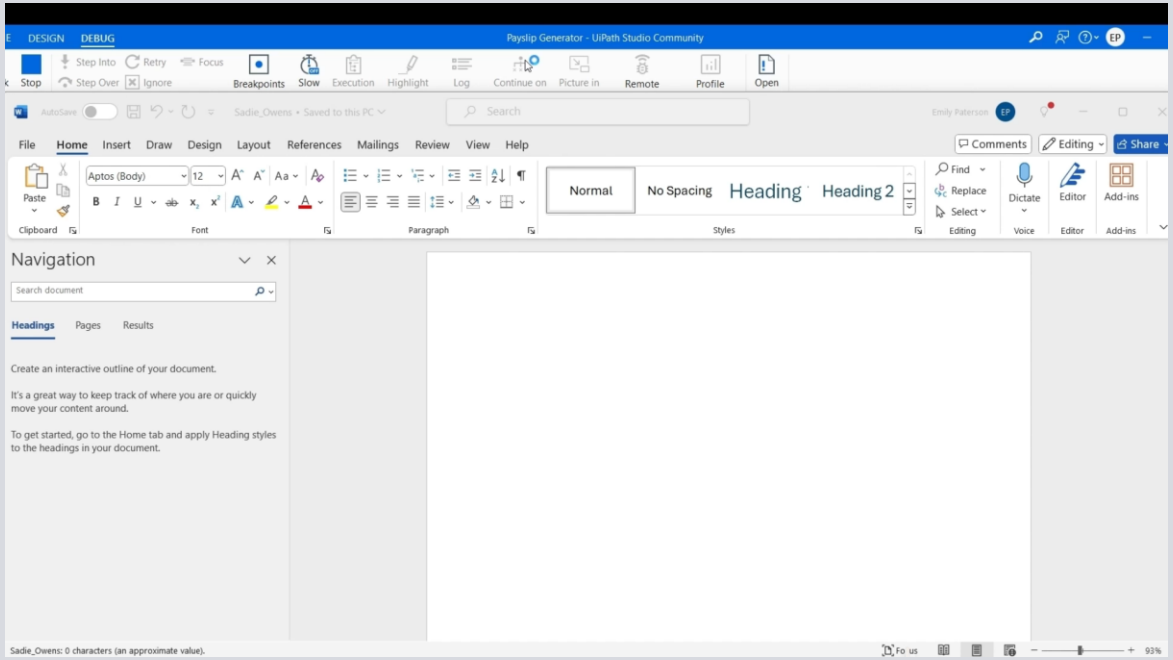
GROSS PAY : $2200, **TOTAL DEDUCTIONS**: $556.77

NET PAY : $1643.23

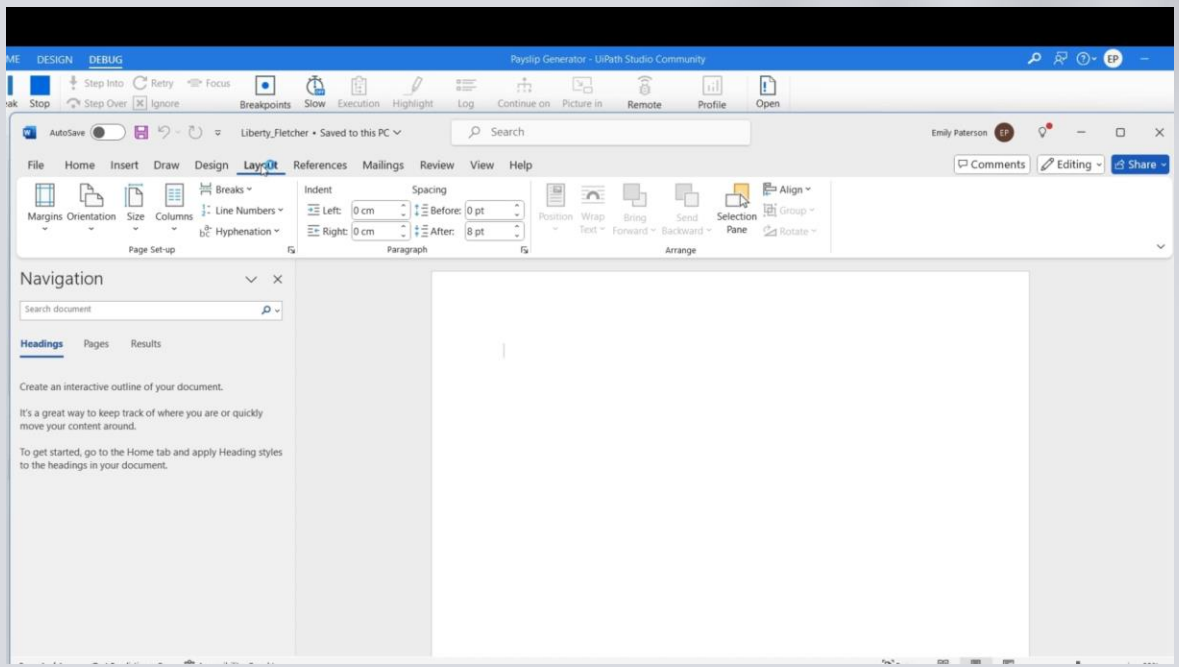
US Pay slip

Videos of process working: (Actions in video are being completed by the bot)

UK Payslip

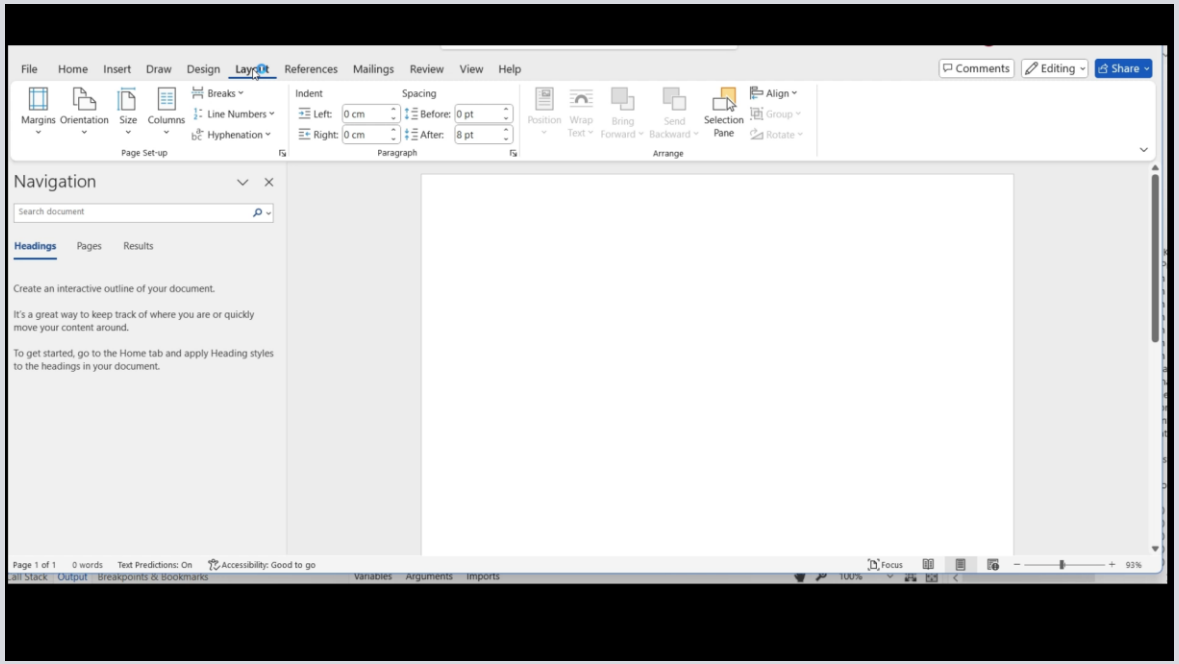


US Payslip

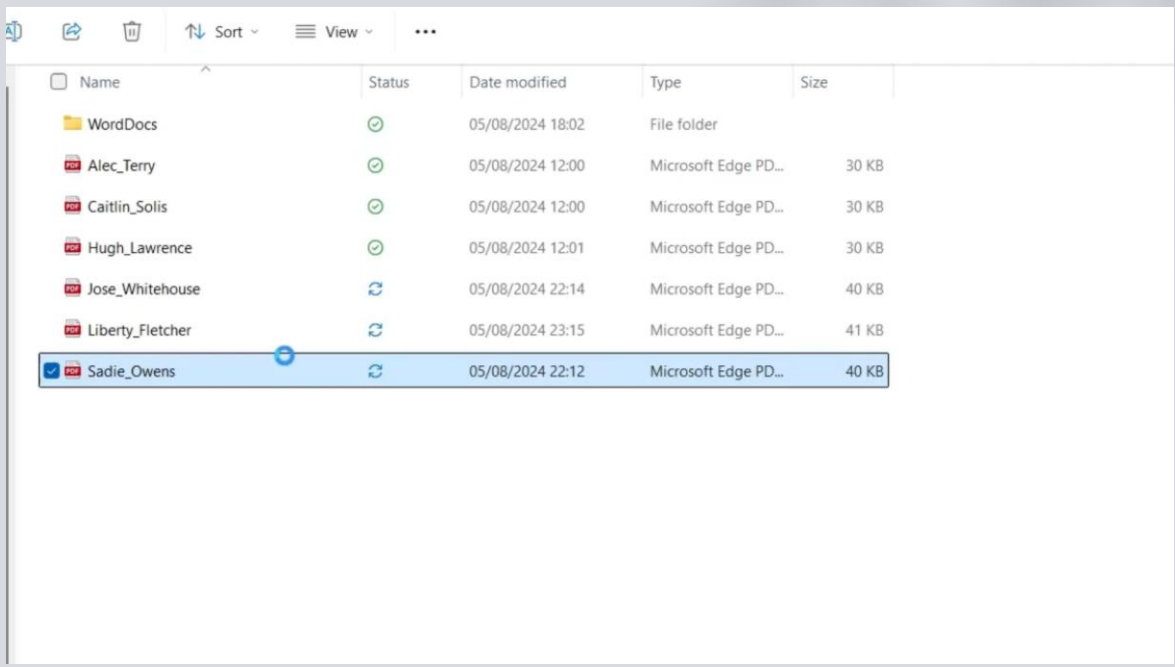


Videos of process working: (Actions in video are being completed by the bot)

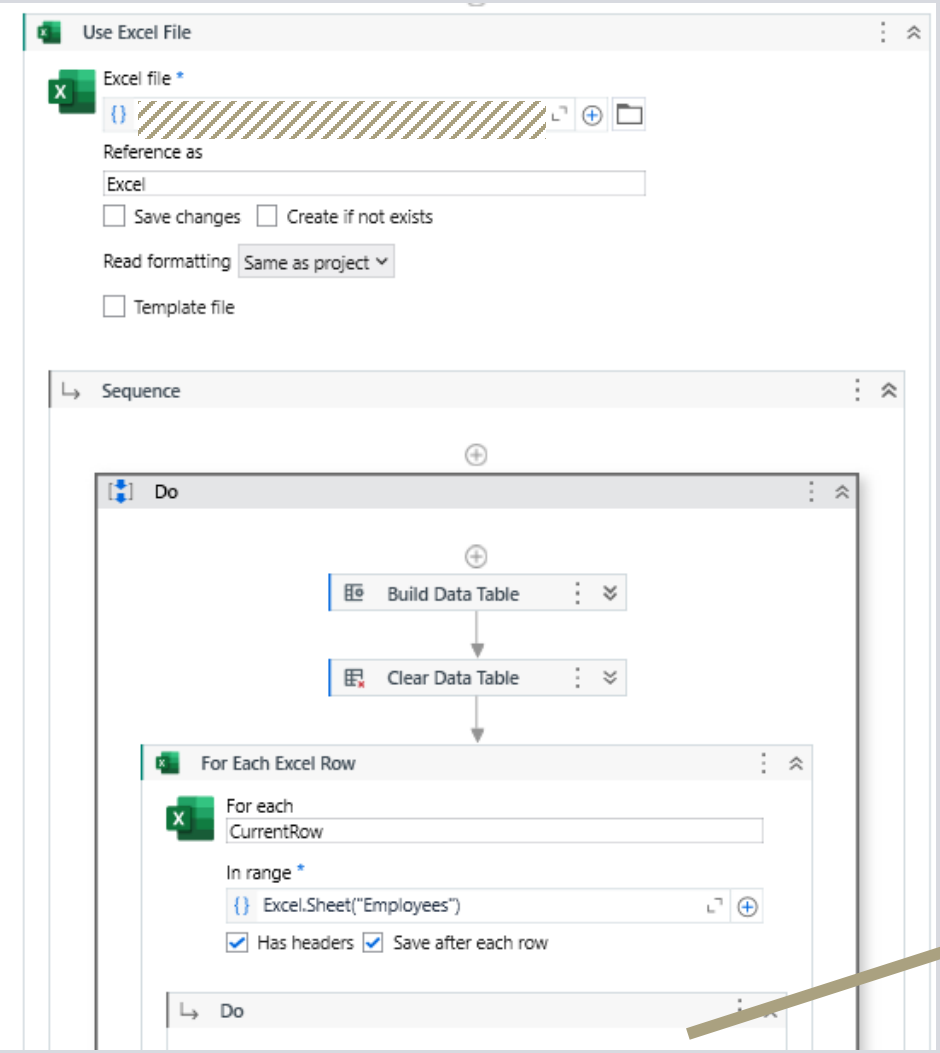
European Payslip



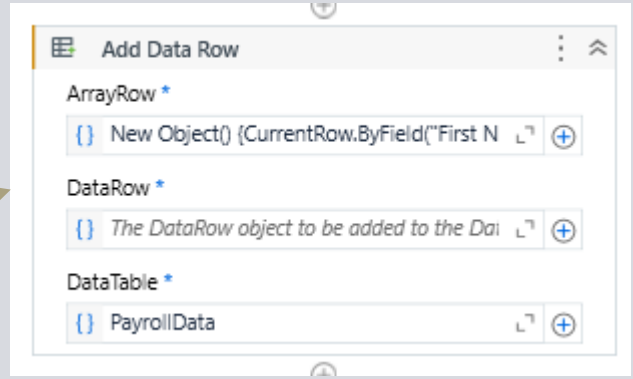
Emailing/PDFs



Step 1: Converting Excel data to Data Table

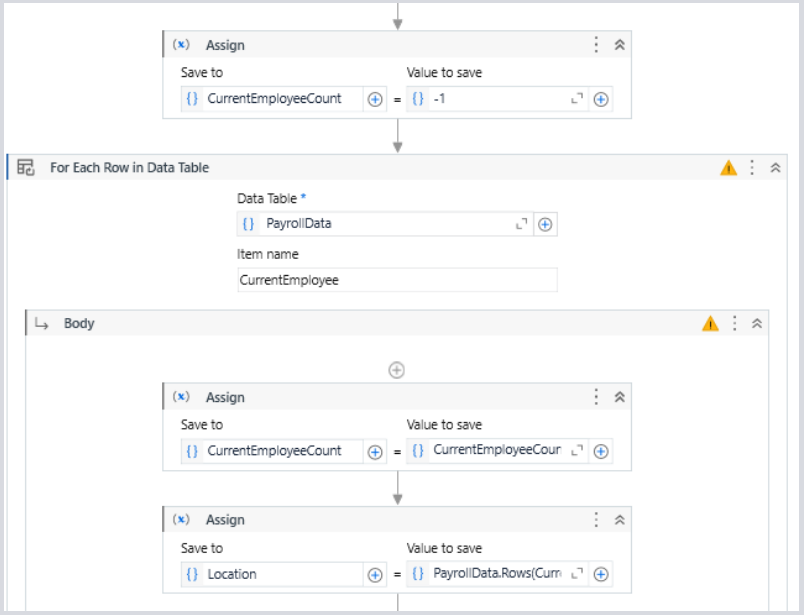


- *Use the excel file containing the employee data*
- *Build a Data Table in UiPath to store this data with the same columns as the excel file and initially clear it*
- *Then for each row in the excel file (each employee) add a data row to the table*
- *The data row goes through each column in the excel sheet and appends that data to the row*

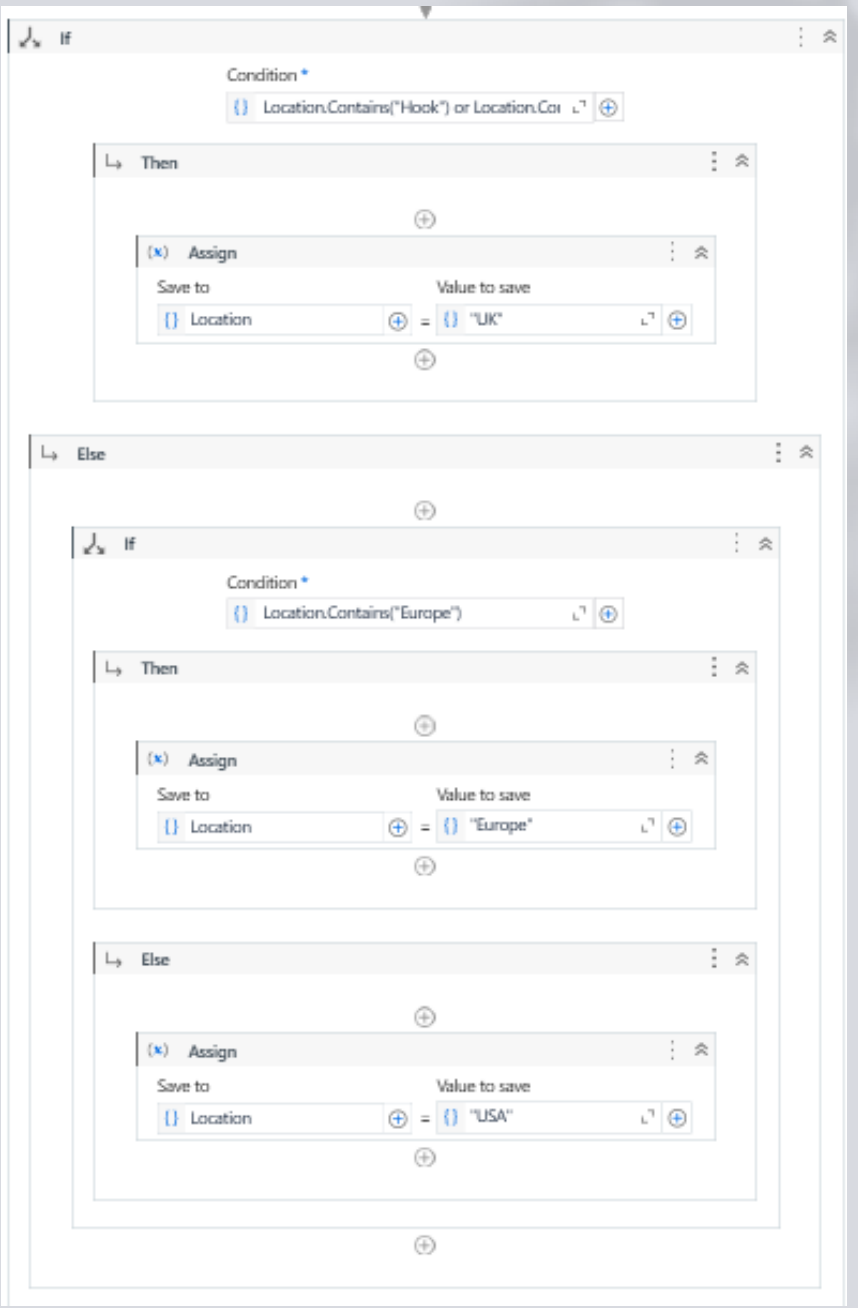


Step 2: Determine the location of the employee

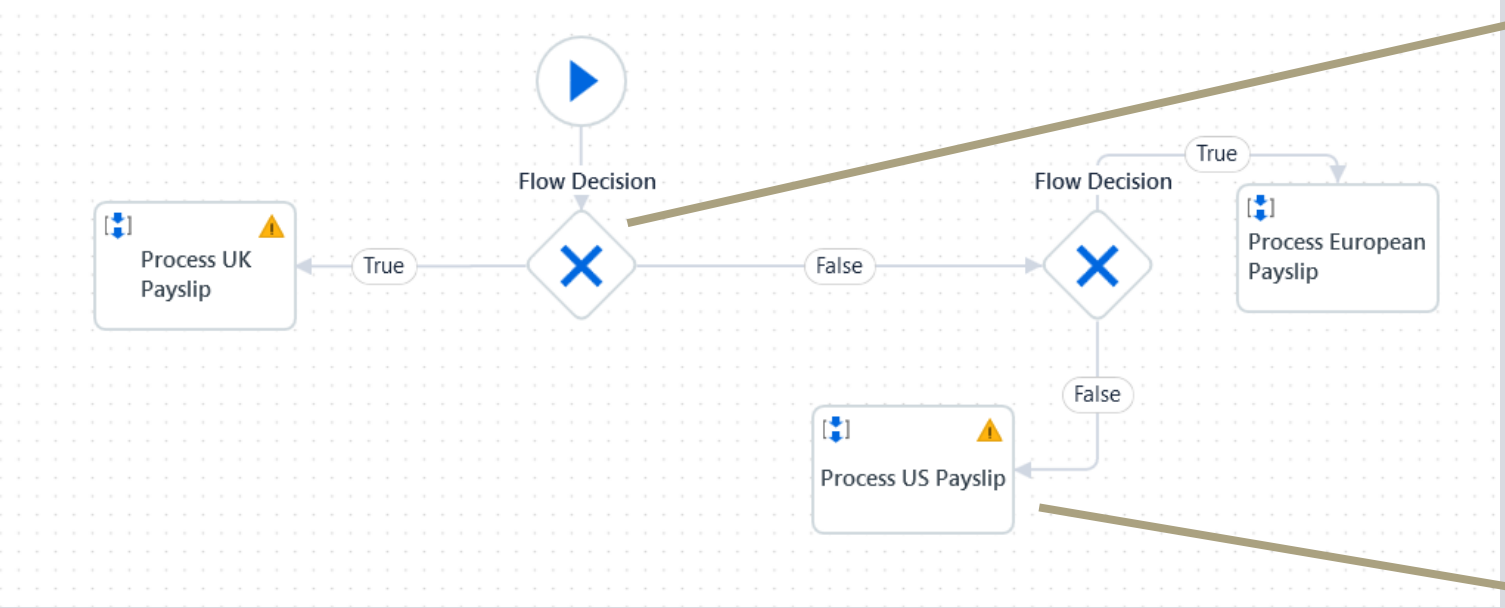
- *Set a variable called CurrentEmployeeCount to keep track of which row we are on*
- *Then for each row in the Data Table, Increment Current Employee Count and temporarily set 'Location' to the employee's home office name*



- *Using a number of If Statements, determine if the home office is in the UK, Europe or US*
- *This is done by checking if the home office contains the UK office name, US office name or Europe Office name*
- *After the location is determined, update the Location variable*



Step 3: Using a flowchart, route the employee to the correct payslip formatter



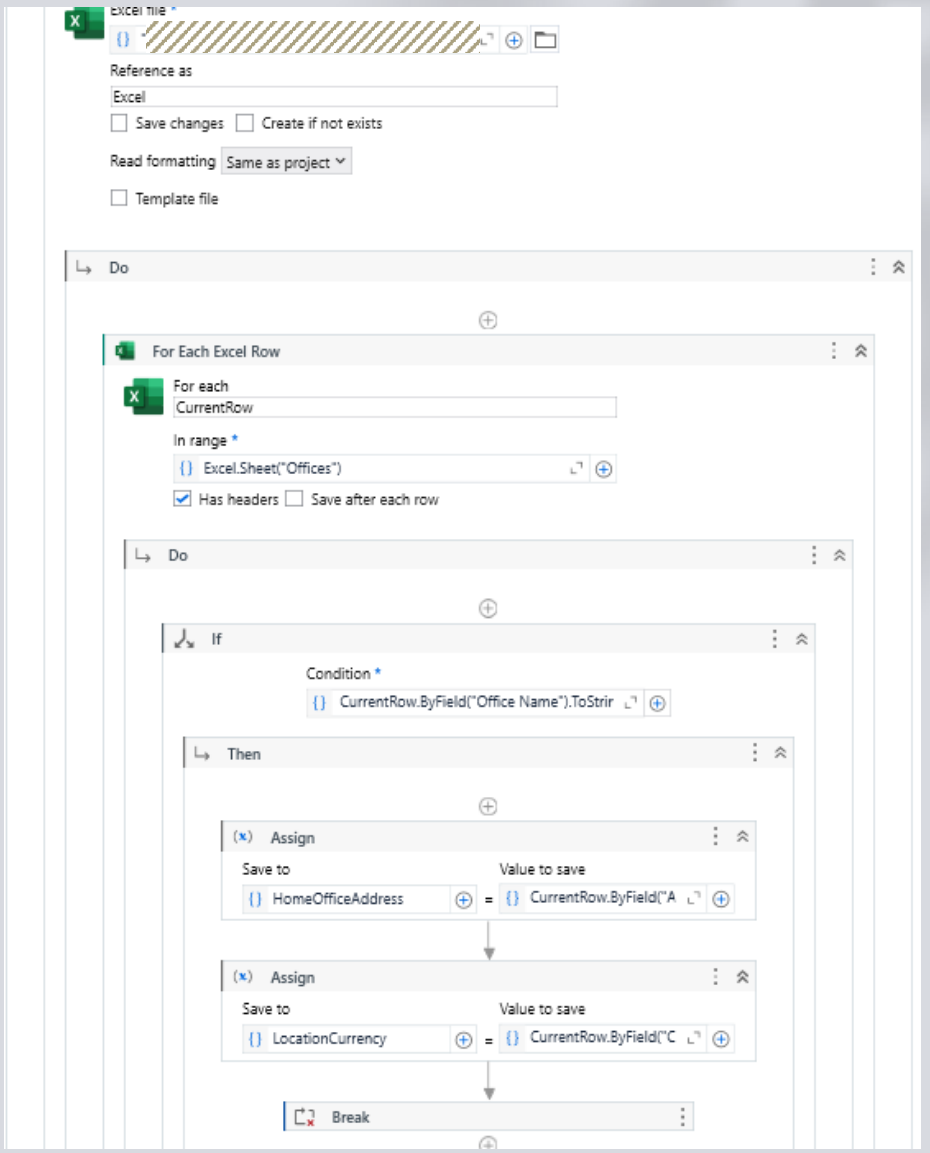
Checks if the location is equal to UK

Checks if the location is equal to Europe

For Commencing Slides:
UK & US & EU Processing: Steps 4 - 6
UK Payslip Processing: Steps 7 - 10
US Payslip Processing: Steps 11 – 14
EU Processing: Steps 15 - 17
UK & US & EU Processing: Steps 18 - 22

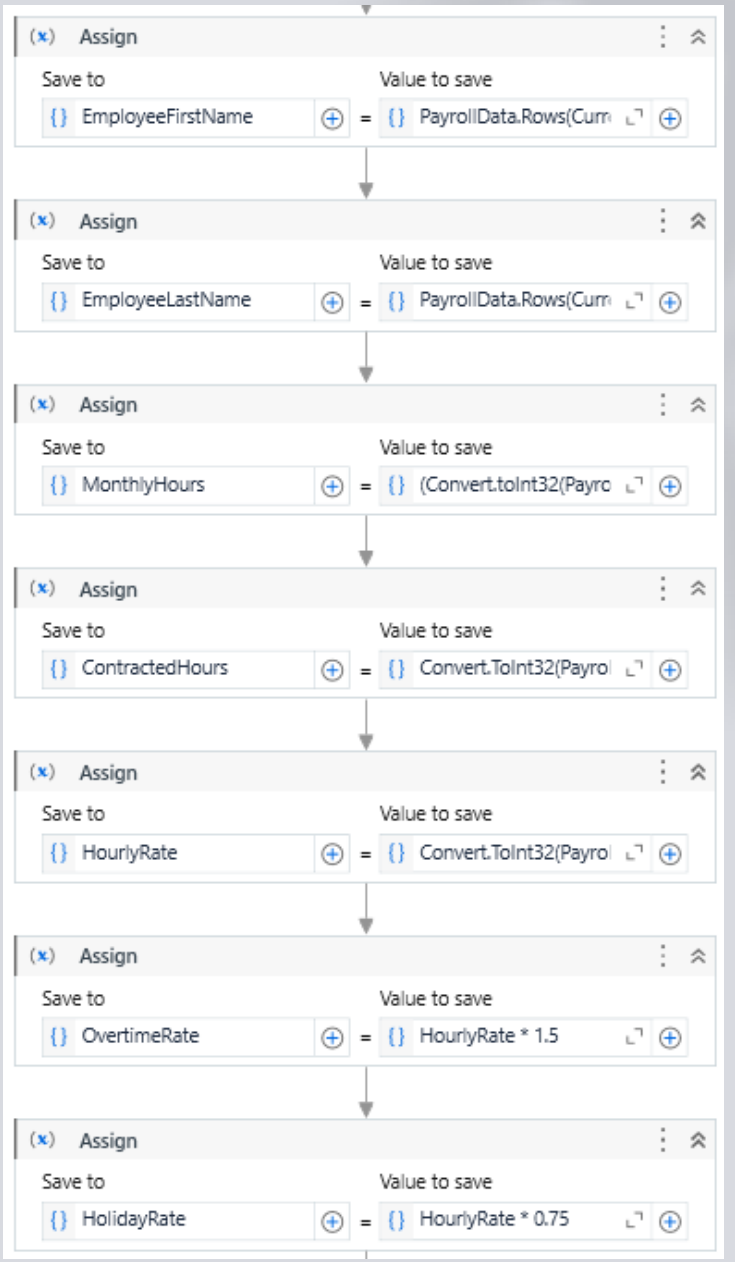
Step 4 (UK & US & EU): Get Home office address & Currency

- *Goes back to the original excel file but uses the Offices sheet rather than the employee sheet*
- *Then goes through each of the rows and checks if the office name is equal to the employee's home office name*
- *If it is equal, then set the home office address to the combination of the Address1, Address2, City and Postcode columns*
- *Also set the location currency*
- *Then break the loop as home office has been found*

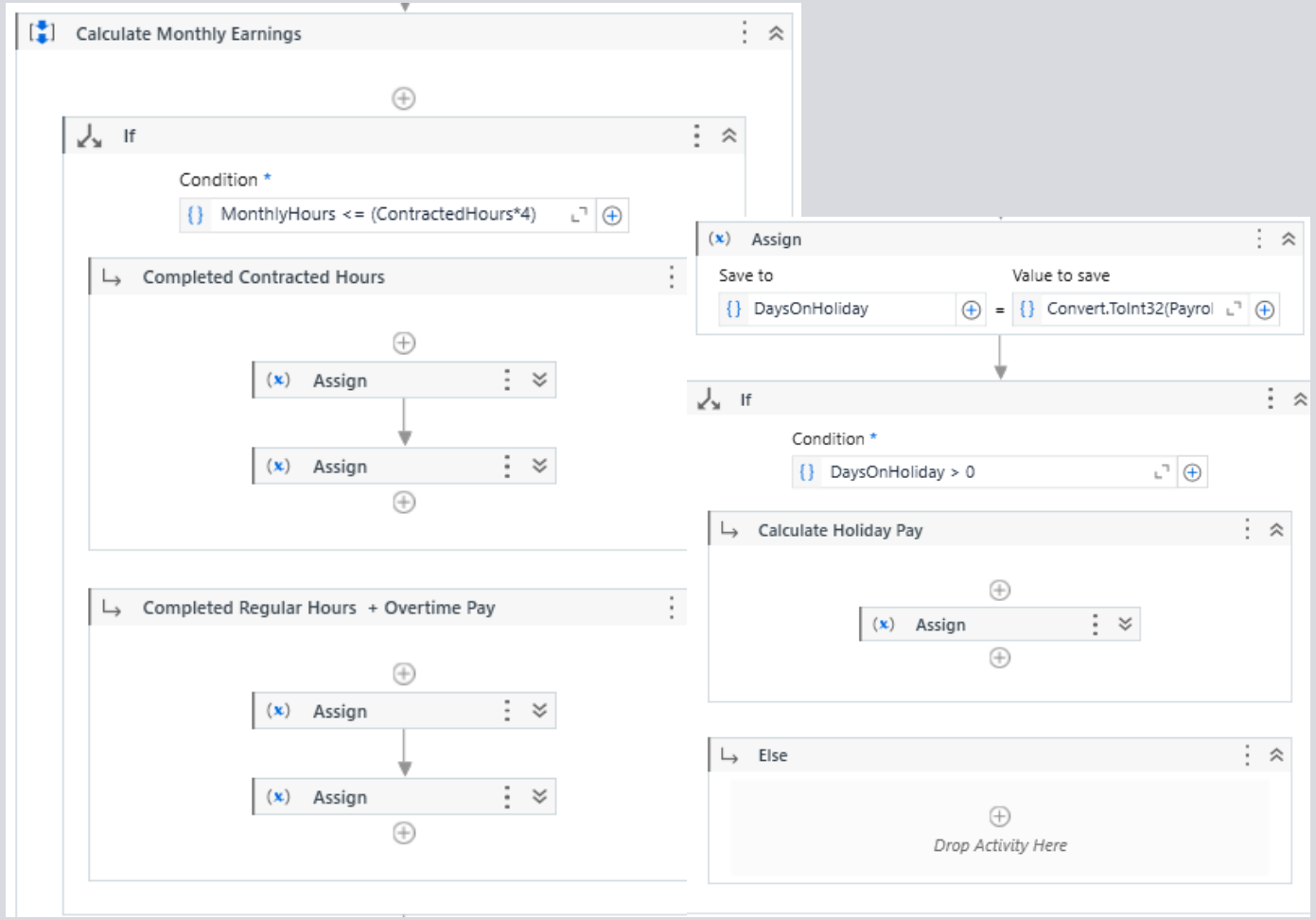


Step 5 (UK & US & EU): Get Employee Details

- *For the static values that are required on the payslip and are needed for further calculations – set a variable equal to these values*
- *Variables Required:*
  - **Employee First & Last Name**
  - **Monthly Hours (Combination of the 4 weekly hours fields)**
  - **Contracted Hours (Weekly contracted hours)**
  - **Hourly Rate**
  - **Overtime Rate (1.5 * Hourly Rate)**
  - **Holiday Rate (0.75 * Hourly Rate)**

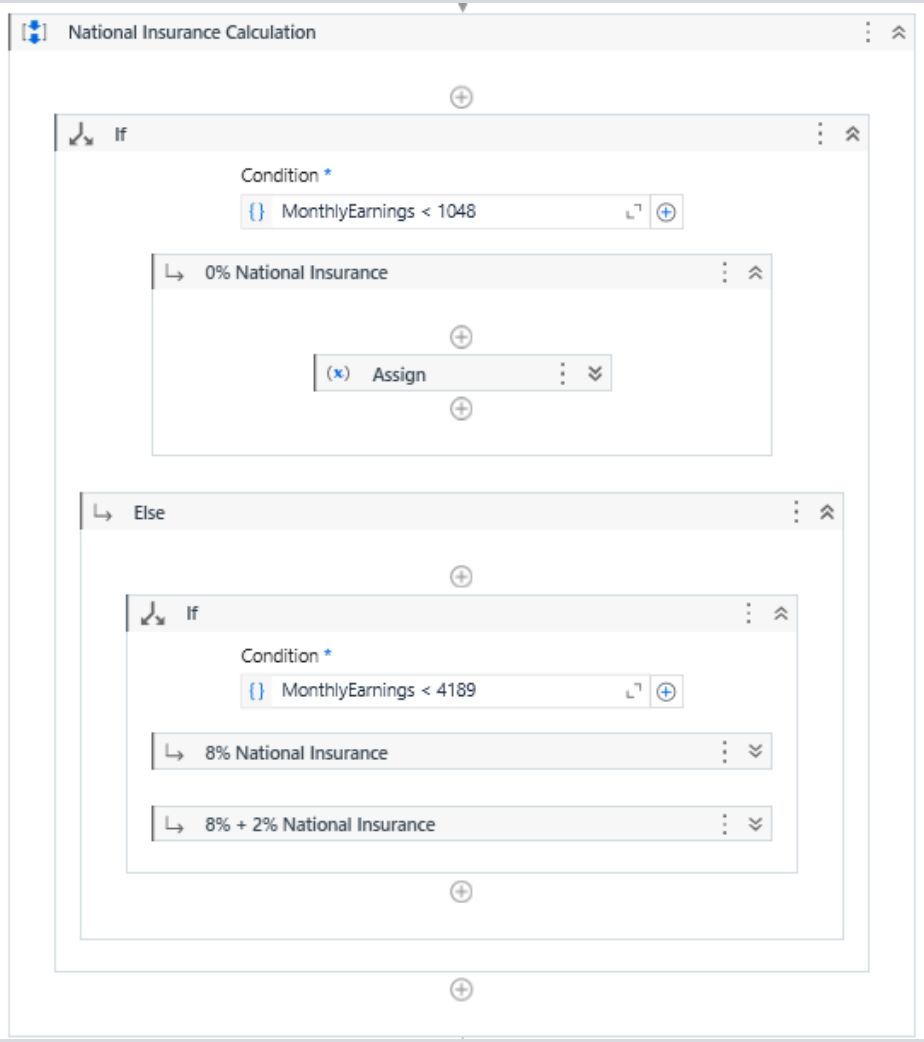


Step 6 (UK & US & EU): Calculate Monthly Earnings



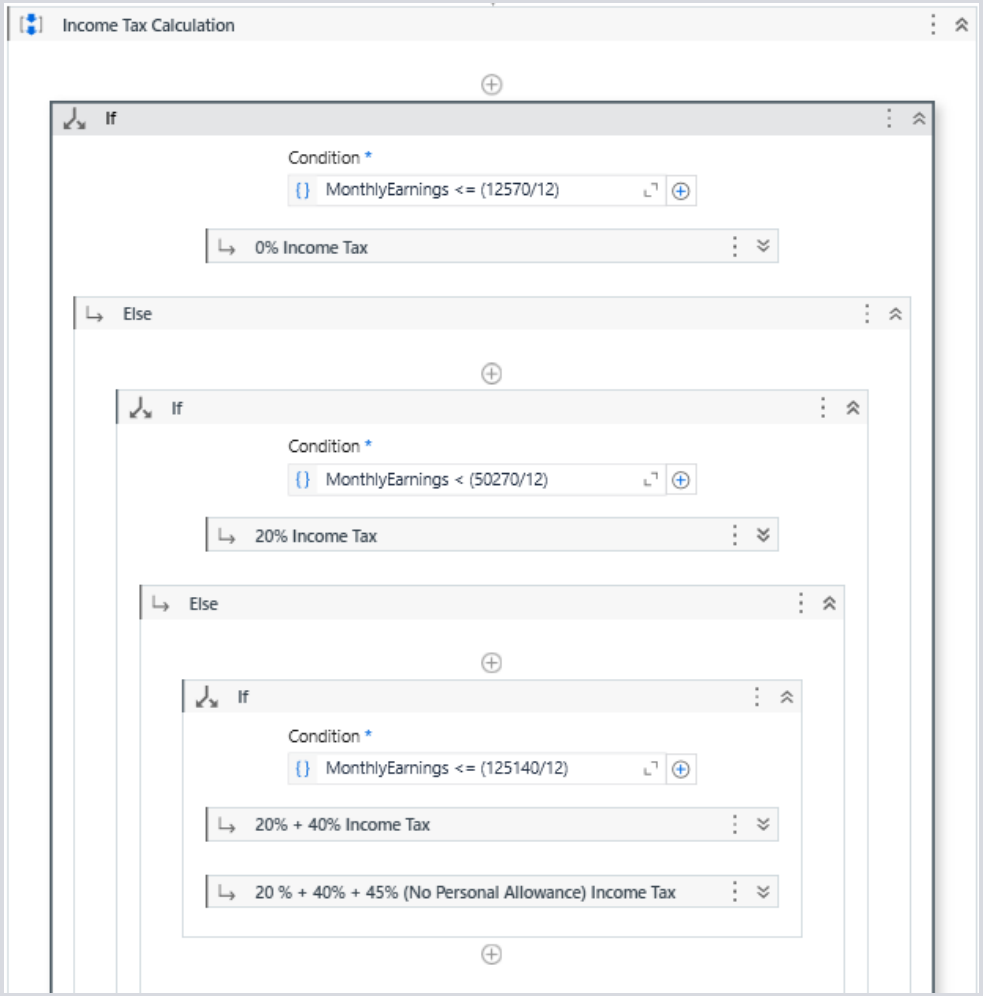
- *Check if the employee has worked overtime (more than their monthly contracted hours)*
- *If they have not, then just multiply their hours by the hourly rate*
- *Otherwise, multiply their contracted hours by their hourly rate and multiply their overtime hours by the overtime rate*
- *Then get the number of holiday days the employee has taken this month by combing the 4 weekly holiday day columns*
- *If the employee has taken some holiday, then add their holiday hours * holiday rate*
- *(assumes a holiday day is 8 hours)*

Step 7 (UK): Calculate National Insurance



- *Using the government stated brackets, calculate the employee's national insurance contribution*
  - **Up to £1047 – 0% NI**
  - **£1048 – £4188 – 8% NI**
  - **Over £4189 – 8% on £1048 - £4188 and the rest is 2%**

Step 8 (UK): Calculate Income Tax

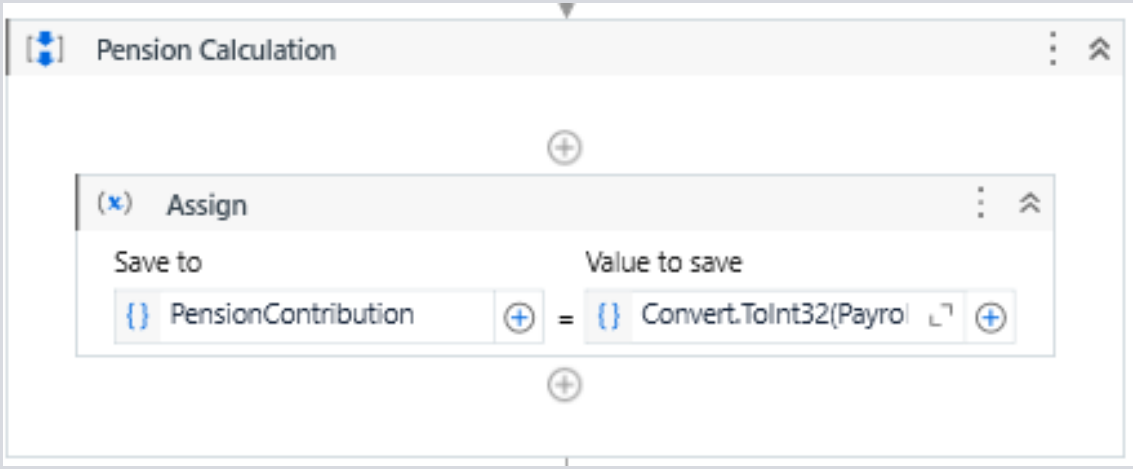


- *Calculate the employee's income tax level*

Band	Taxable income	Tax rate
Personal Allowance	Up to £12,570	0%
Basic rate	£12,571 to £50,270	20%
Higher rate	£50,271 to £125,140	40%
Additional rate	over £125,140	45%

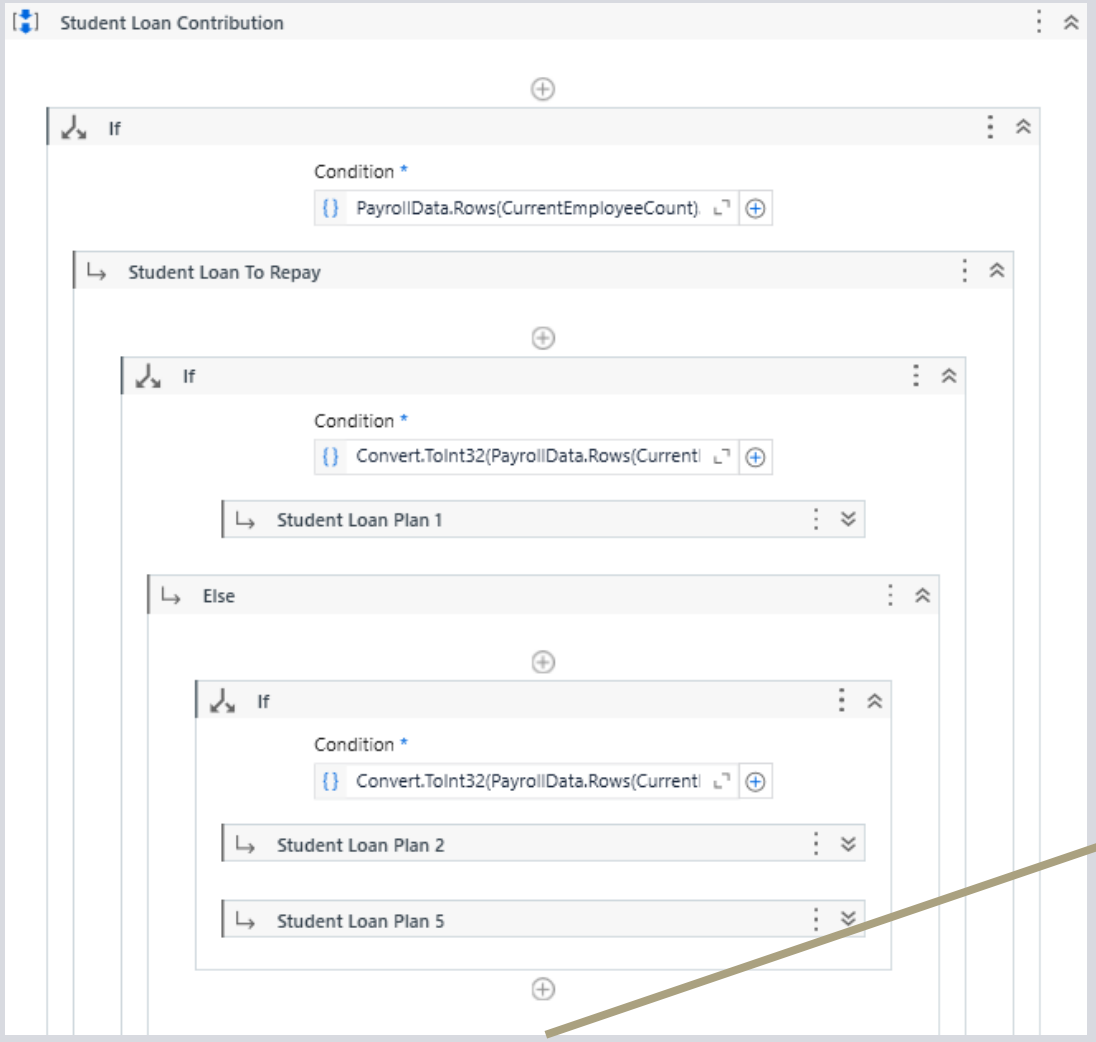
You can also see the rates and bands without the Personal Allowance. You do not get a Personal Allowance on taxable income over £125,140.

Step 9 (UK): Calculate Pension Contribution

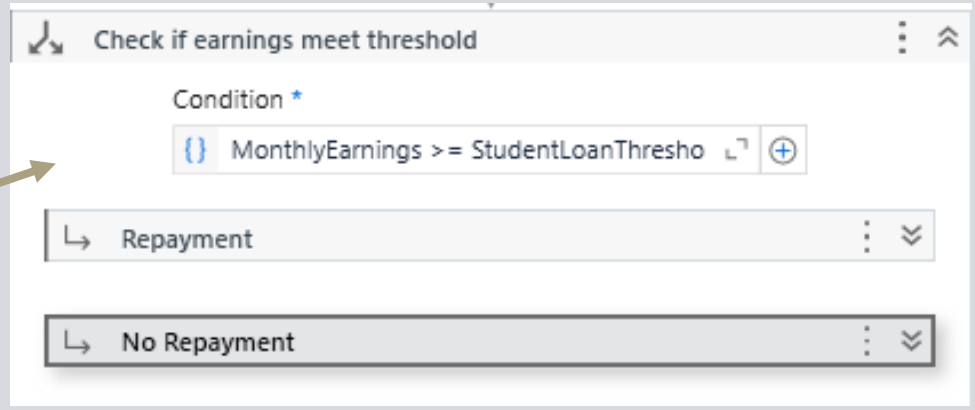


- Get the employee's pension contribution percentage from the Data Table
- Multiply that percentage by the monthly earnings

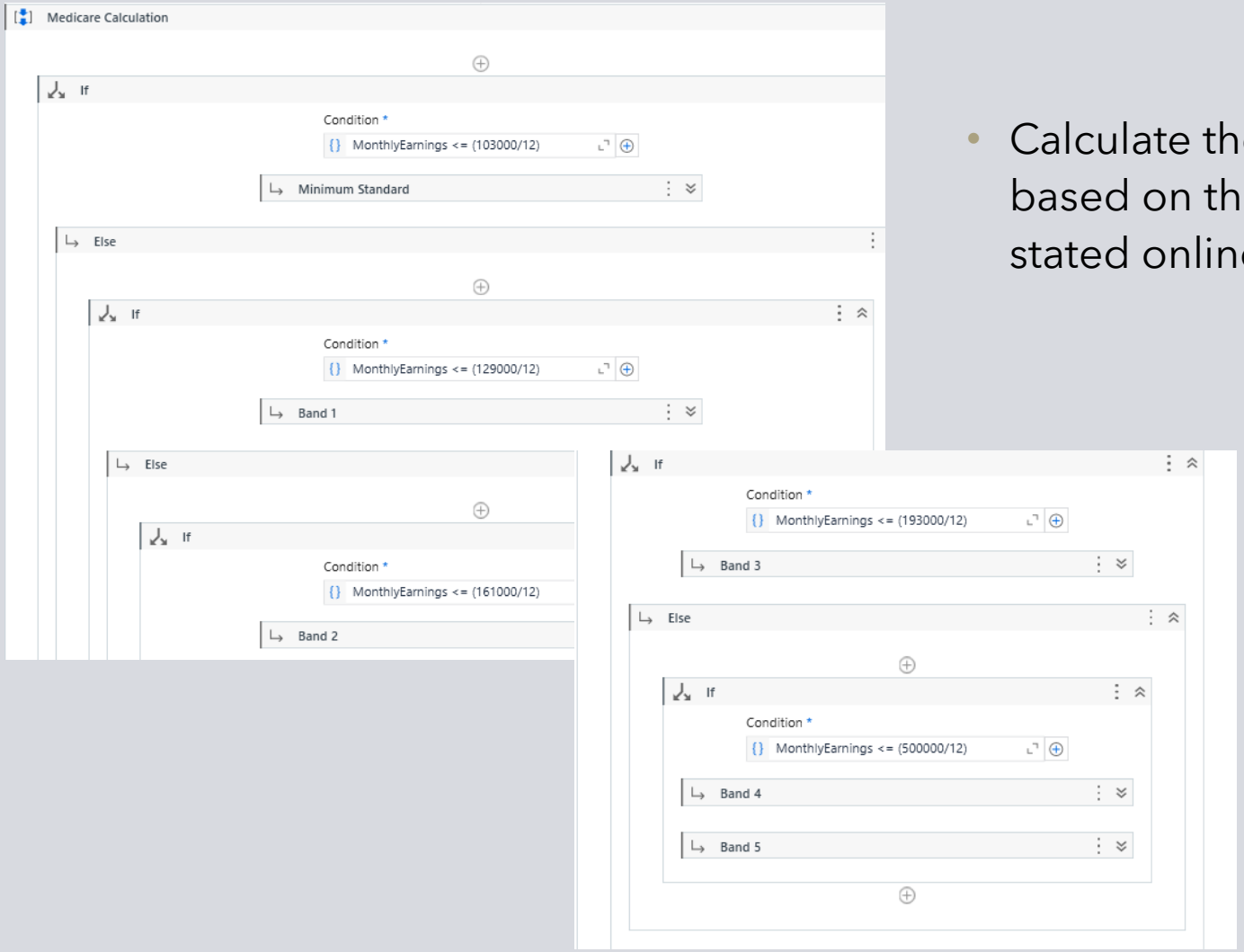
Step 10 (UK): Calculate Student Loan Contribution



- *First if statement checks if the employee has outstanding student debt*
- *If they do, then (based on their plan type) calculate the threshold at which the employee starts paying back to their debt*
- *Once the threshold is calculated, check if the employee meets that threshold, if they do then 9% of their monthly income becomes their student debt repayment*



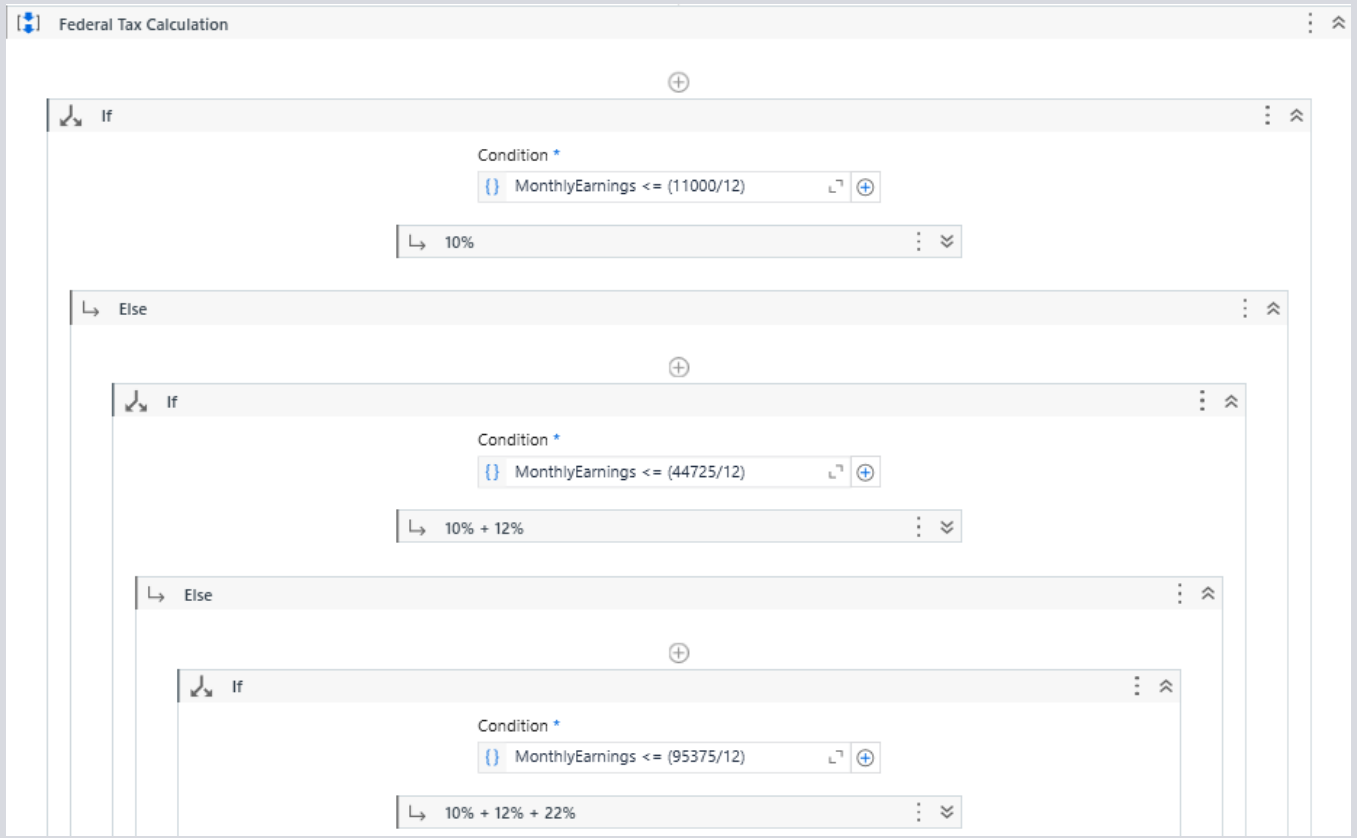
Step 11 (US): Medicare Calculation



- Calculate the employee's Medicare tax based on their monthly earnings & the stated online brackets for payment

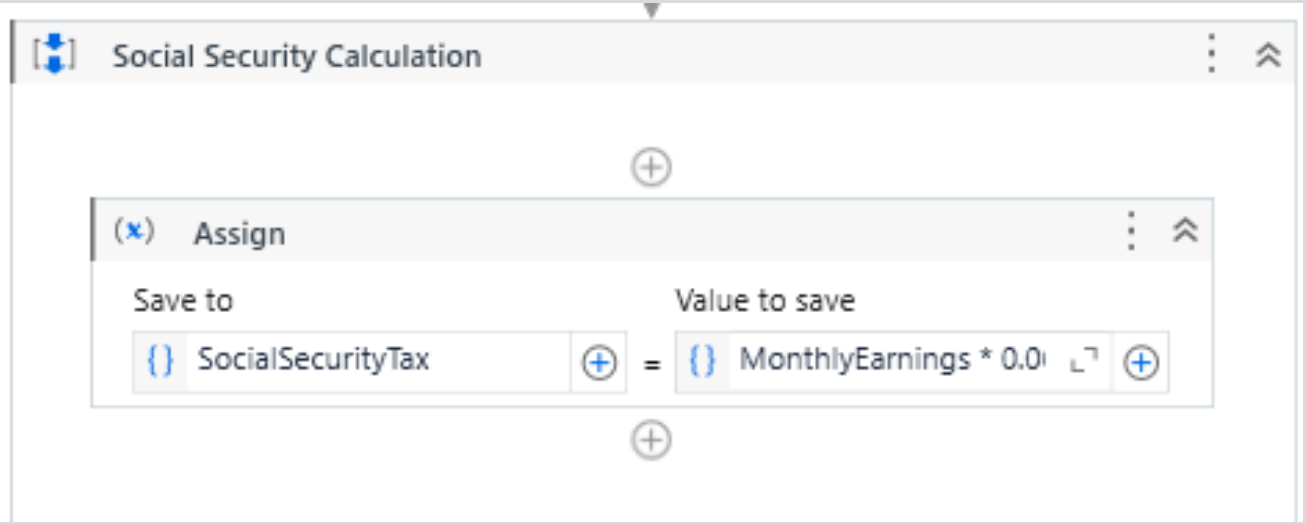
If your yearly income in 2022 (for what you pay in 2024) was			You pay each month (in 2024)
File individual tax return	File joint tax return	File married & separate tax return	
$103,000 or less	$206,000 or less	$103,000 or less	$174.70
above $103,000 up to $129,000	above $206,000 up to $258,000	Not applicable	$244.60
above $129,000 up to $161,000	above $258,000 up to $322,000	Not applicable	$349.40
above $161,000 up to $193,000	above $322,000 up to $386,000	Not applicable	$454.20
above $193,000 and less than $500,000	above $386,000 and less than $750,000	above $103,000 and less than $397,000	$559.00
$500,000 or above	$750,000 or above	$397,000 or above	$594.00

Step 12 (US): Federal Income Tax



- *Calculate Federal Income tax based on monthly earnings & tax brackets*

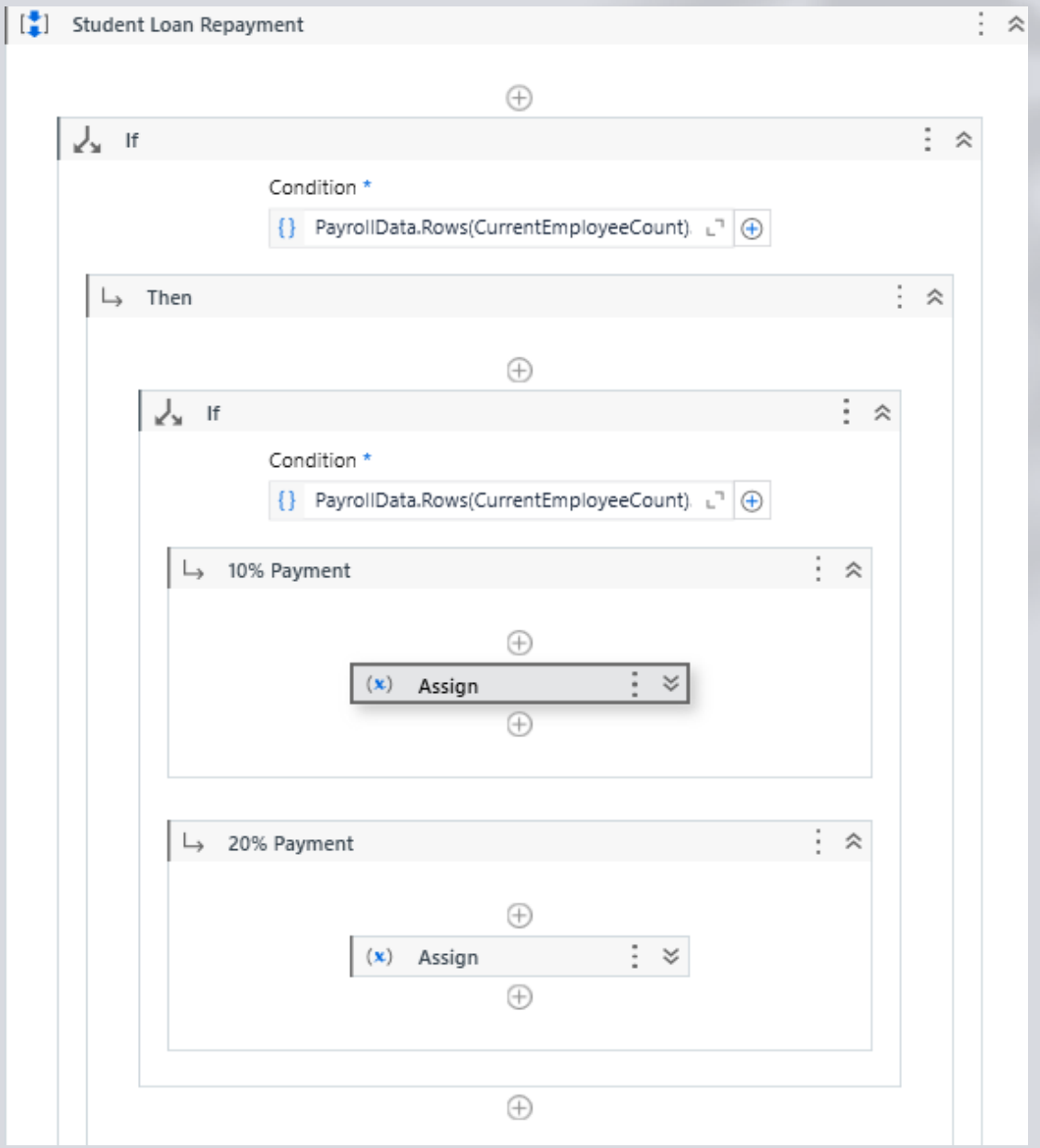
Step 13 (US): Social Security Tax



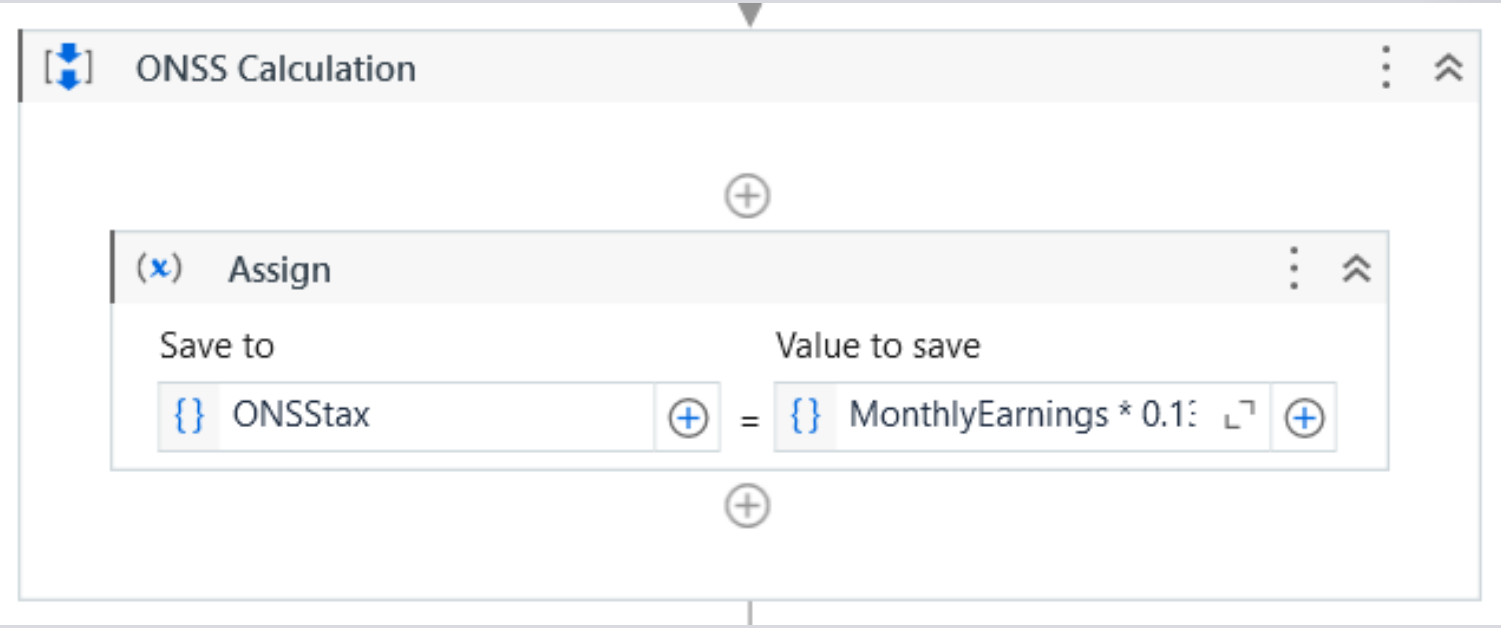
- *Calculate social security tax*
- *Flat percentage of 6.2%*

Step 14 (US): Student Debt Repayment

- *Based on the employee's plan type, calculate the percentage they pay back their student loan*

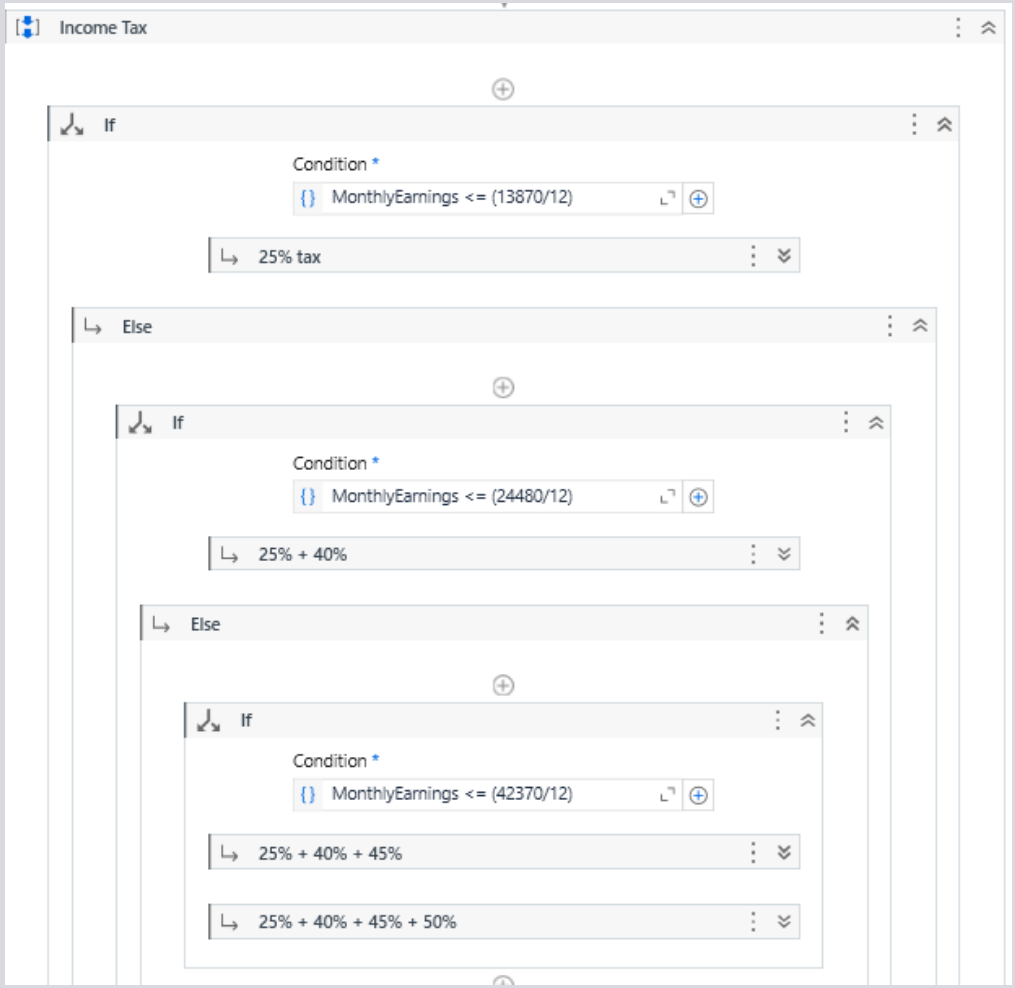


Step 15 (EU): Calculate ONSS tax



- *ONSS is the social security tax in Belgium*
- *It's set at a flat rate of 15.07% so this is calculated*

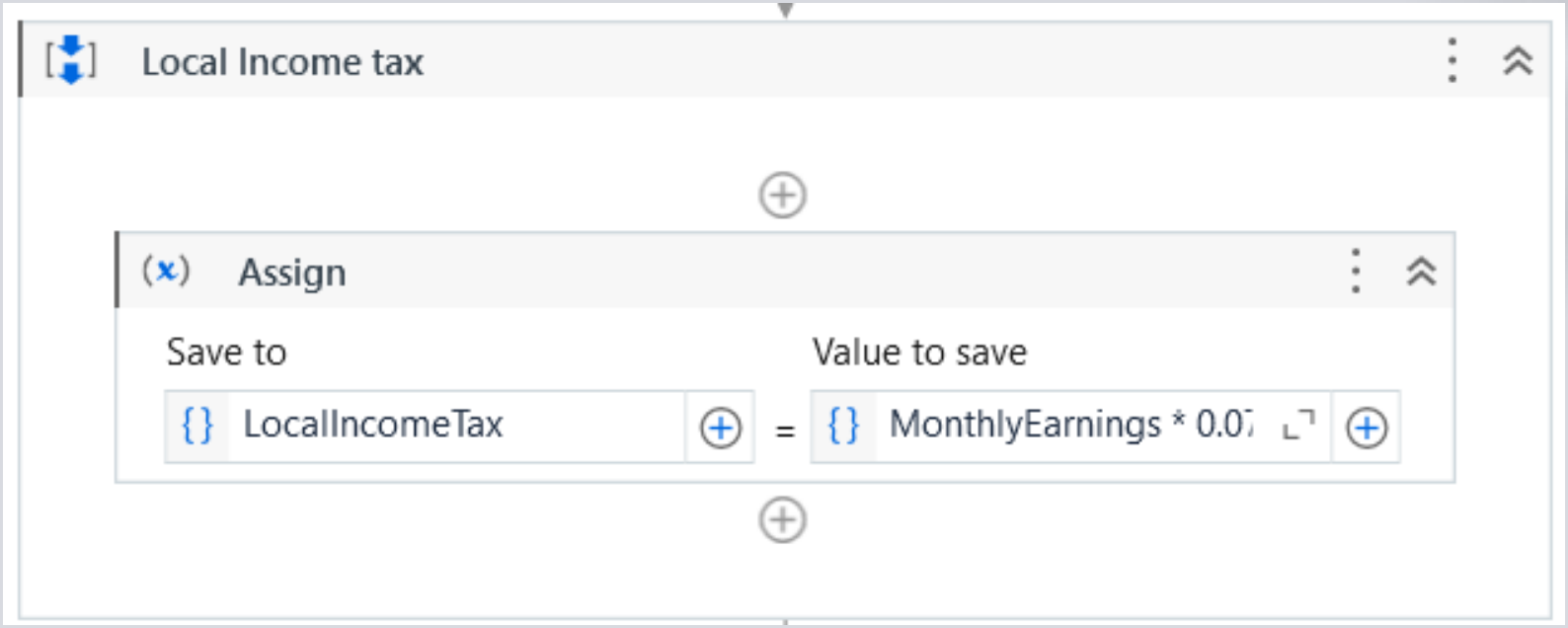
Step 16 (EU): Calculate Income Tax



- *Calculate income tax based on the income rate brackets*

Annual Income	Tax Rate
< 13,540 EUR	25%
13,540.01 EUR – 23,900 EUR	40%
23,900.01 EUR – 41,360 EUR	45%
Over 41,360.01 EUR	50%

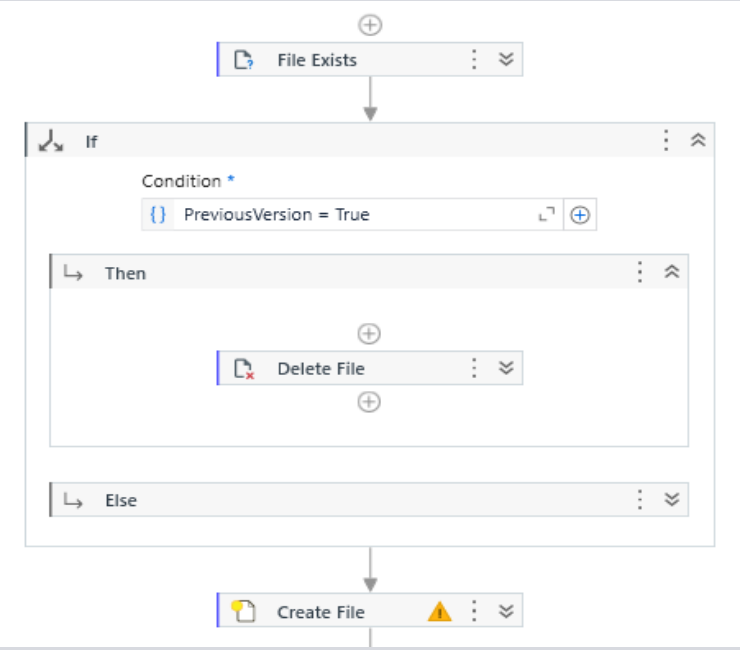
Step 17 (EU): Calculate Local Tax



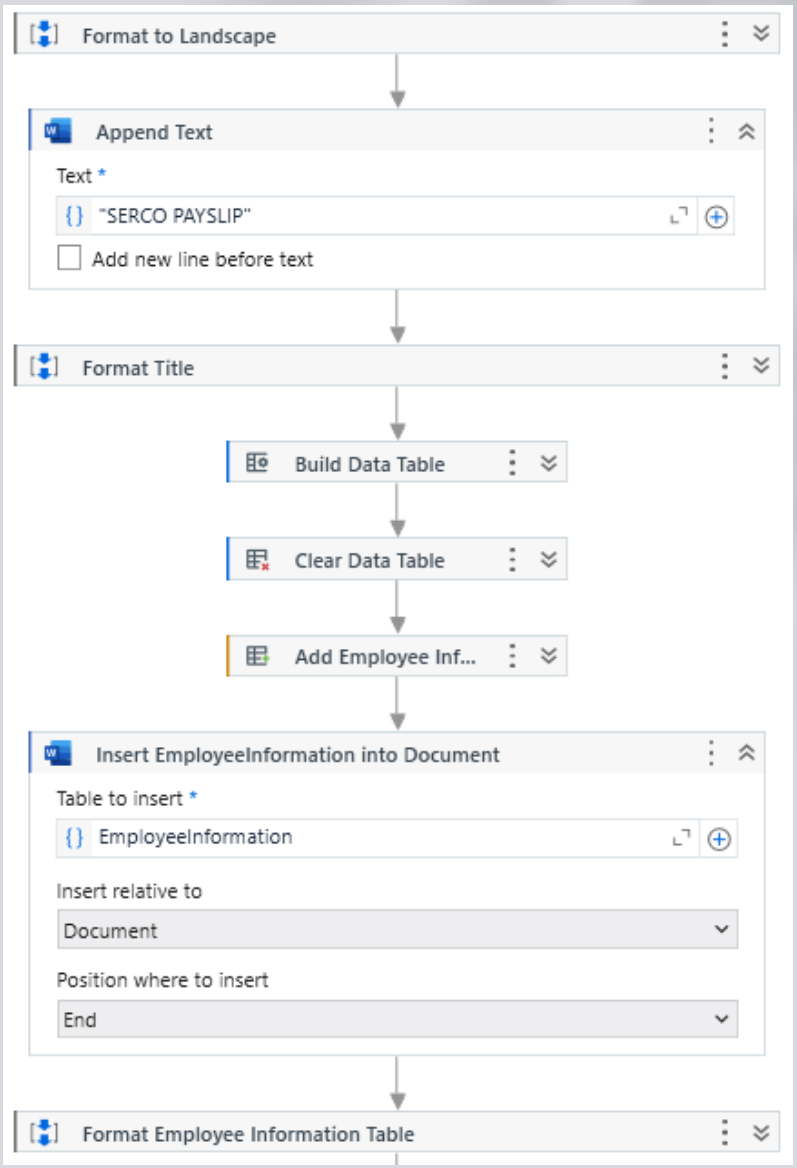
- *Calculate local income tax*
- *For this project I'm going with the average rate which is 7%*

Step 18 (UK & US & EU): Format Payslip

- *First check if a word document for the employee already exists*
- *If it does, delete the file*
- *Then create a new word document for the employee (dynamically titled with their name)*

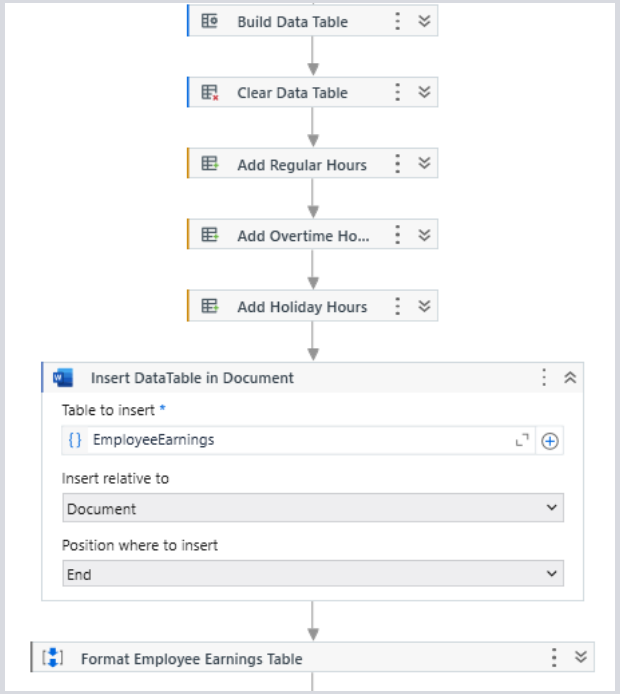


- *First format the word doc to landscape*
- *Then add the title "SERCO PAYSLIP"*
- *Then build a new data table including the employee's information*
  - **Name**
  - **Address**
  - **NI/Social Security number**
  - **Etc...**
- *Then add that table to the document and format it*

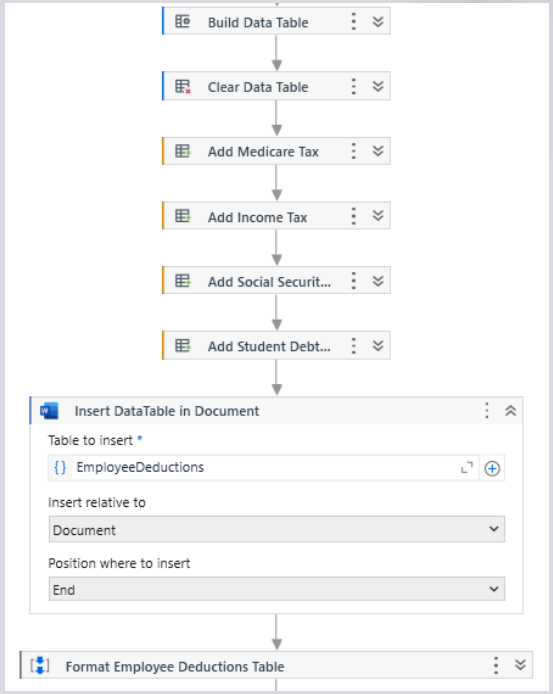


Step 19 (UK & US & EU): Format Payslip

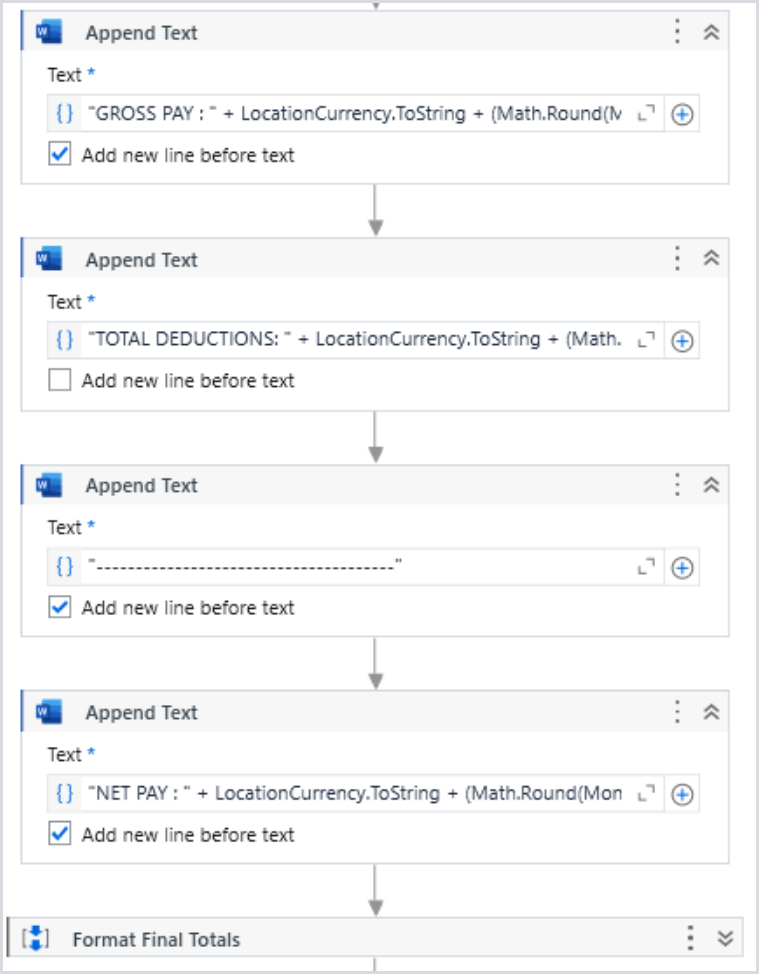
- *Create a new data table for their earnings*
- *Add their regular, overtime and holiday hours to the table*
- *Format the table*



- *Create a new data table for their deductions*
- *Add all relevant deductions based on location*
- *Format the table*

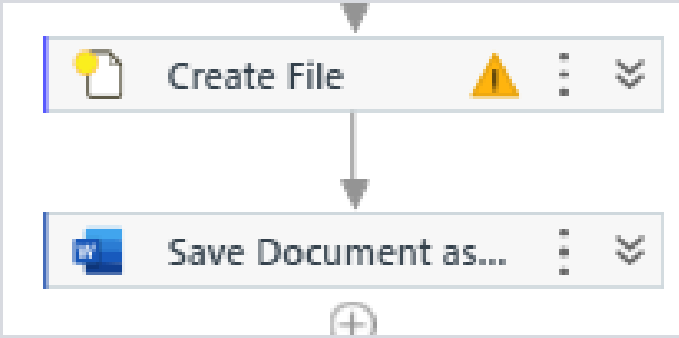


Step 20 (UK & US & EU): Format Payslip & Save as PDF

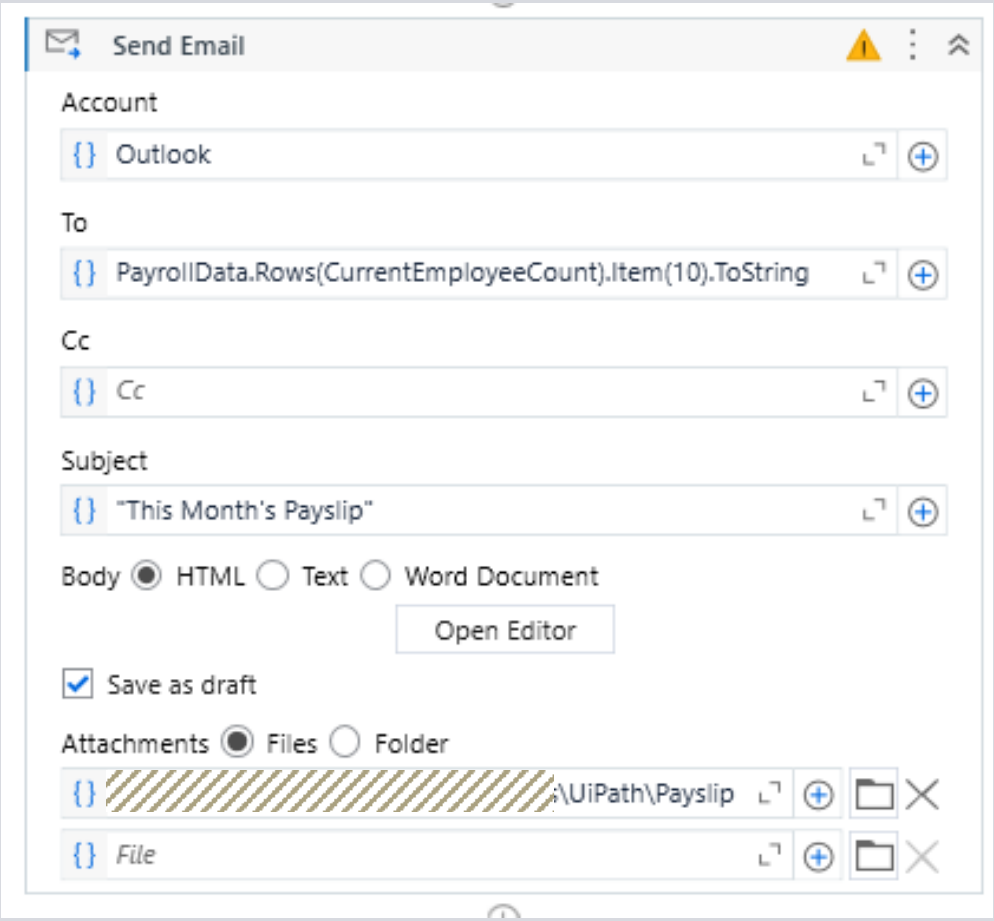


- *Add to the bottom of the document their Gross Pay, Total Deductions and Net Pay*
- *Then format this text*

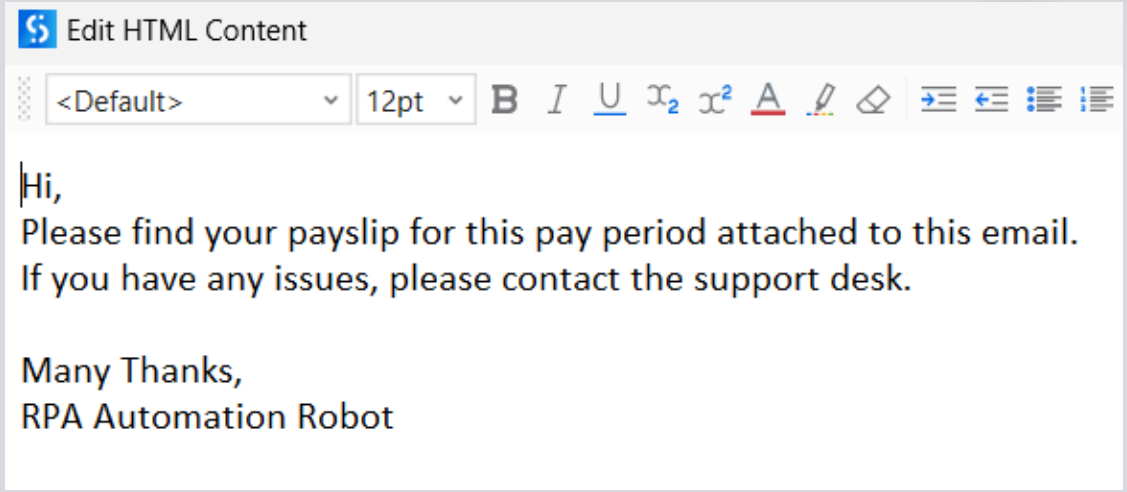
- *Then create a new PDF file titled with the employee's name*
- *Save the document to this pdf*



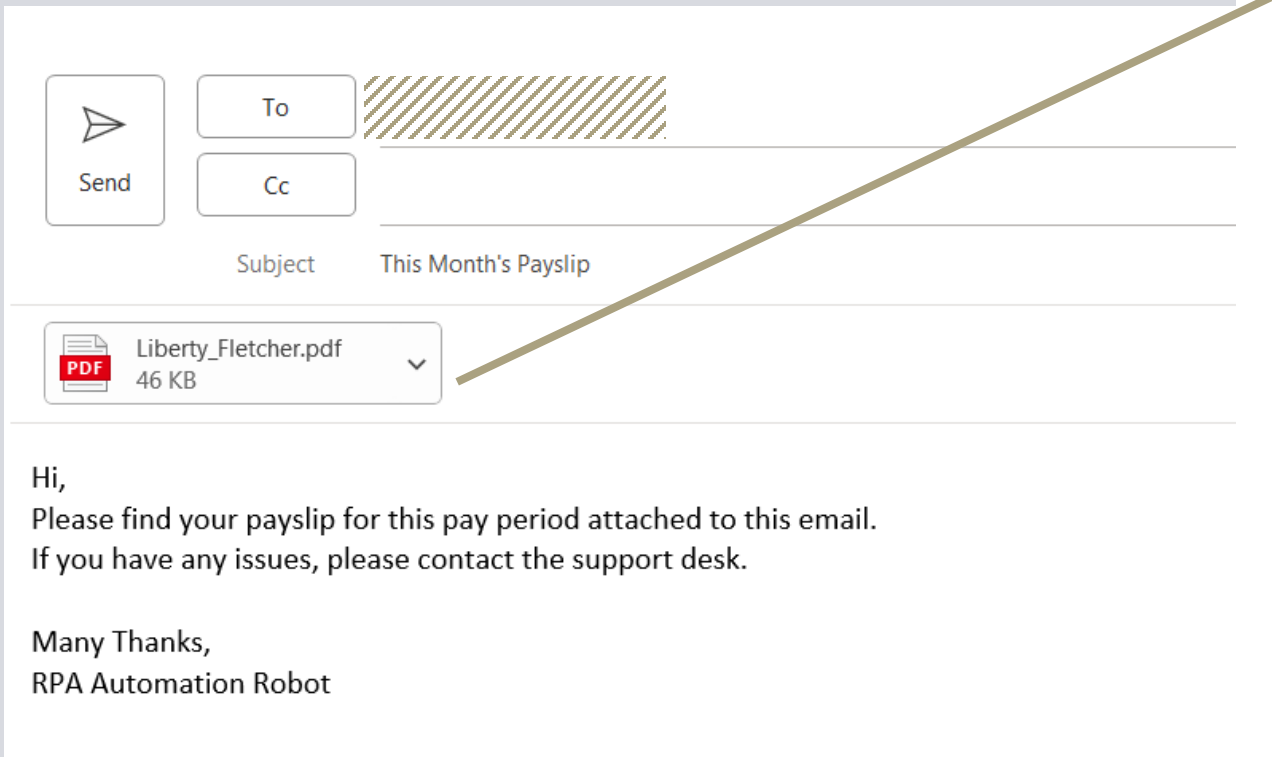
Step 21 (UK & US & EU): Email the Payslip to the employee



- *Send the email using outlook to the employee's email address*
- *Attach the payslip*



Step 22 (UK & US & EU): Final Result



SERCO PAYSLIP

EMPLOYEE INFORMATION	PAY DATE	HOME OFFICE ADDRESS	EMPLOYEE ADDRESS	NI NUMBER	EMAIL	JOB TITLE
SADIE OWENS	08/05/2024 22:10:18	16, Bartley Wood Business Park, Hook, RG27 9UY	1 Church Lane,Portsmouth,PO57 7XK	FJ375937O	sadie.owens4@serco.com	Cleaner

EARNINGS	HOURS	RATE	TOTAL
STANDARD PAY	120	14	1680
OVERTIME PAY	8	21	168
HOLIDAY PAY (3 DAYS)	24	10	252

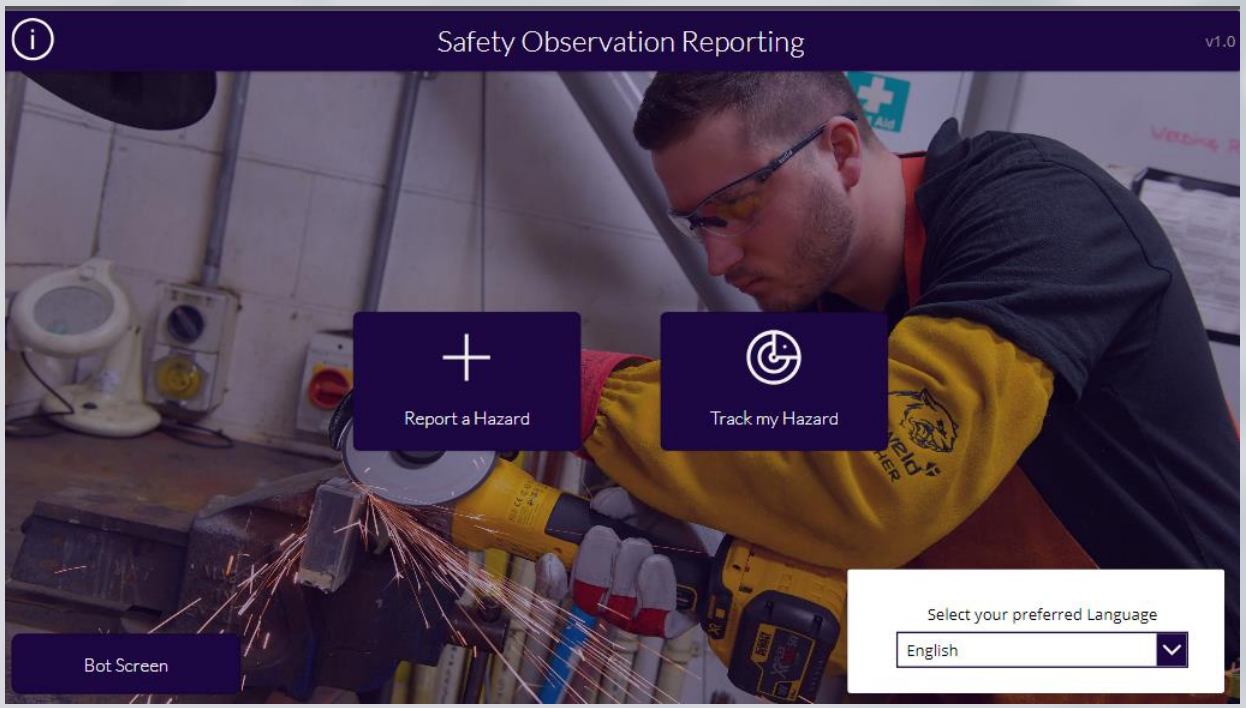
DEDUCTIONS	TOTAL
NATIONAL INSURANCE	84
INCOME TAX	210
PENSION CONTRIBUTION	105
STUDENT DEBT REPAYMENT	189

GROSS PAY : £2100, **TOTAL DEDUCTIONS**: £588.66

NET PAY : £1511.34

Safety Observations App

- **PROBLEM:** Finding an autonomous solution to adding Safety Reports to Assure from a Power App, logging all the details & dealing with any exceptions
- **CURRENT SOLUTION:** Using paper-based forms which are then manually inputted by managers on site (Inefficient & Time Consuming, also not inclusive to other languages)
- The solution I've developed in the next slides has been refined to make it as efficient as possible. By making these changes I improved the process time for 5 reports from almost 2 minutes to around 30 seconds.
- The full breakdown of the solutions is shown in Week 4 from slide 66 onwards



New Power App Developed to allow workers to digitally submit Safety Reports

Week 4 – Site Visit & Continued Development

Site Visit to the Restart Team

- Went on a site visit to the Restart Team
- This team aids people get new job opportunities through job coaching and upskilling candidates through courses in key skills such as IT.
- The site visit was focused on solutions analysis. We had several discussions with team members about new automations and improvements to current solutions!

Continued work on the performer for Safety Observations

- Continued to work on the development of the performer section of the automation process
- Moved on from extracting the data to inputting that data into Assure and creating a new report

Site visit to Restart Team

- I was very grateful to have gone on a 1-day site visit to the Restart team who aid people in getting new jobs via coaching and sessions in crucial skills such as IT.
- The site visit was extremely informative and was focused on the solutions analyst part of the RPA team. We talked to various key team members and discussed possible new automation solutions and improvements for current solutions to be implemented in further iterations.
- Discussions with team members involved going through the processes step-by-step to understand how they are implemented. By watching the process, we could see various areas that could be automated, and we discussed with the team how these automations would work so that they were well-informed. We also looked at how efficiently the current solutions were working.
- One particular solution we looked at had been previously implemented. However, it was only able to automate around 1/3 of cases. This meant the team still had to manually process around 200 data forms daily which is very time-consuming and wasteful of employee's time. By reviewing how the automation bot was working, we were able to notice that by adding one extra check into the process, we should be able to significantly reduce the number of reports the robot can successfully process.
- It was great to see how work done behind the scenes has a real-life impact!

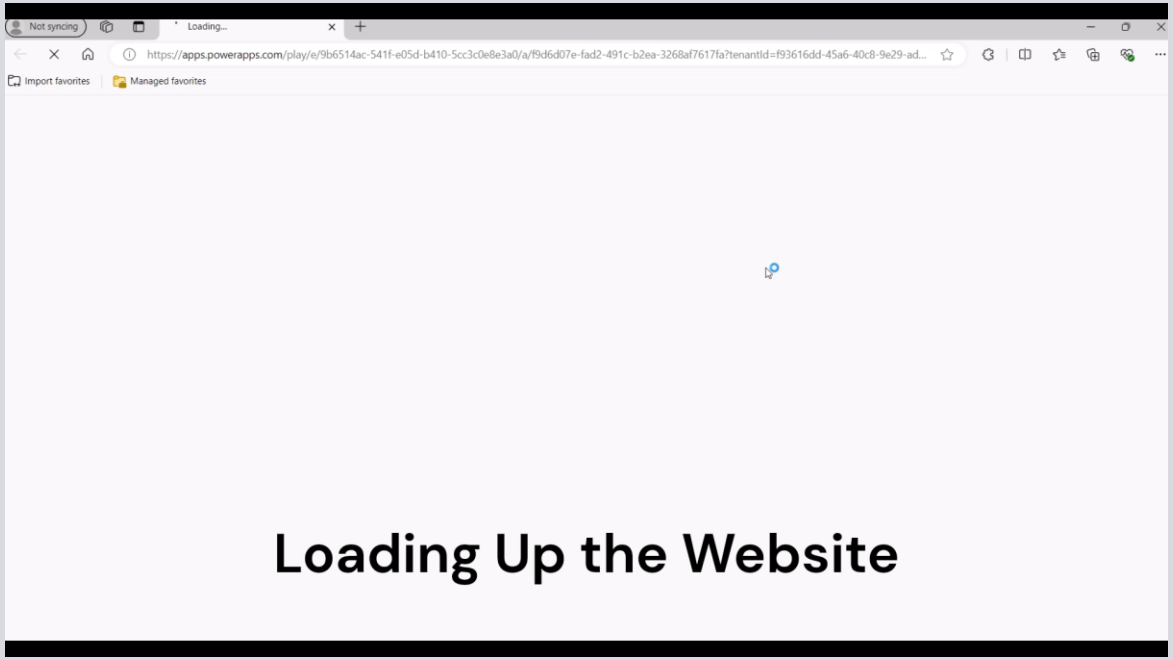
- Link for more information on the Restart Team: https://www.serco-ese.com/restart-scheme

Development of Healthcare App

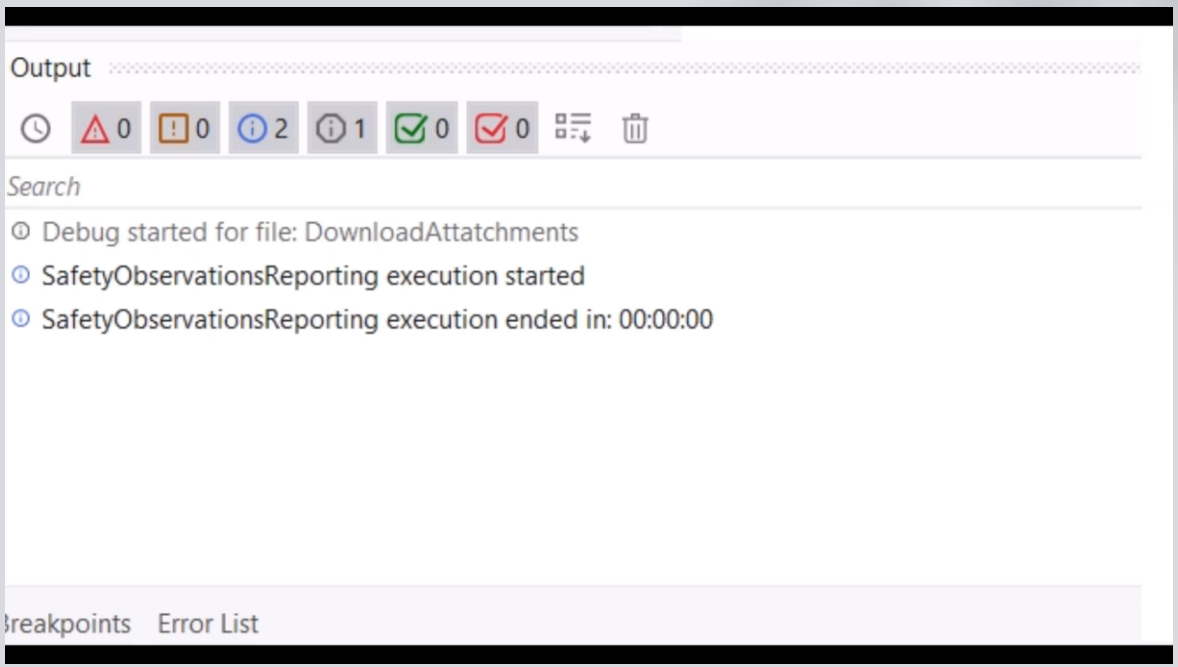
- This process involves opening the Safety Observations power app which is where users can enter a new observation
- The bot will read all the current open reports and add these to a queue
- Then one by one the bot will extract the data from the report and then add this data into the observation management system used by Serco (Assure)
- As long as no exceptions are thrown, the report can be closed and the next one is processed.
- If an exception is thrown, it's handled by the exception handler
- (This is not a complete solution due to some technical issues with the Assure platform which limited the amount I could automate at that date)

Videos of Process Working

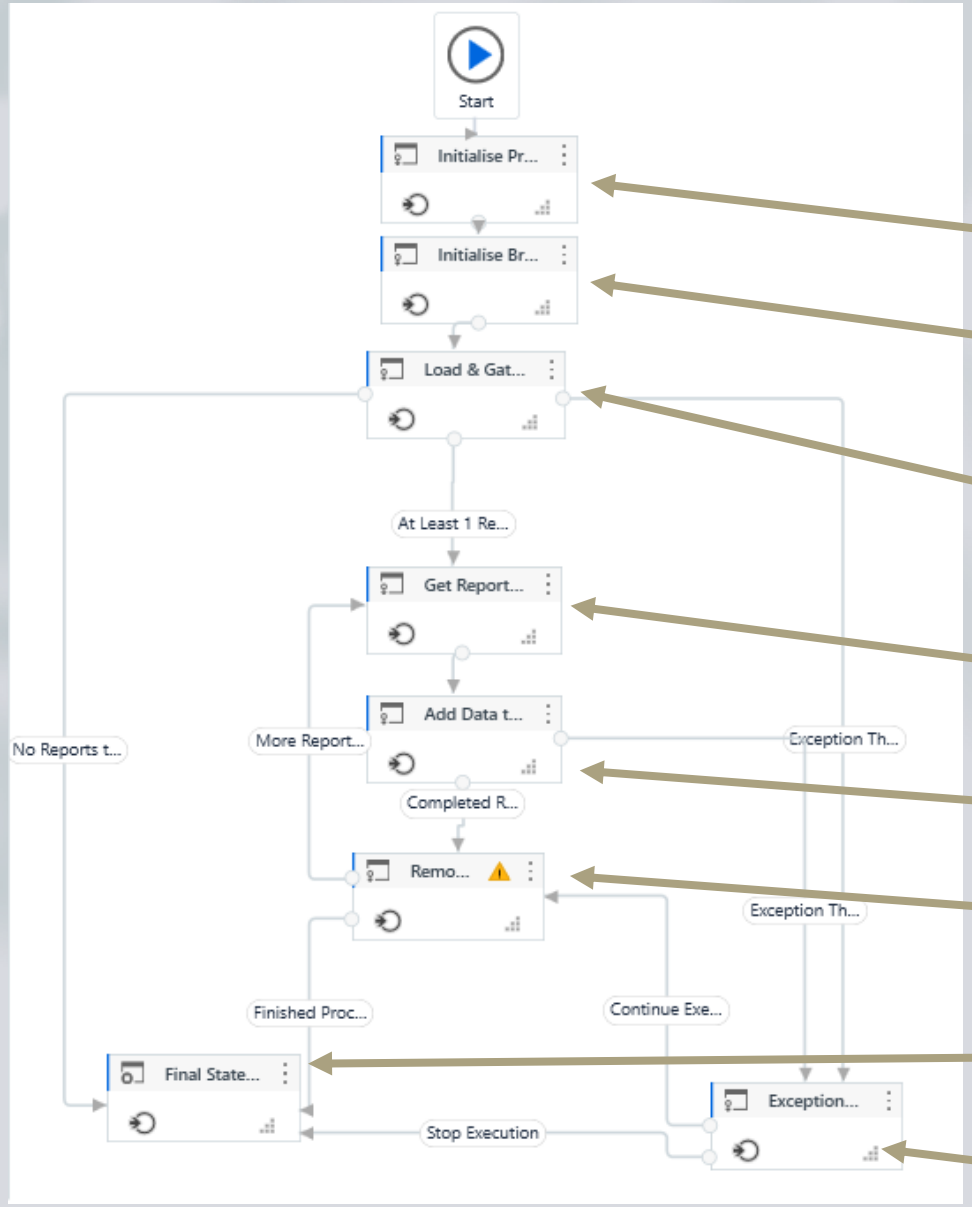
View in PowerApp/Assure



Logs



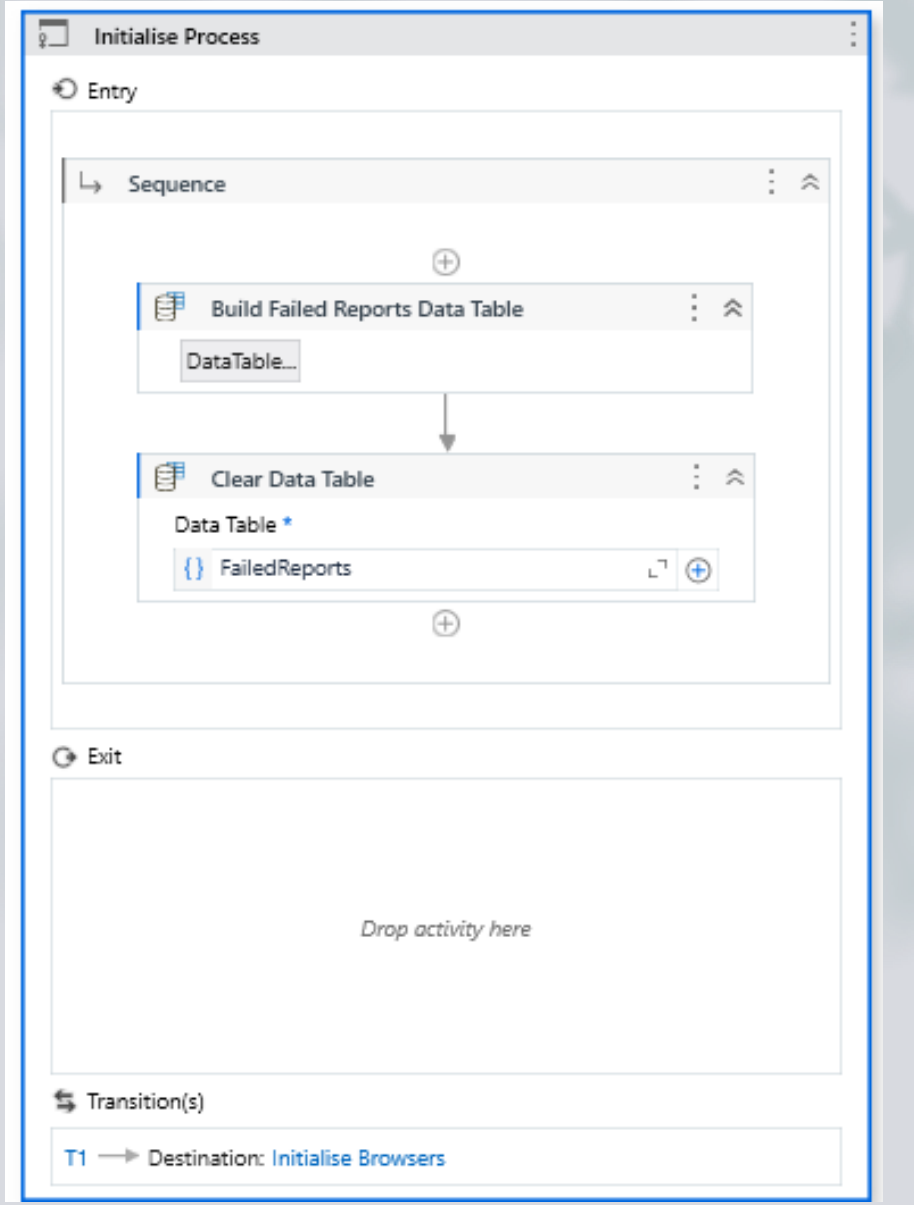
Overview of Process



1. Initialise Process – Set data variables (Slide 63)
2. Initialise Browsers – Open web browsers (Slide 64)
3. Load & Gather Reports – Get report numbers (Slides 65-73)
4. Get Report Data – Extract data from report (Slides 74 - 84)
5. Add Data into Assure – (Slides 85 - 101)
6. Remove report from queue – (Slide 102)
7. Close down applications as process finished (Slide 103)
8. Error handler – (Slides 104 - 106)

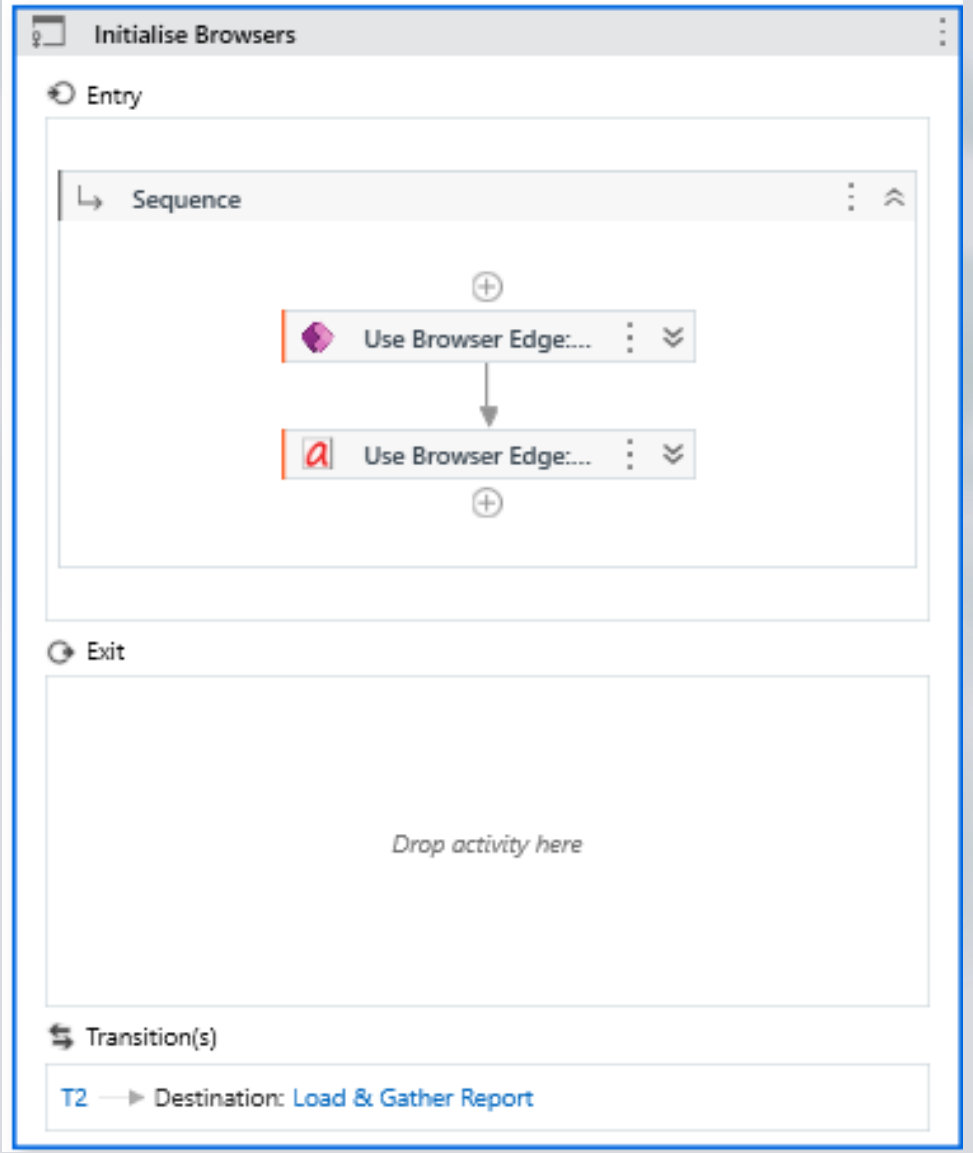
Initialise Process

- This state will build an empty data table that contains all the failed report numbers
- This can then be reported back to an actual staff member so the report can be dealt with manually

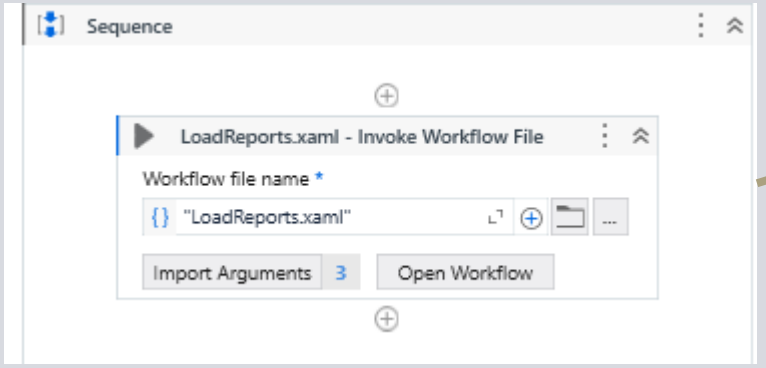


Initialise Browsers

- This process just opens the 2 web browsers
  - Opens the Safety Observations Power App
  - Opens Assure

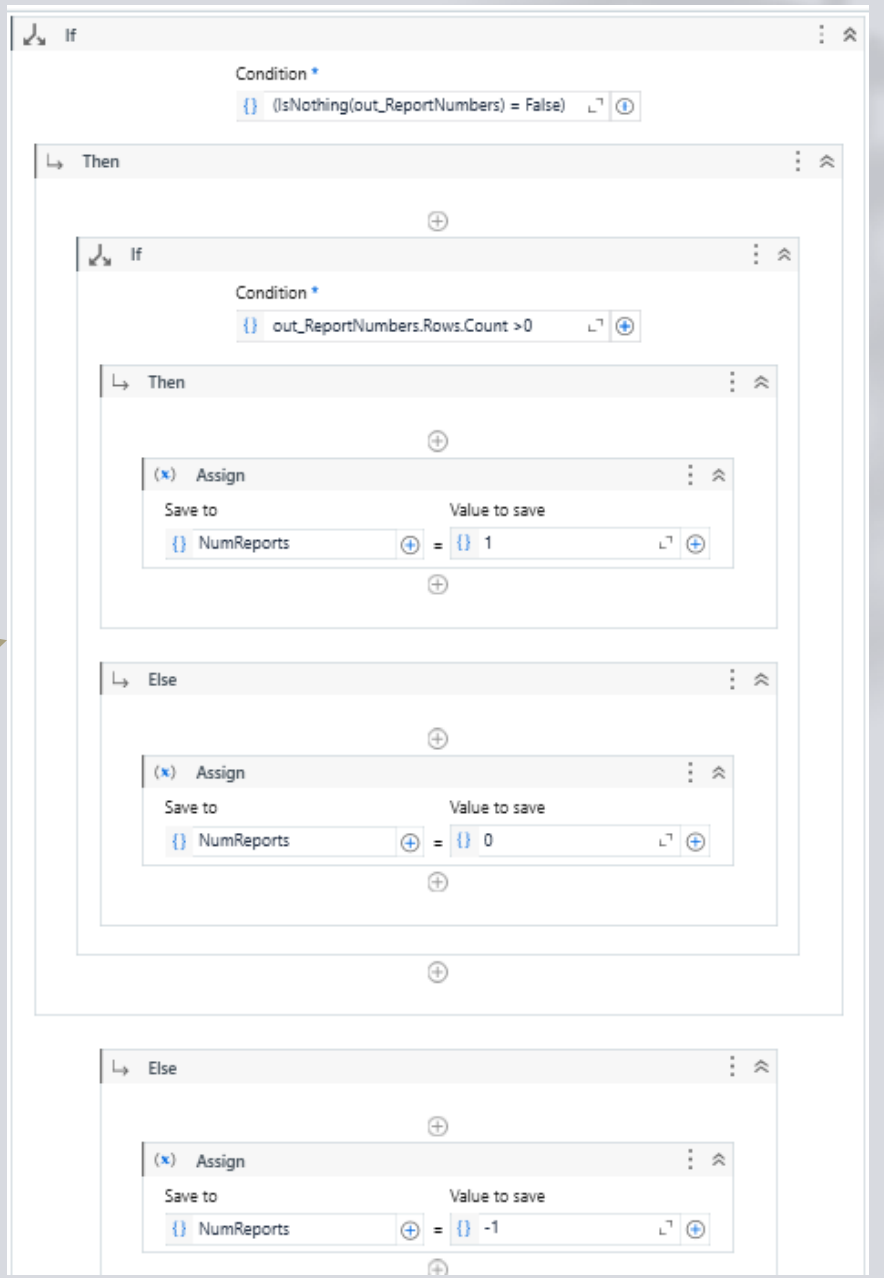


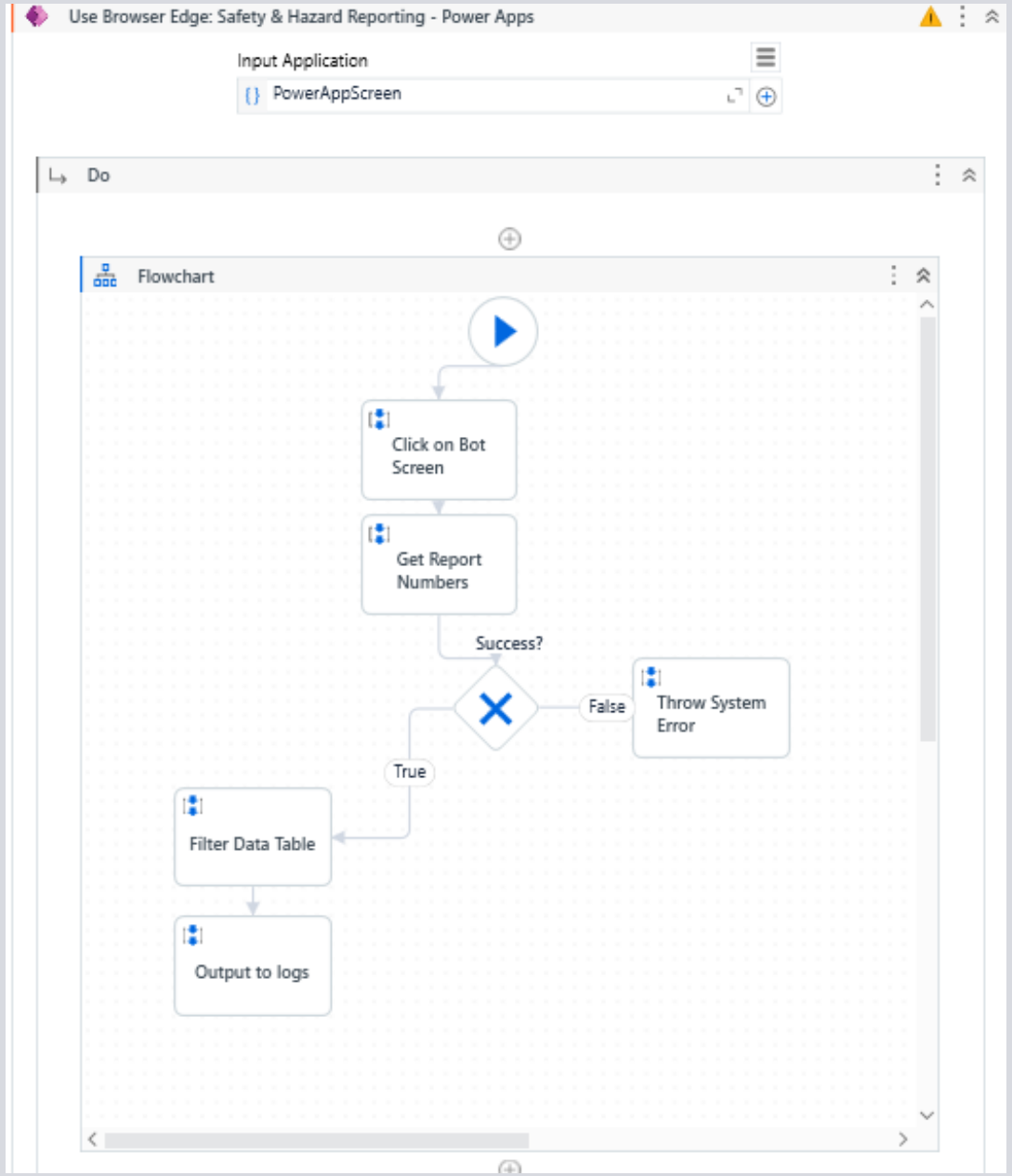
Load and Gather Reports (the dispatcher)



1

2

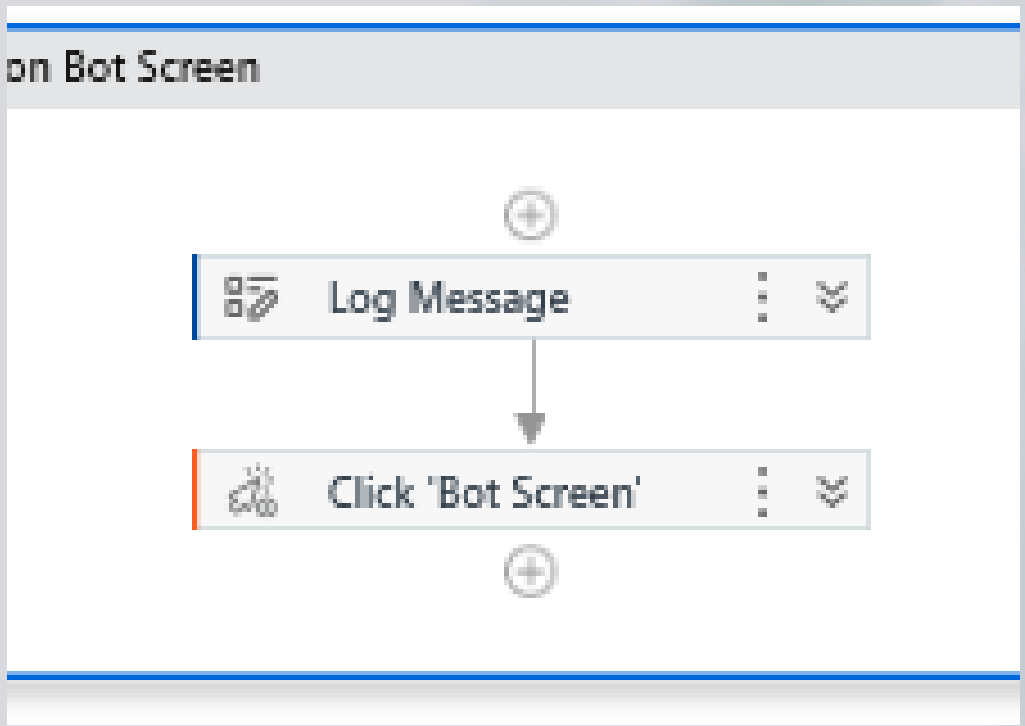




The structure the workflow

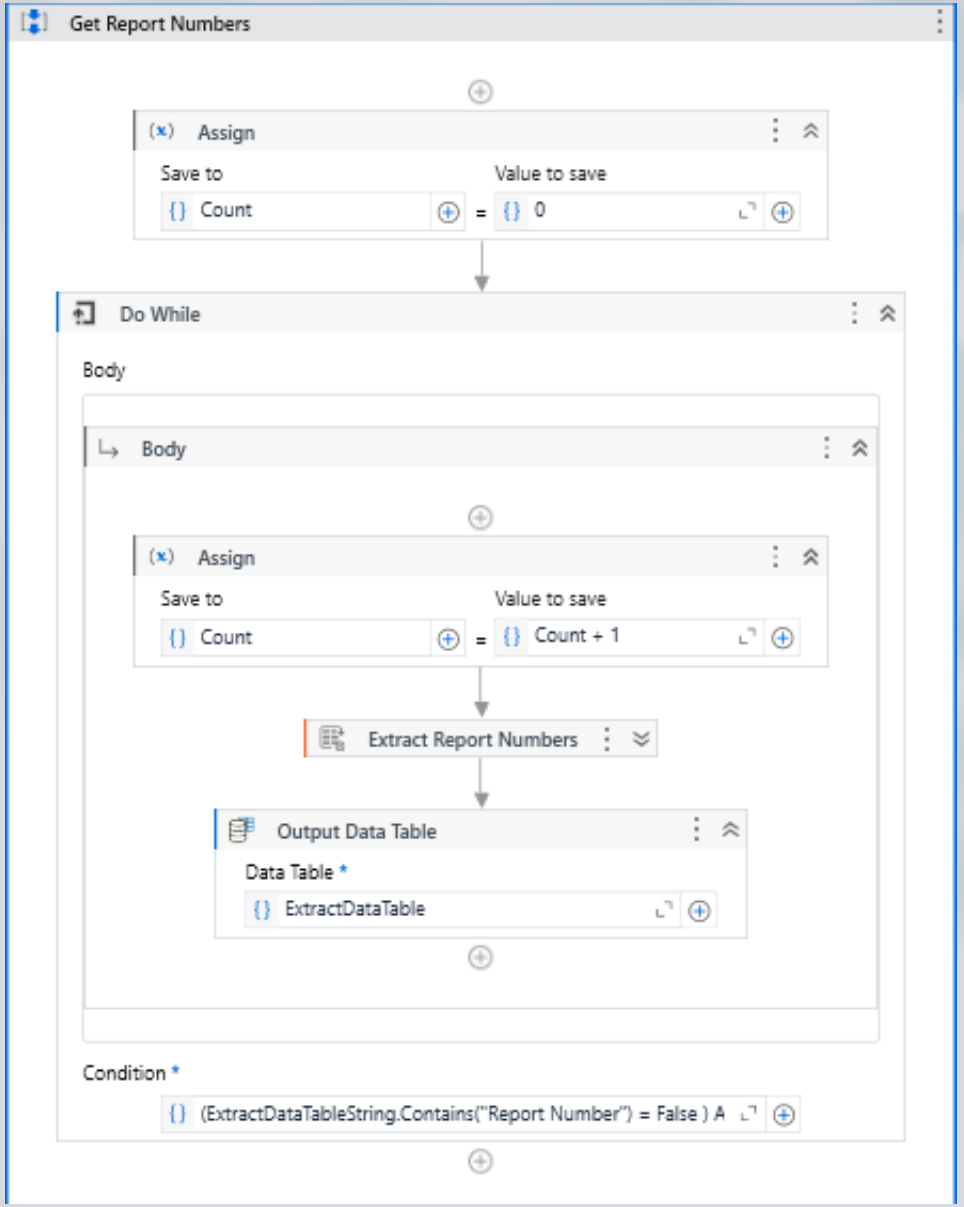
Step 1:

- This step tells the bot to click on the bot screen in the safety app.

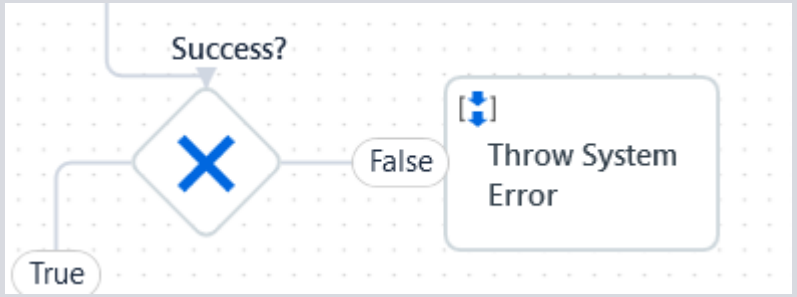


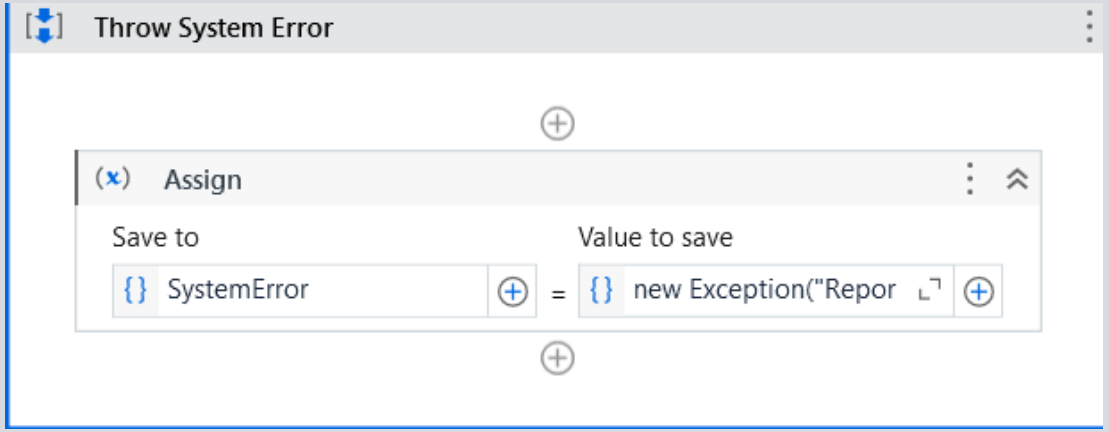
Step 2:

- This step is prone to failing, therefore I have this in a do-while loop until the reports are successfully captured. This is only done 3 times, however, if it continues to fail repeatedly there may be another issue so an error is sent so that this can be dealt with by a staff member
- This step uses a table extraction to extract the list of report numbers
- To check for success, we check if the outputted data table (As a string) is empty or null



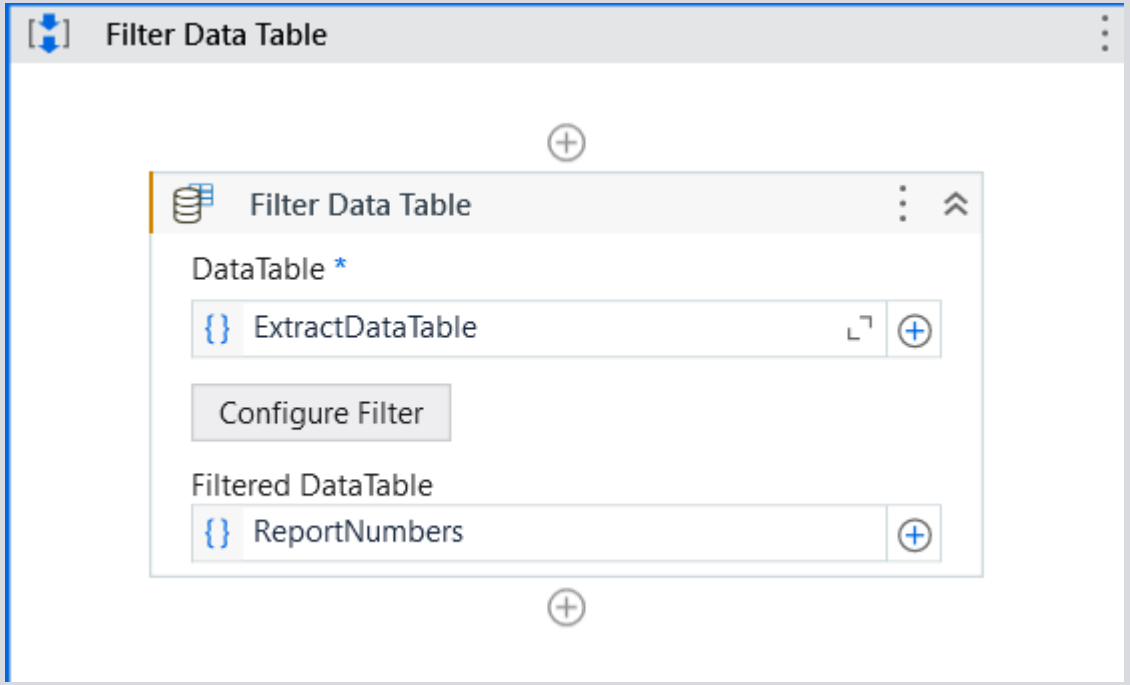
Step 3:





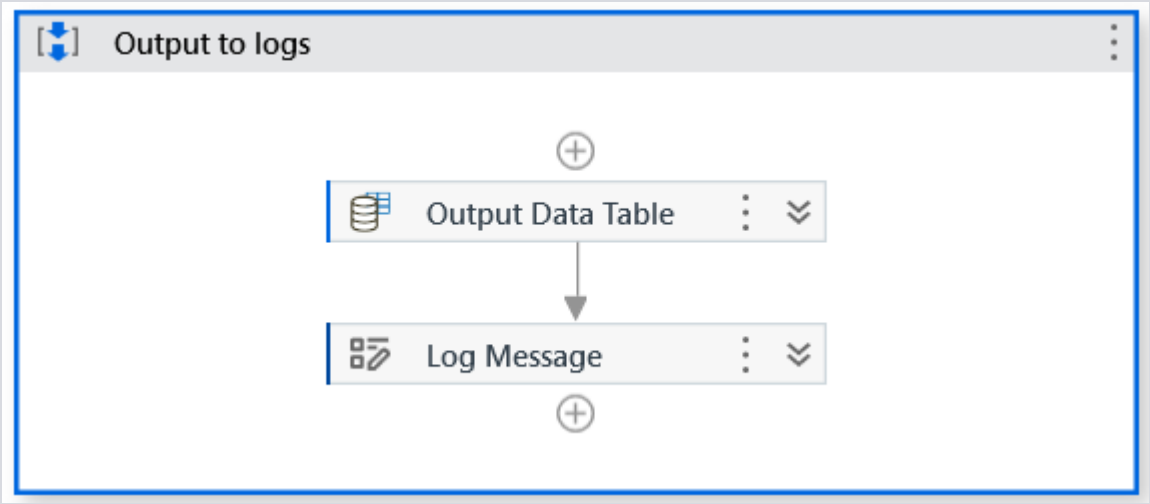
- This section of the flow chart checks if the table is null/empty
- If it is still null or empty, the error is still occurring after 3 attempts, so a system error is thrown, and the process stops

Step 4:



- If the extraction is successful, we move on to the filtering of the data table
- As the table extraction takes in a couple of extra lines of data we don't want, we filter the table to only include lines containing "Report Number:"

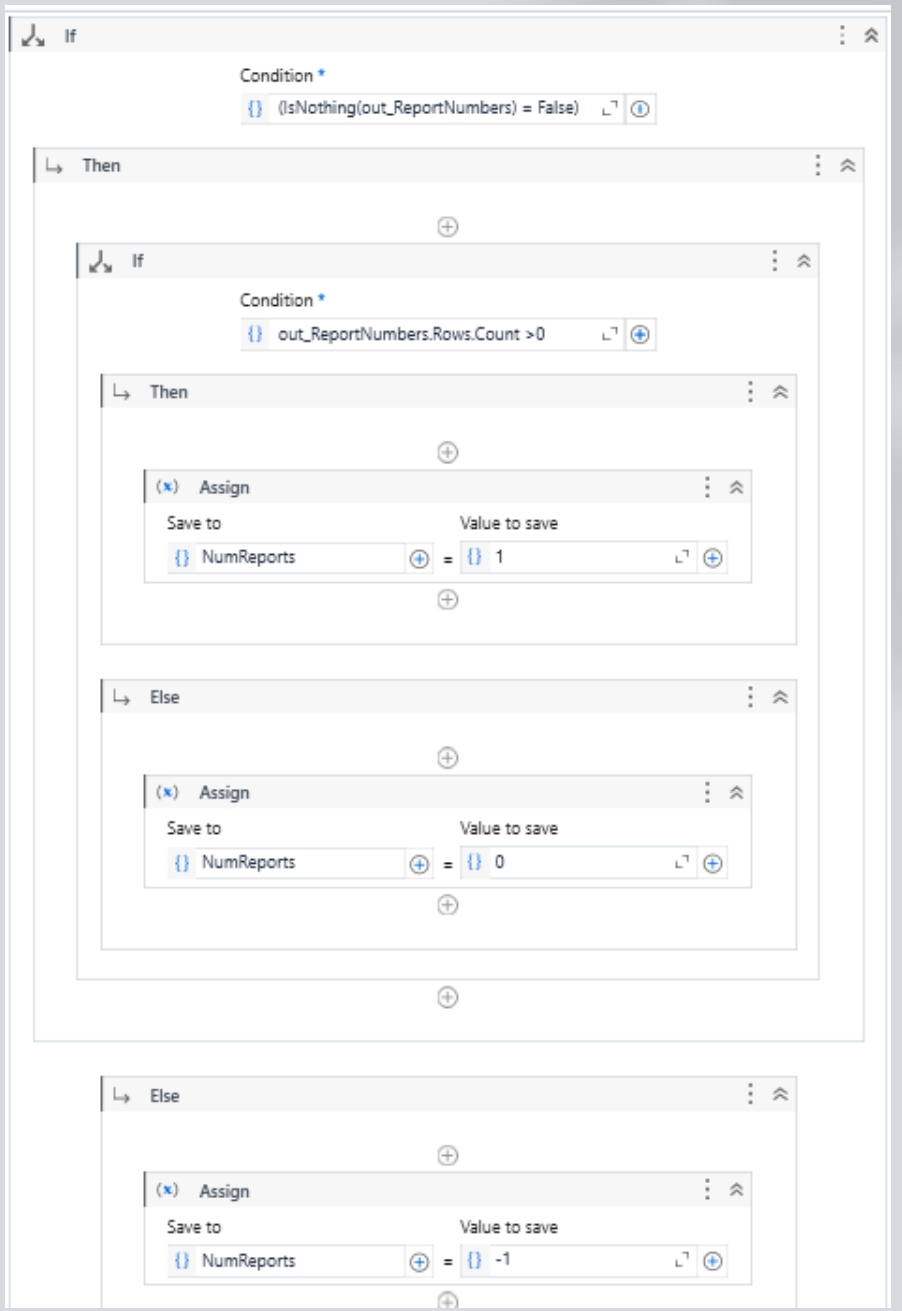
Step 5:



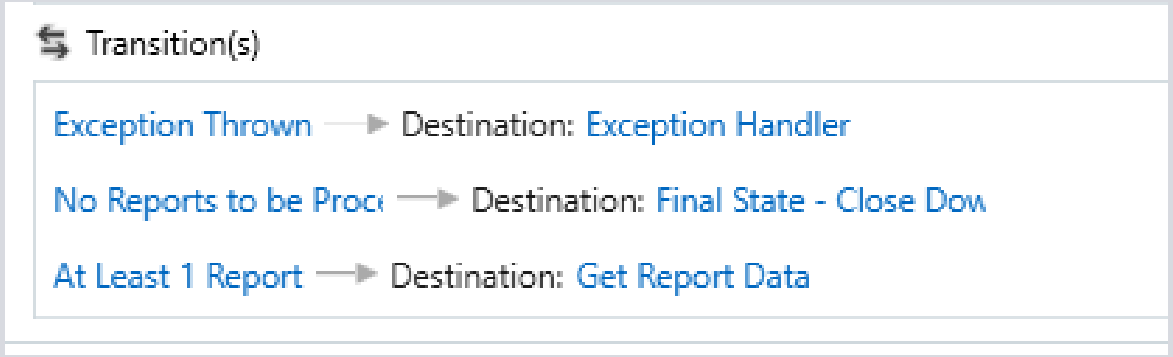
- This step is mainly for testing purposes
- This outputs the list of report numbers extracted to the log

Step 6:

- We then check (using an if statement) if there is
  - A) No Reports
  - B) More than 1 Report
  - C) An error
- If there are no reports, the output argument is not null but the number of rows in the table is 0. If the number of rows is more than 0 then there are multiple reports
- If the output argument is null then an error was thrown
- Based on this we set the variable NumReports to 1, 0 or -1
- This variable can then be checked when deciding the next state in the process

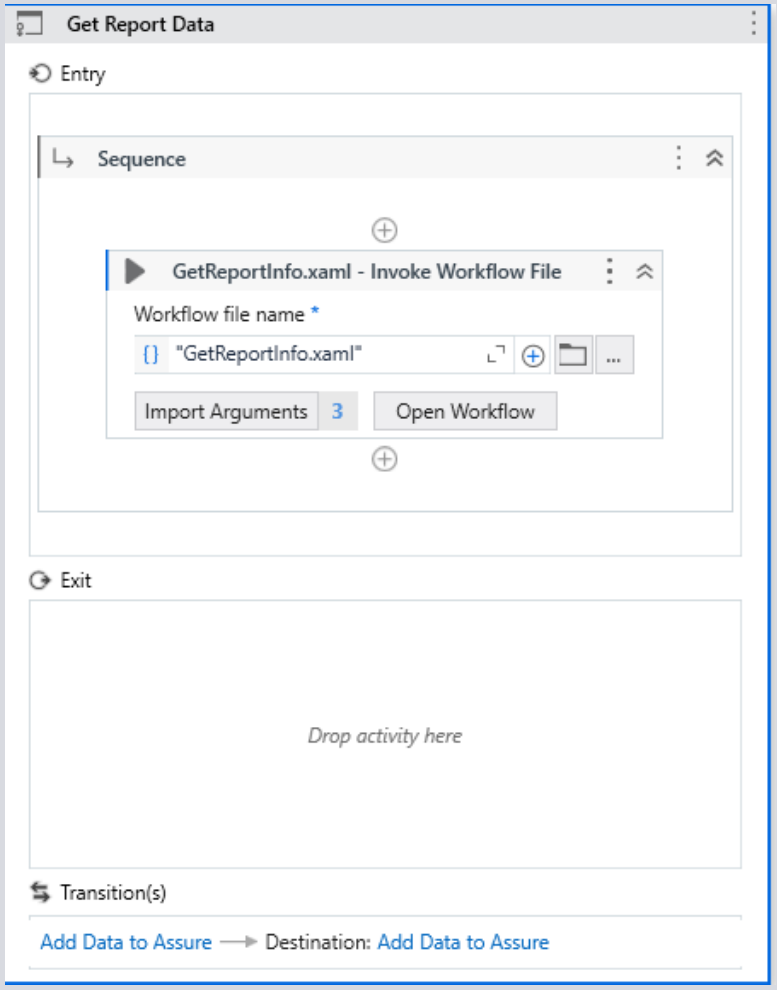


Step 7: Choosing the next state

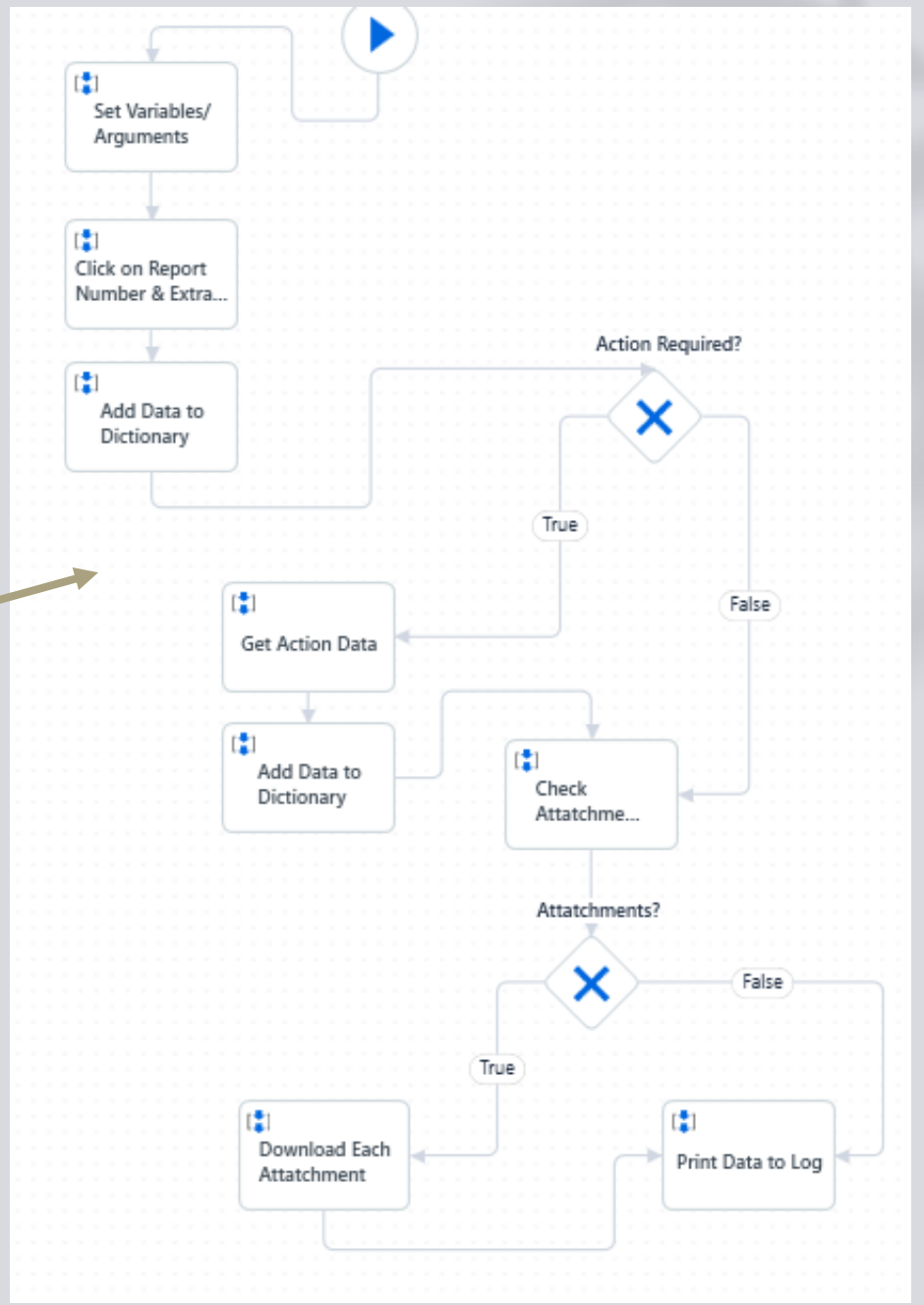


- There are 3 different transitions
- These transitions check the value of NumReports to follow the right path
  - A) Error Thrown – Path to Exception Handler (Slides 104)
  - B) No Reports – Path to the final state and close down process (Slide 103)
  - C) At least 1 report – Continue to Get report data (Go to Slide 74)

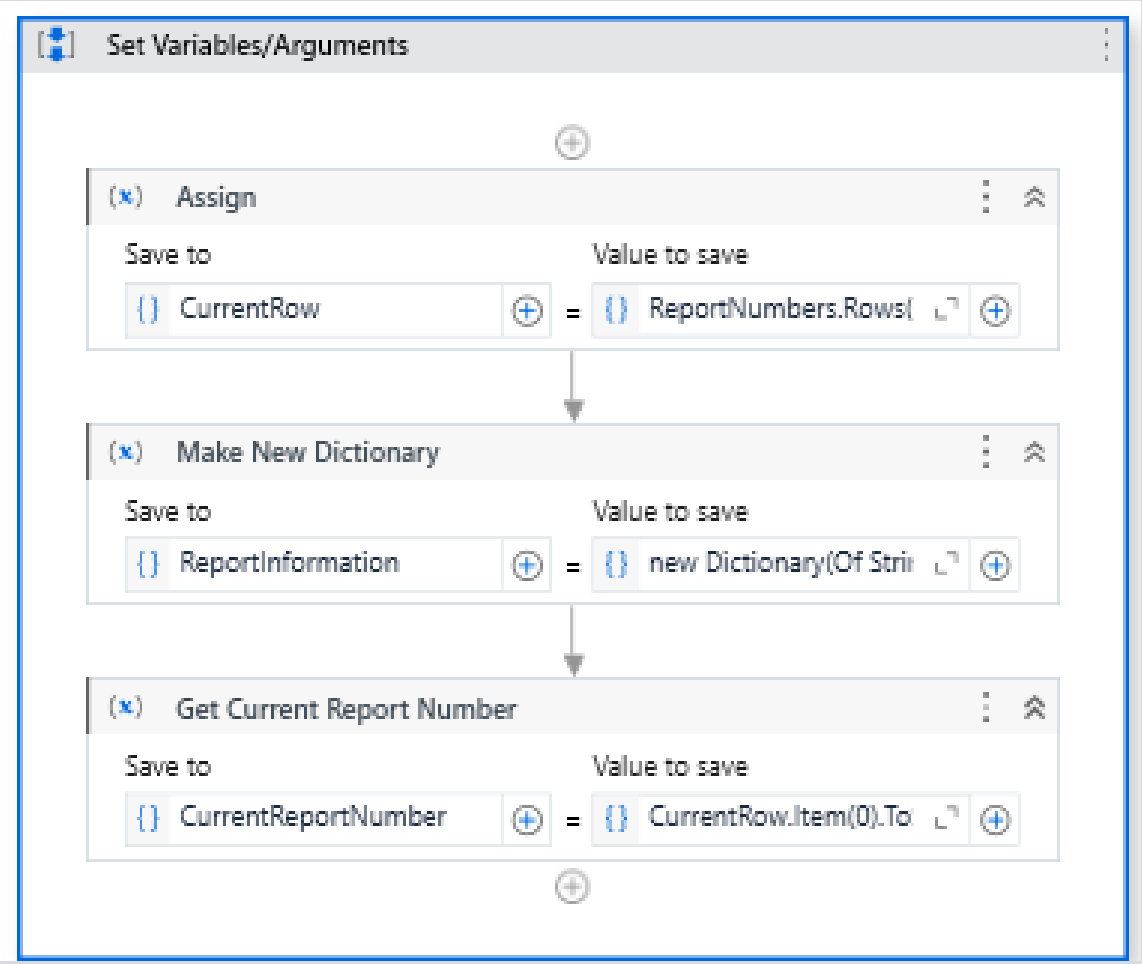
Get Report Data



Structure of this workflow

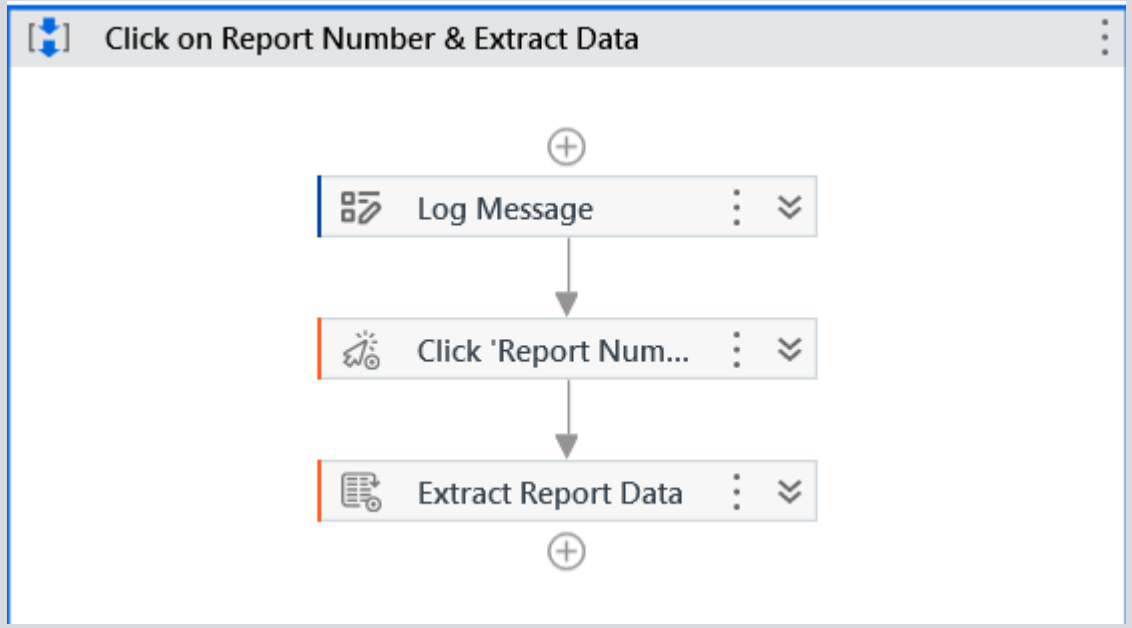


Step 1:



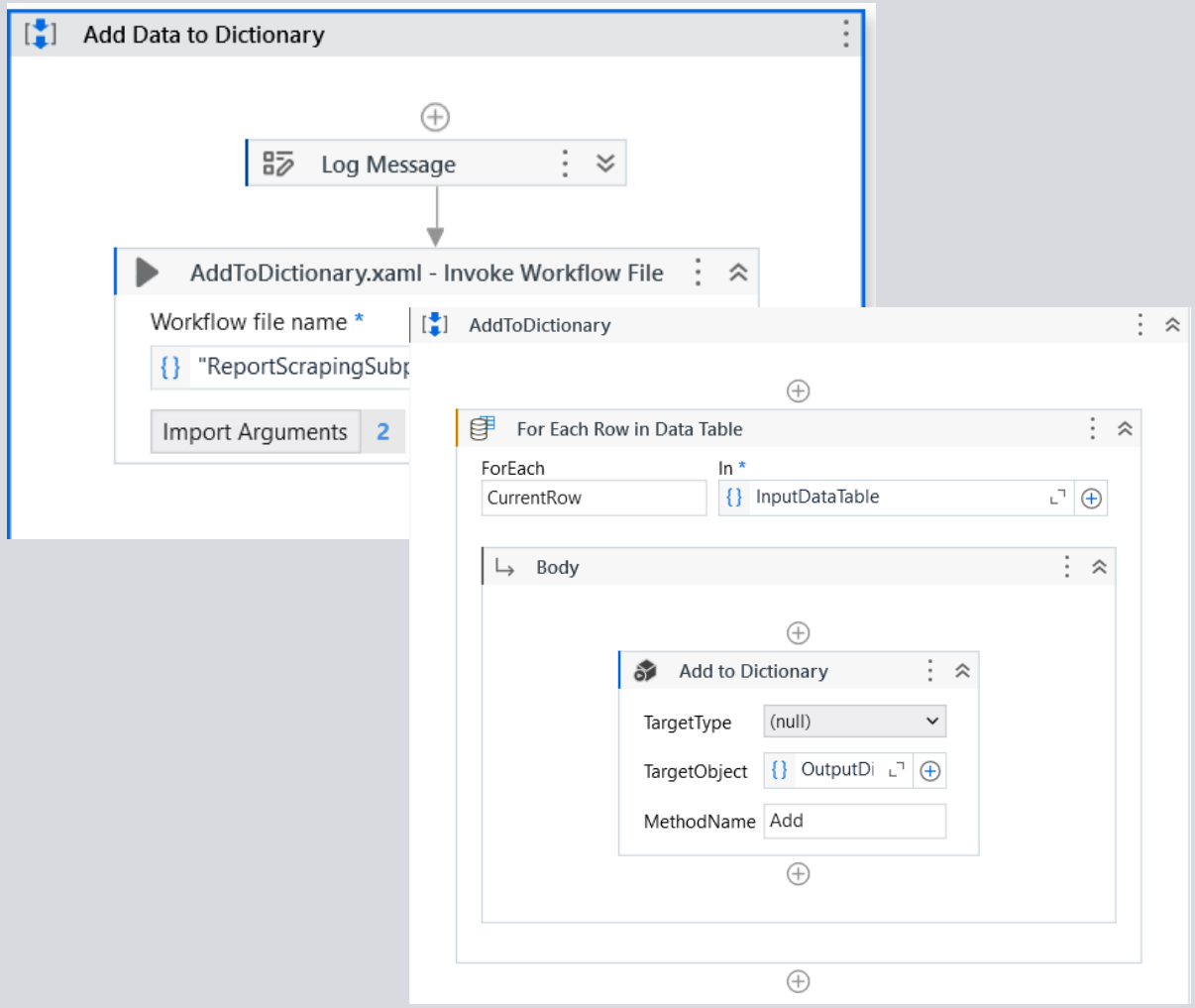
- This step gathers the current report number from the current row in the ReportNumbers data table and initialises the dictionary ReportInformation which we will use to store the data

Step 2:



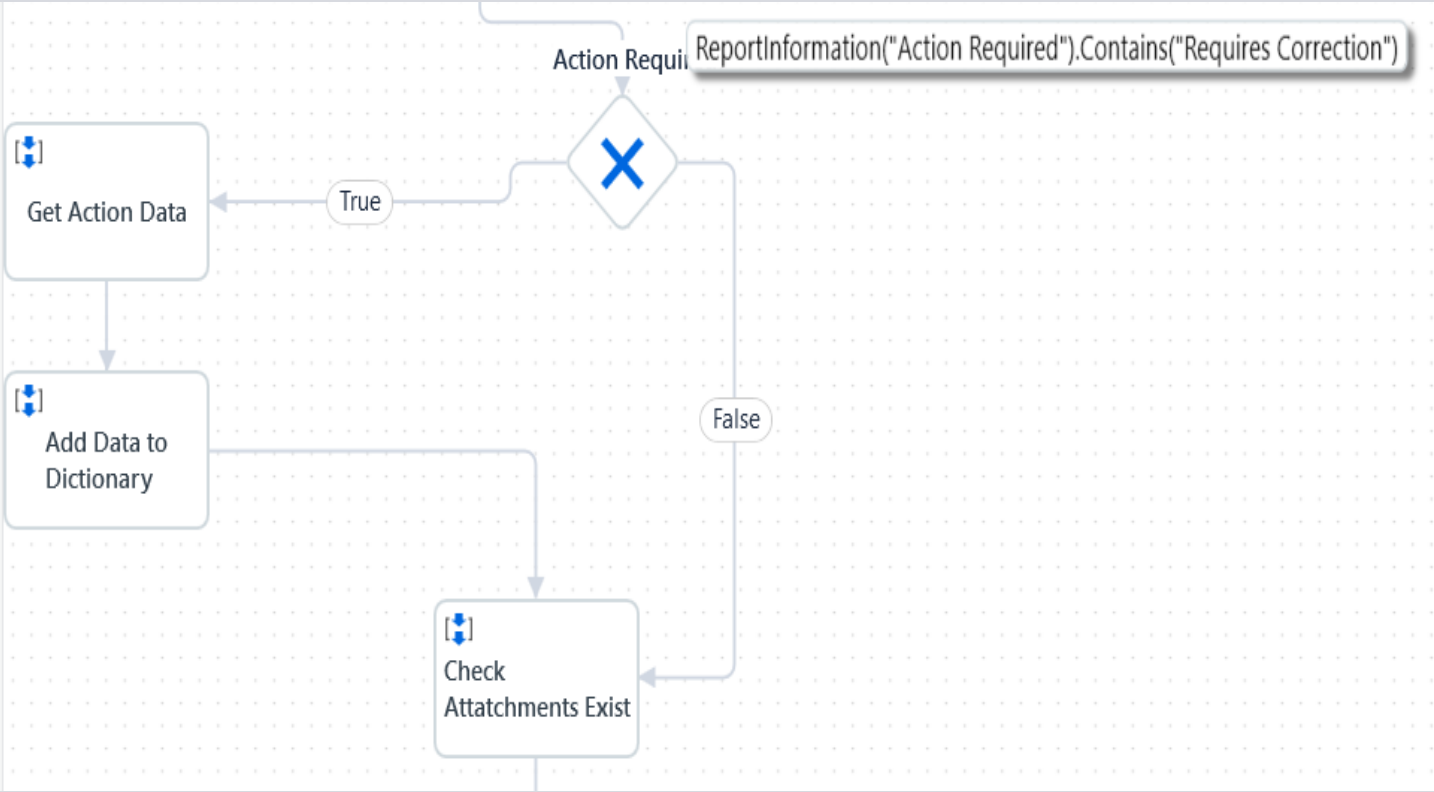
- This step will click on the report number from the list of reports on the left-hand side
- Then using a table extraction, the bot will extract all the data from the main data section

Step 3:



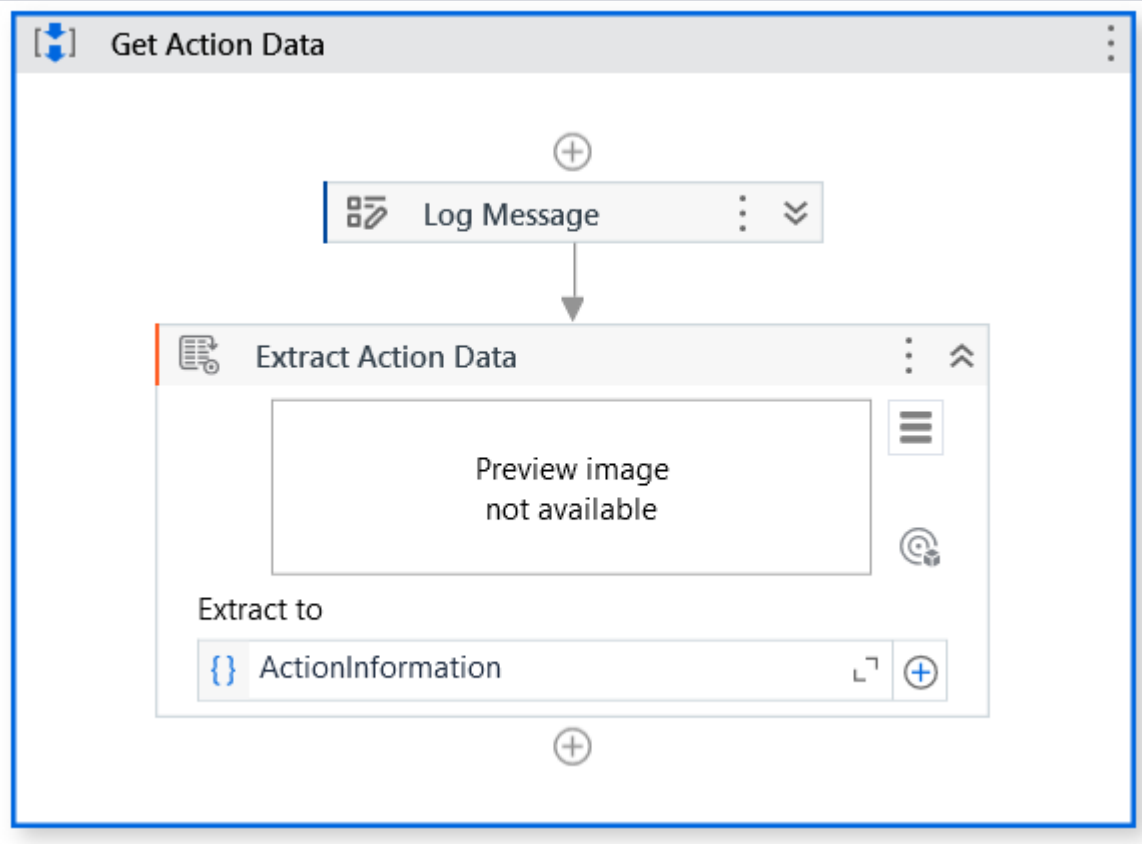
- This step uses the add to dictionary workflow
- This process will take the extracted report data table and go through it row by row and add each type of data to the dictionary

Step 4:



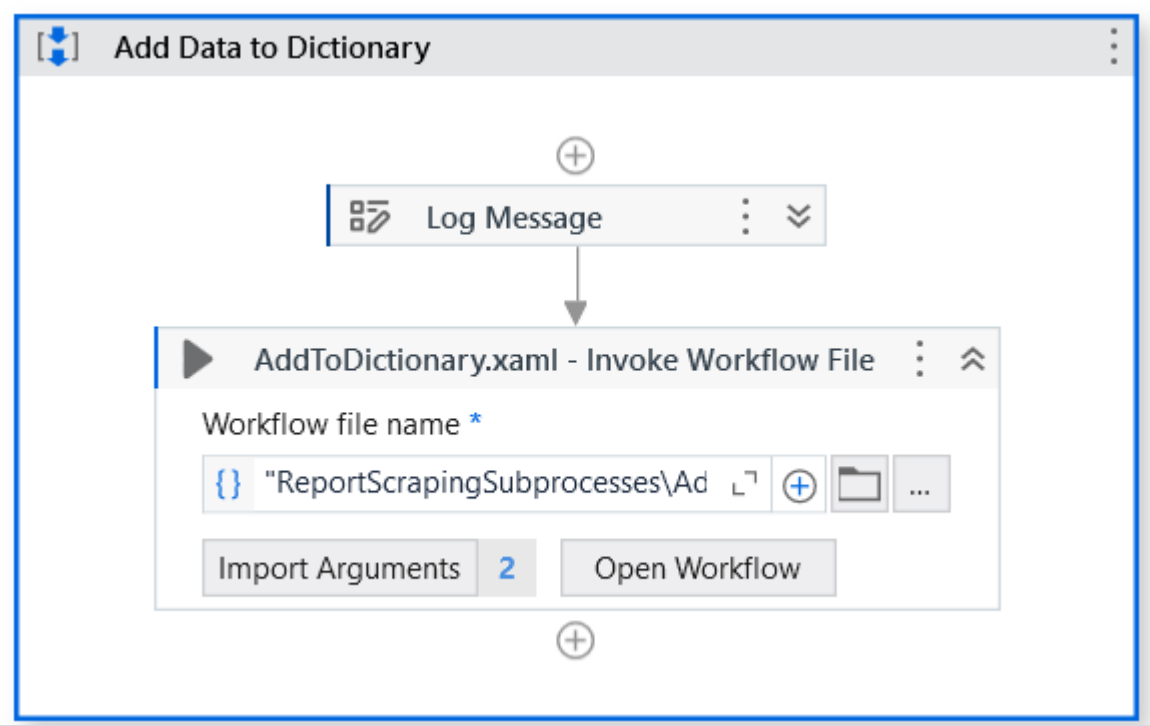
- This part of the flowchart checks if action is required for the report
- We do this by checking a field in the dictionary called ‘Action Required’

Step 5: ACTION REQUIRED



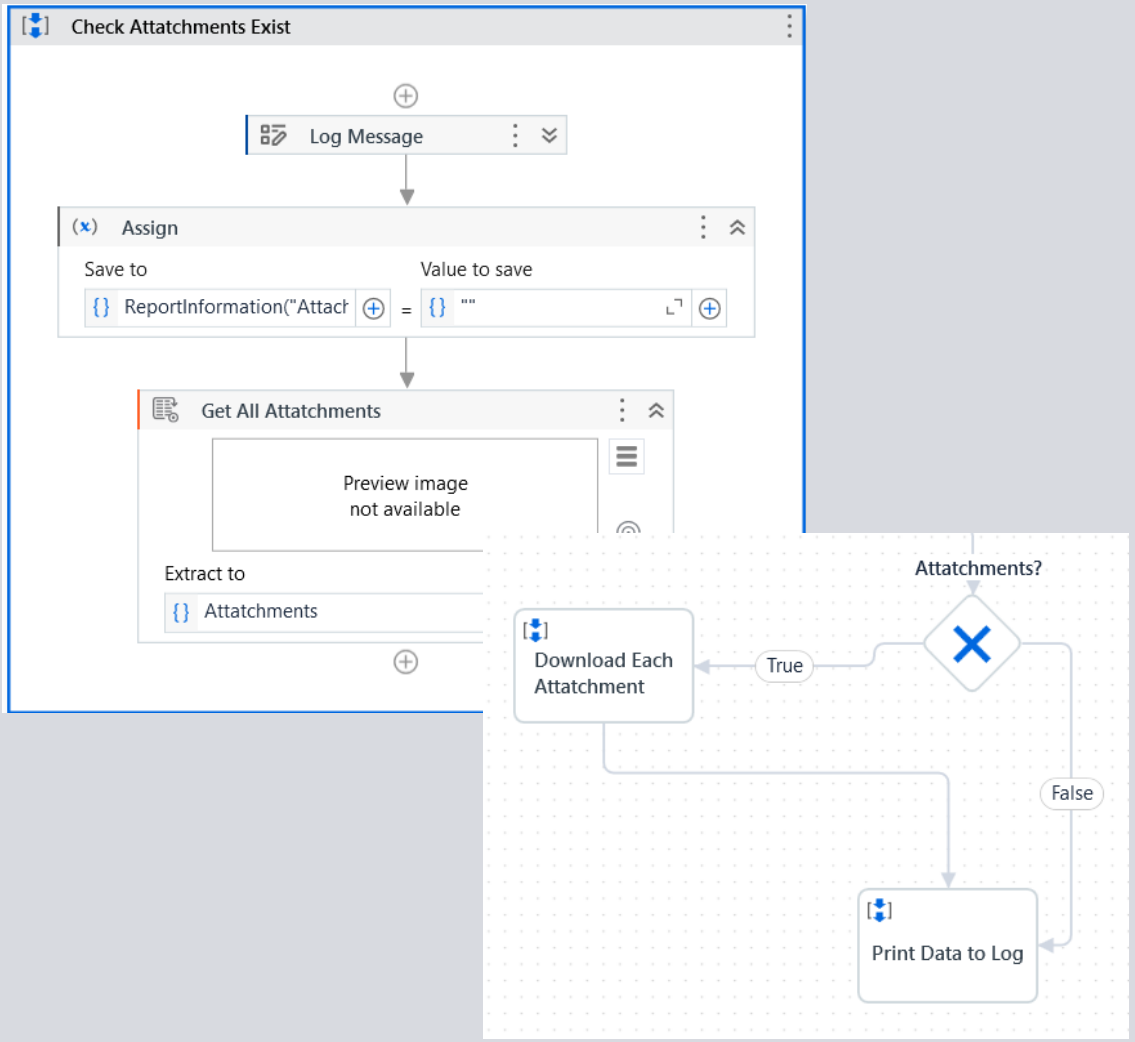
- This process will use the table extraction again to get the action data

Step 6: ACTION REQUIRED



- This process adds the data to the dictionary
- This is the same process as shown in slide 78 – Step 3

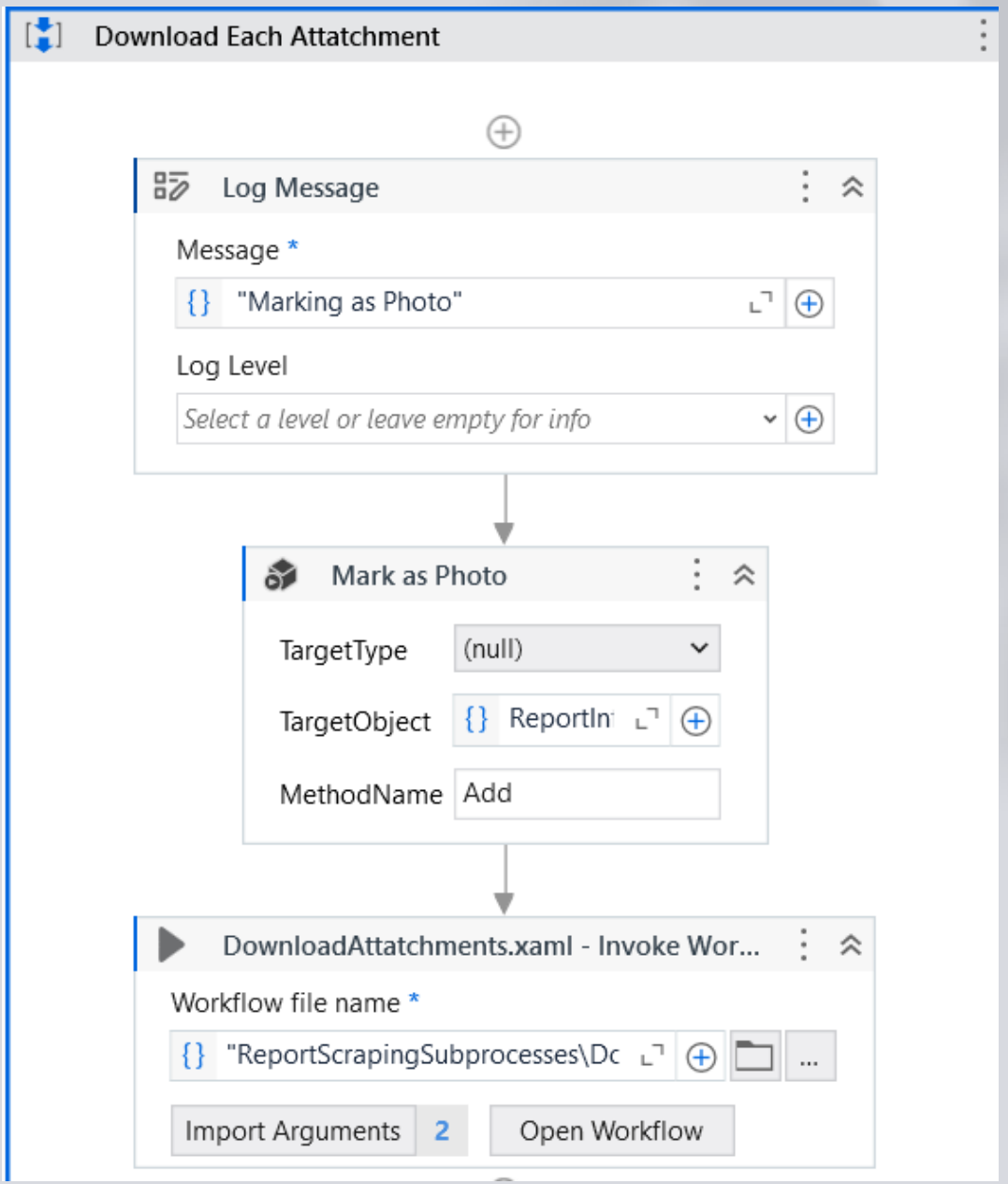
Step 7:

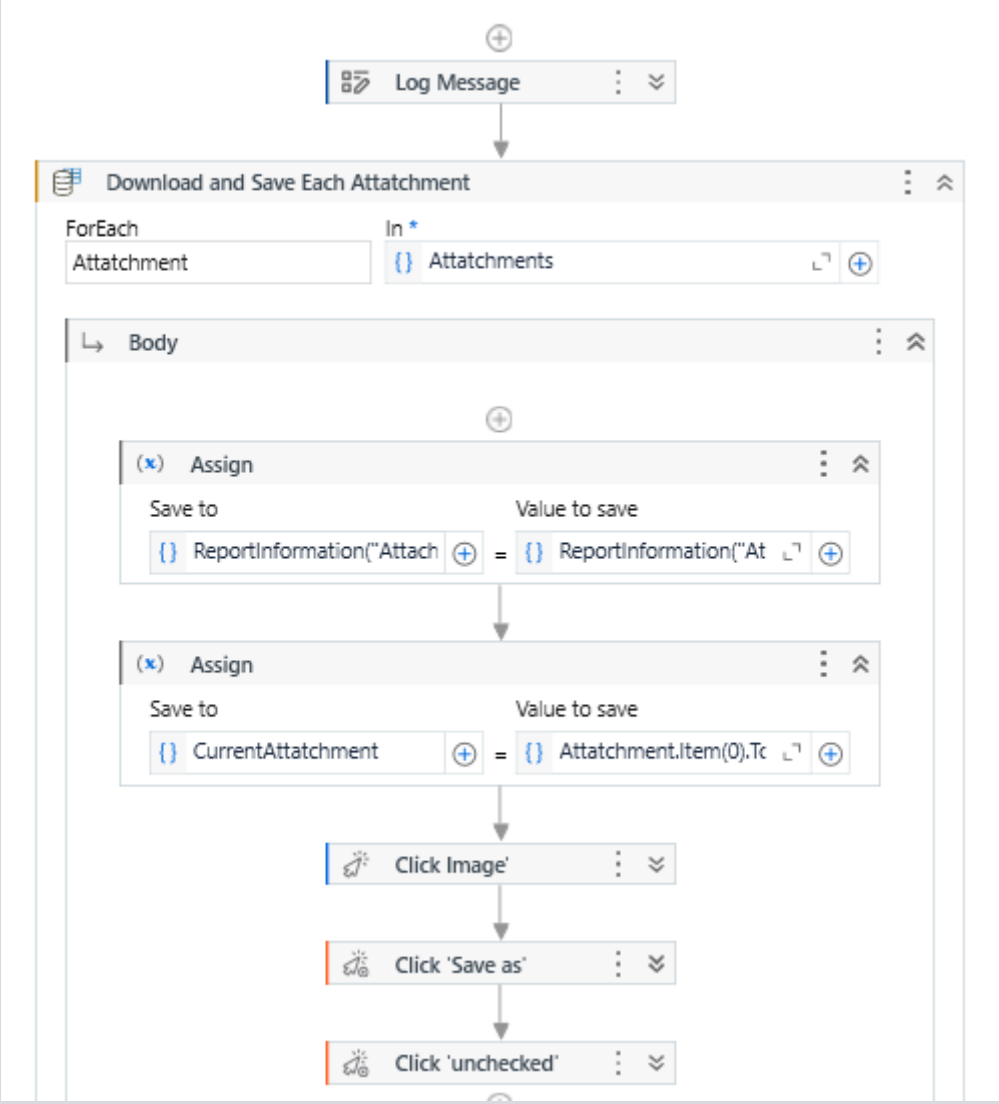


- This process will check if there are any attachments that we must save
- This will use the extract data table again to look at data stored under the ‘Attachments’ title
- This is then checked by the flowchart to see if the extracted data table is empty or not. If it isn’t empty, there are attachments to process

Step 8: ATTACHMENTS TO PROCESS

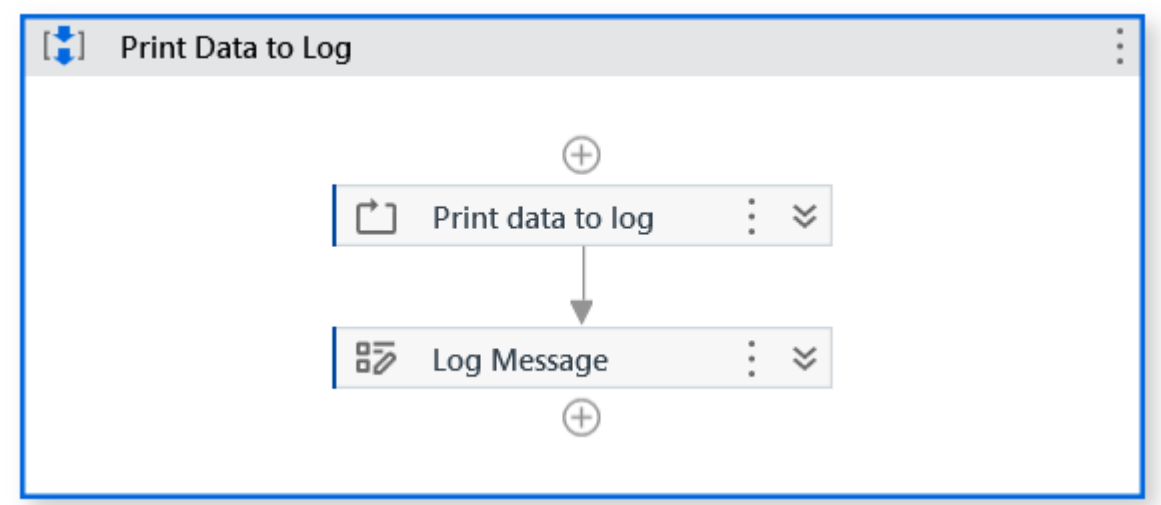
- This process will mark the report as photo
- Then it will go through each attachment and download them (this is shown in the next slide)





- This process will go through each attachment in the table and…
  - Add the attachment to the dictionary
  - Then click on the attachment
  - Save the image
  - And click ok
  - Then repeat this for the next attachment (if there is one)

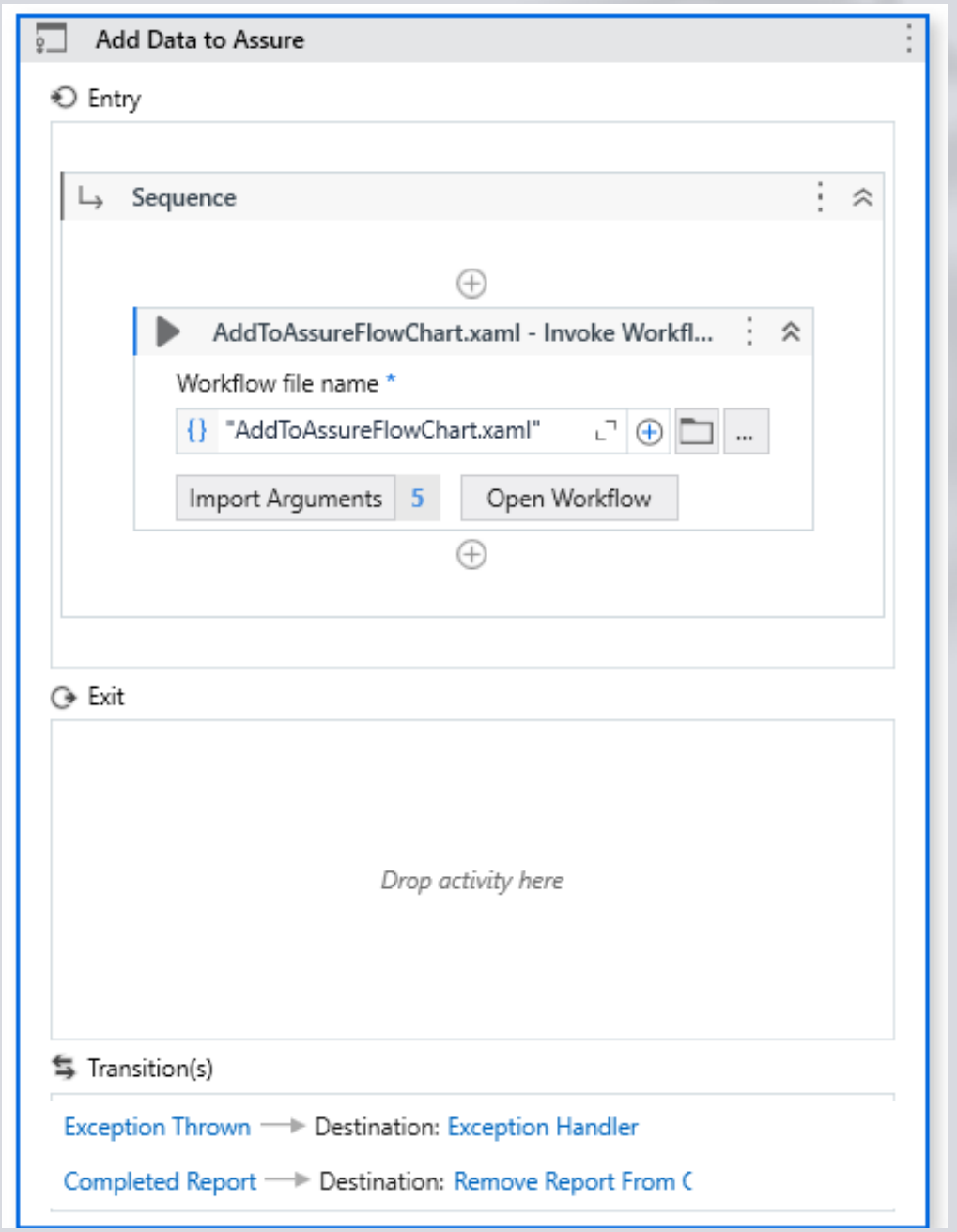
Step 9:

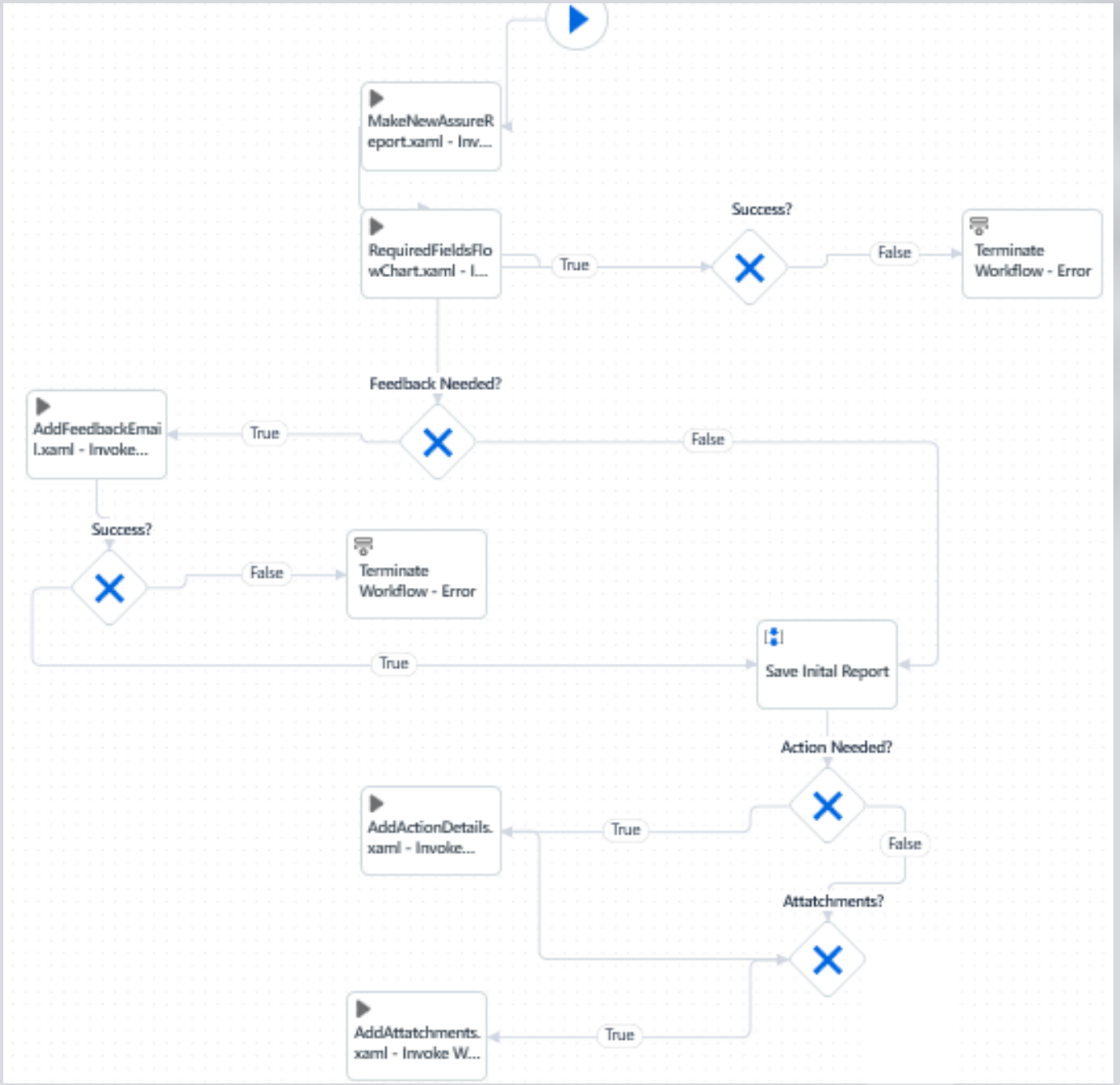


- Finally, we output the collected data to the log
- This step is mainly for testing purposes

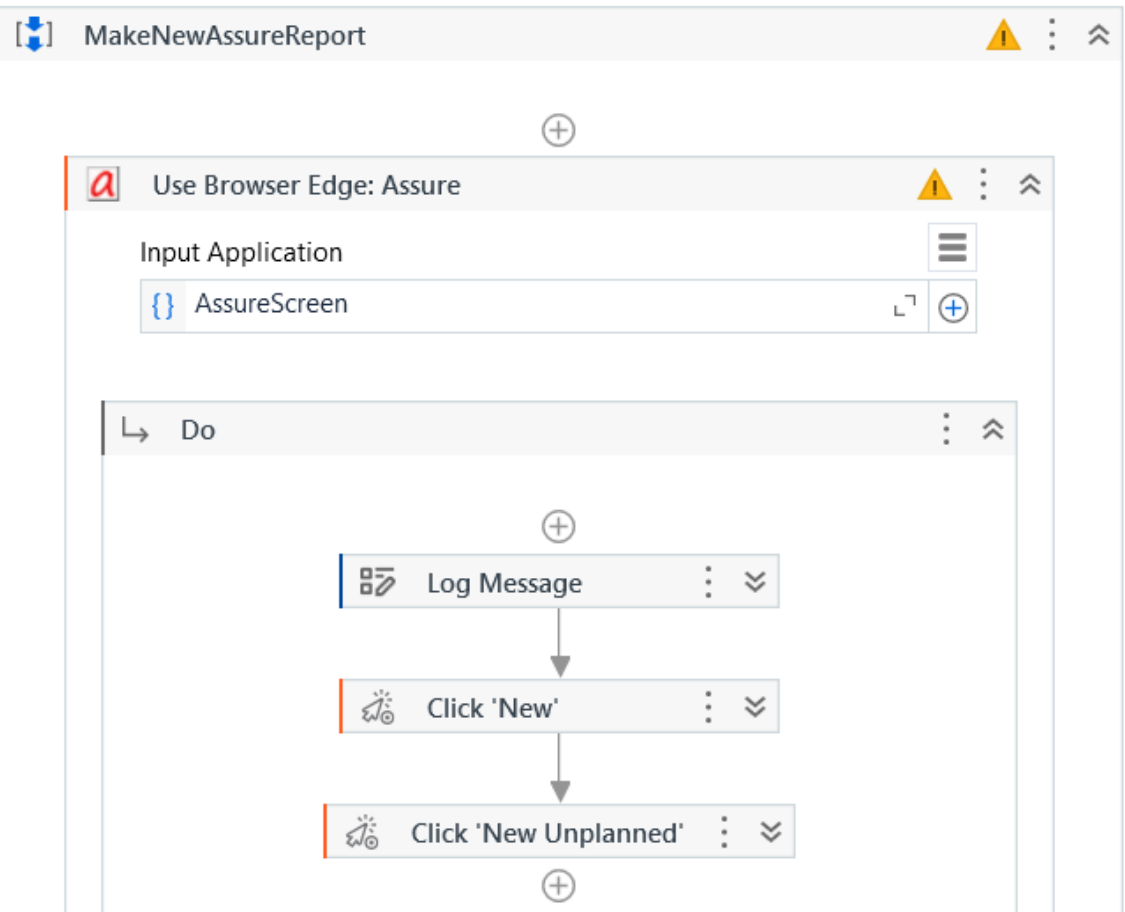
Add data to Assure

- This process adds the extracted data from the power app into Assure
- The structure of this workflow can be seen on the next page



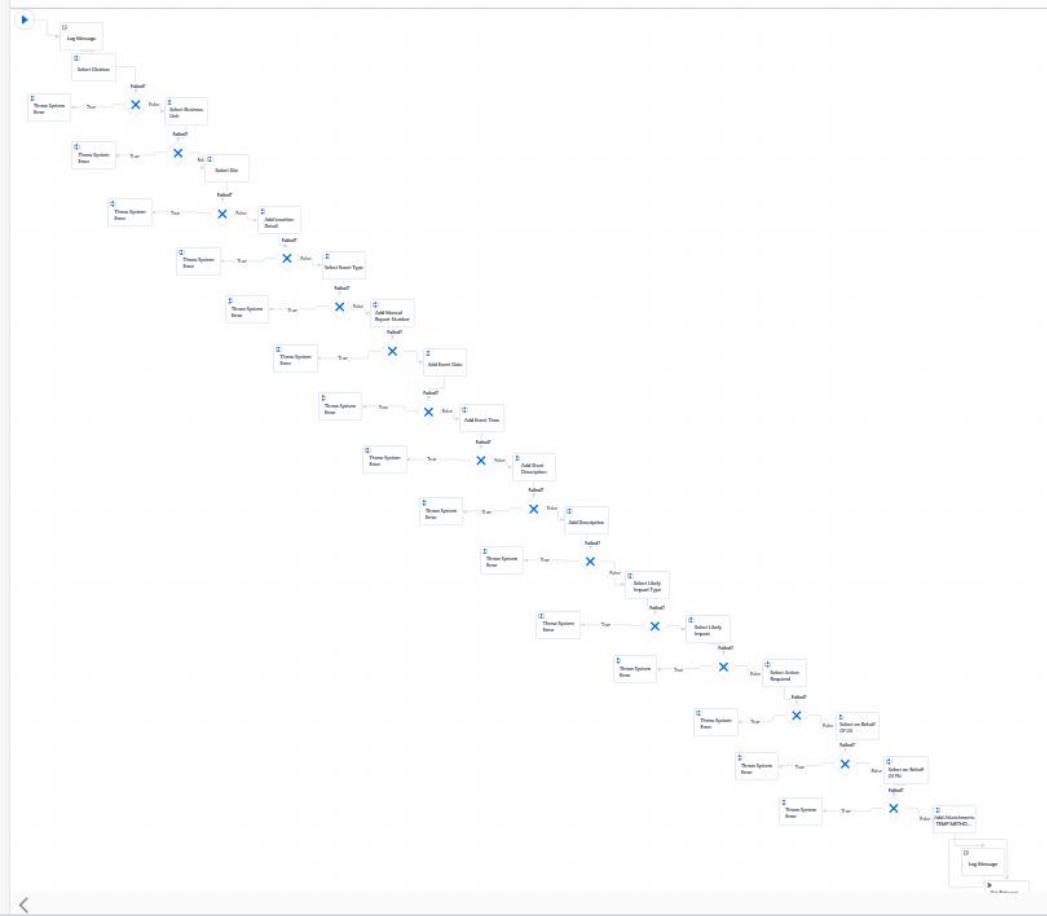

MakeNewAssureReport.xaml - Inv...
RequiredFieldsFlowChart.xaml - I...
True
Success?
False
Terminate Workflow - Error
Feedback Needed?
AddFeedbackEmail.xaml - Invoke...
True
False
Success?
False
Terminate Workflow - Error
True
Save Inital Report
Action Needed?
AddActionDetails.xaml - Invoke...
True
False
Attatchments?
AddAttatchments.xaml - Invoke W...
True

Step 1: Make a new Assure Report

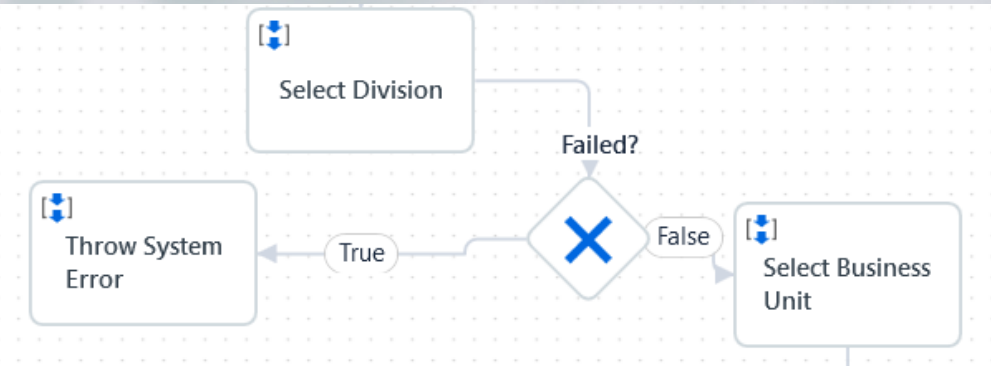


- This involves going into Assure and creating a new unplanned event

Step 2: Adding the required fields



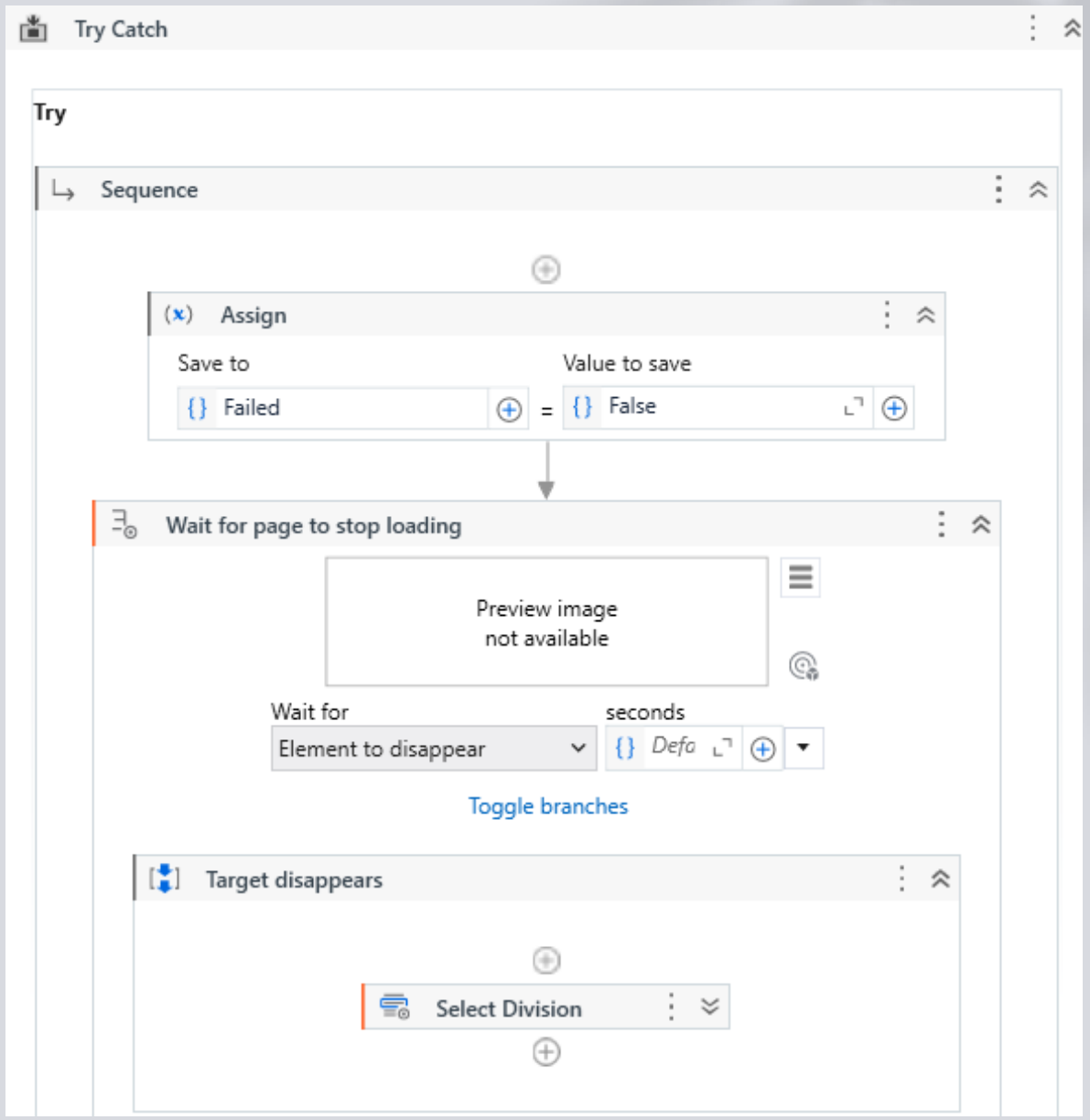
- As you can see here, this process has many steps
- Each step in this flow chart correlates to a data field that requires data in assure
- Each field is its own step as these steps are prone to error. Therefore, we use a standard structure for each data field that includes retries and error checks
- If the process fails for any data field, an error is thrown and the process is stopped



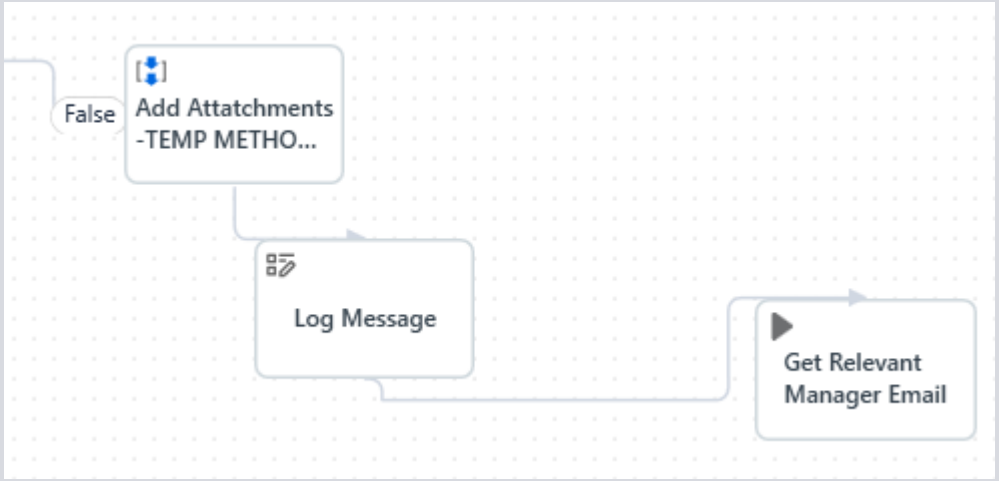
- This is an example of what each of the steps looks like
- The flow chart tries to select/type into the field
- If an error occurs, then throw that system error and stop the process
- Otherwise, carry on to the next field

Step 2a:

- This is from the ‘Select Division’ Step
- This process is inside a do while loop. That loop will check for any errors caught by the try catch and it keeps track of how many retries. If the process fails and it has retried less than 3 times, it will retry again.
- After 3 errors, the process will stop and an error is thrown to the main process so it can be dealt with by a human
- The process itself will wait for the page to stop loading by checking for a ‘Please Wait’ element. Then it will either select the item from a dropdown or type into a box. If at any point the process throws an error, it is caught by the try catch.



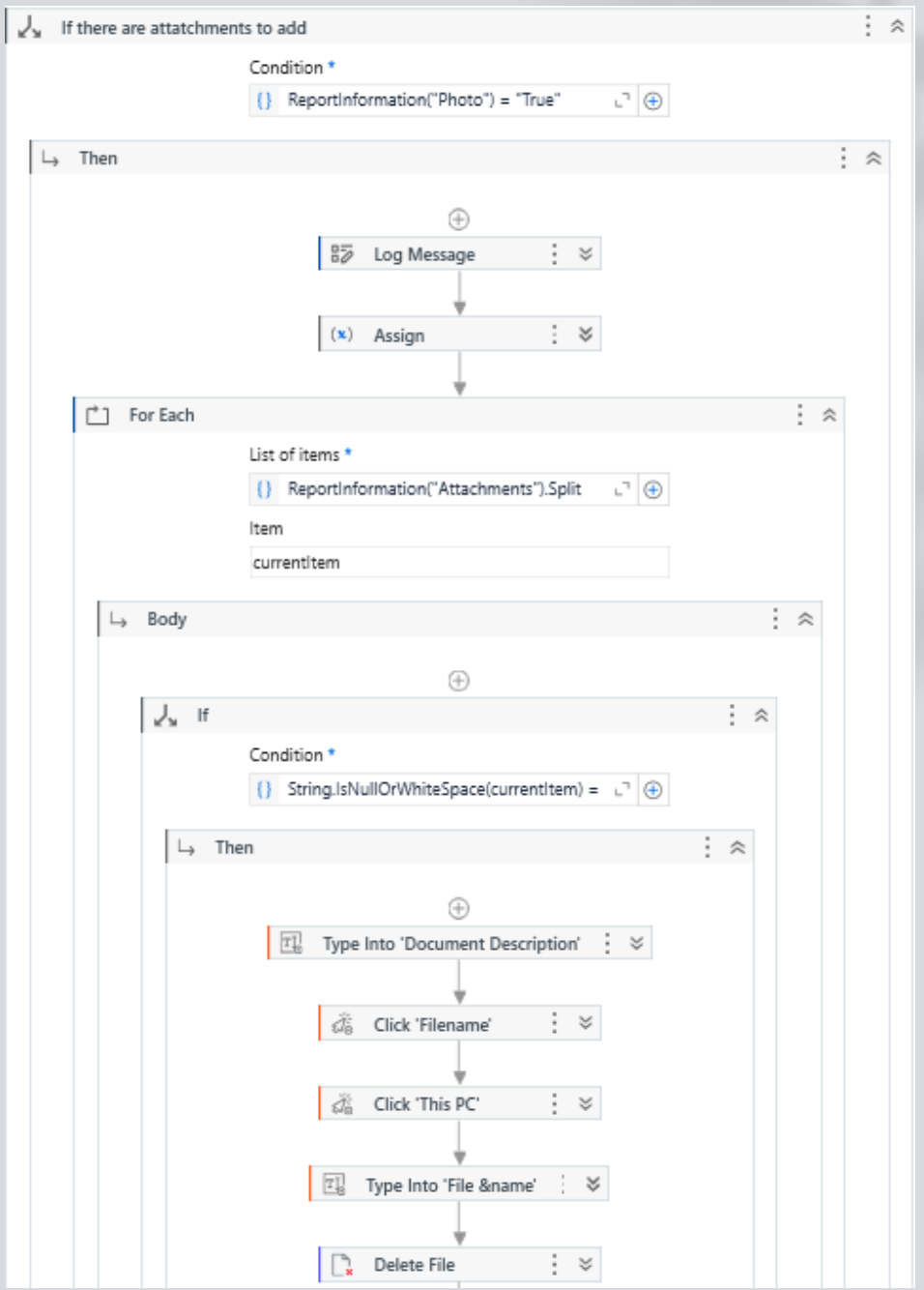
Step 2b:



- The final stages of step 2 are not the same as ‘Select Division’
- These stages include a temporary way of adding attachments to the report. The reason this is temporary is due to technical issues with Assure at the time of development.
- Then it includes the method for finding the relevant manager’s email

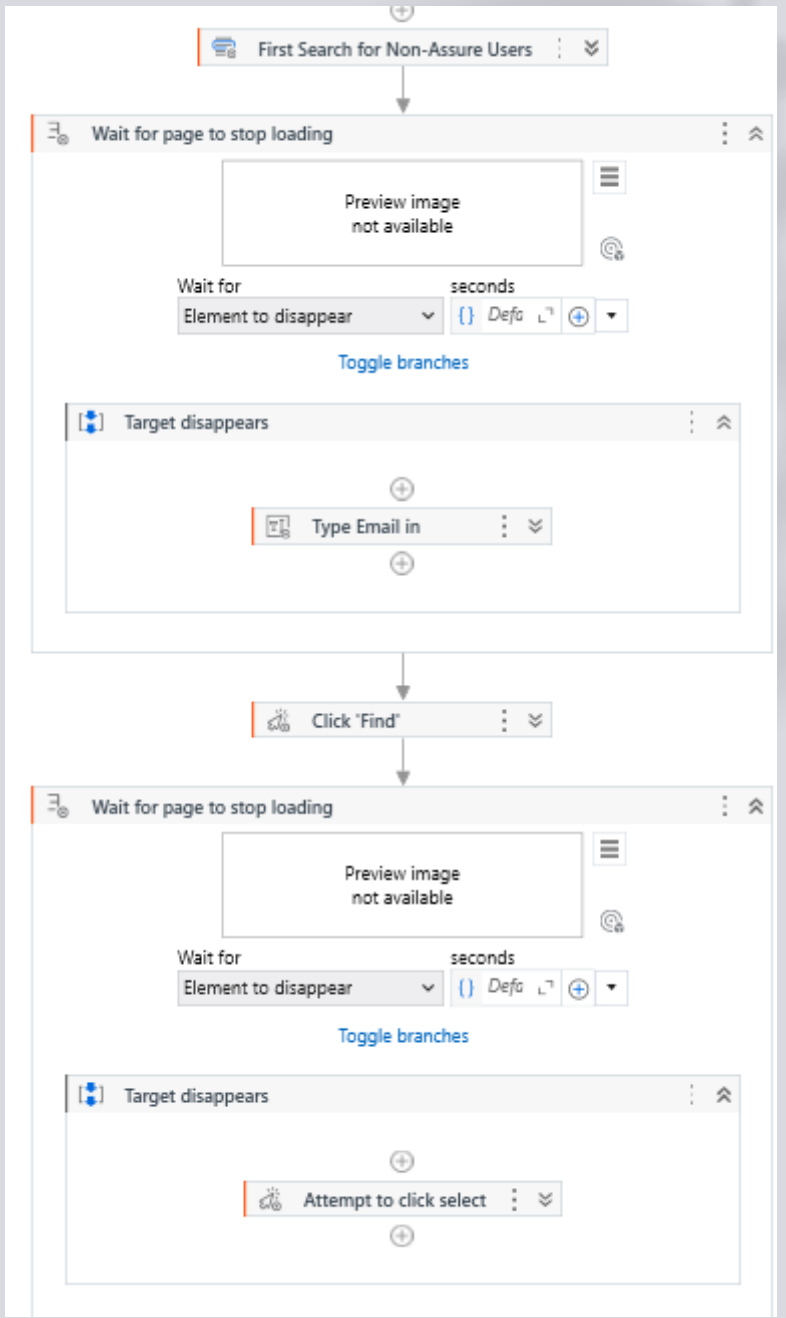
Step 2c: Add Attachments (Temp)

- This will check if the report is flagged as a photo
- If it is, then get all the attachment names from the dictionary
- One by one, add them to the report by searching in the user's downloads for the file name
- After it has been attached, delete the file



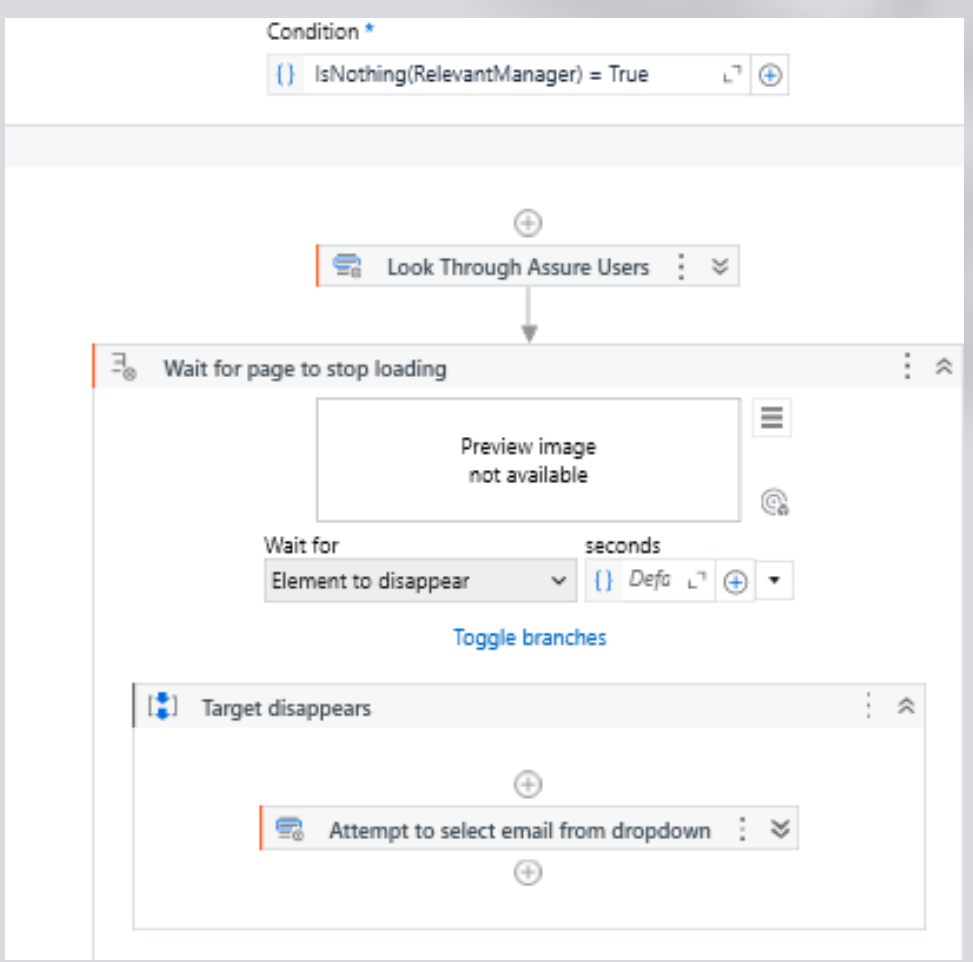
Step 2d: Find Manager Email

- First, we search through all assure users
- To do this, we type the manager email into the text box and click find
- If a manager exists, then a select button will appear
- If a select button appears, the user has been found and we return to the main process
- If not, we continue to the next slide



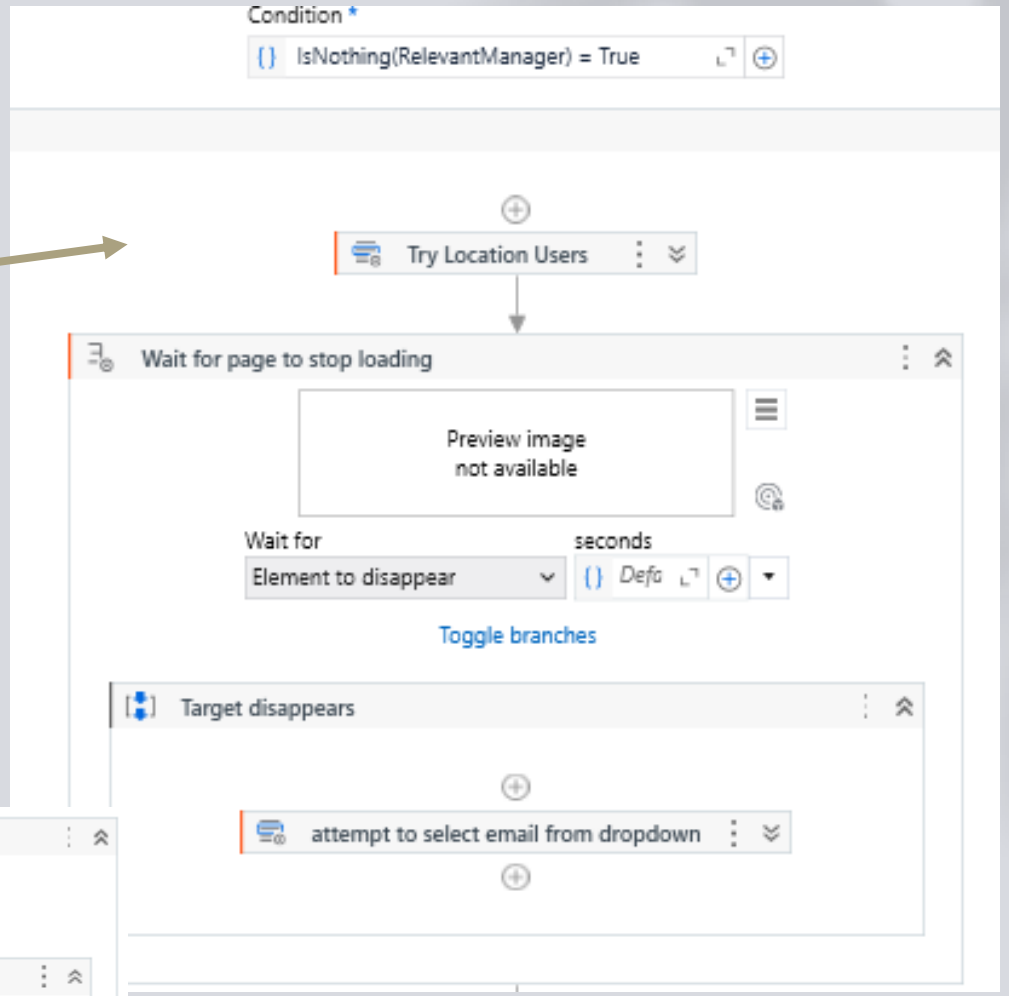
Step 2d: Find manager email

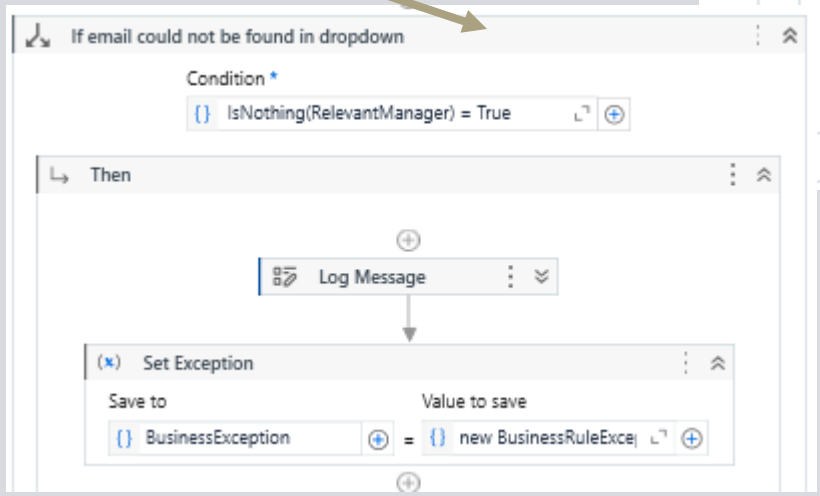
- Then we will search through Assure users
- This is a dropdown menu so we will attempt to find the email in the dropdown
- If the email is found, we can return to the main process
- If not, we continue to the next slide



Step 2d: Find manager email

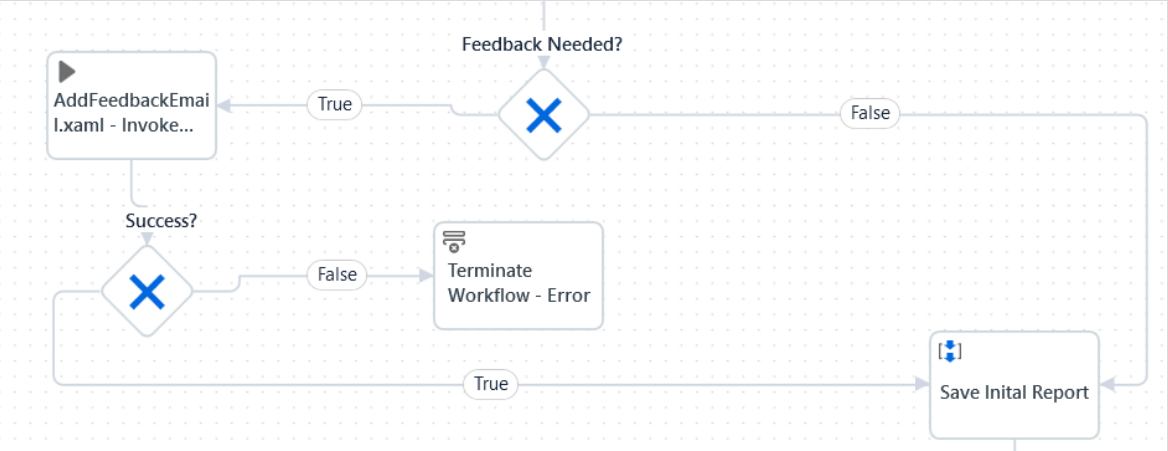
- We then try location users
- Again, this is a dropdown so we attempt to select the email
- If the user is still not found, then we throw a Business Exception





- We would then return to the main process and check this part of the flowchart
- If an error has been thrown at any stage, the workflow is terminated and passed to the exception handler
- Otherwise, it continues

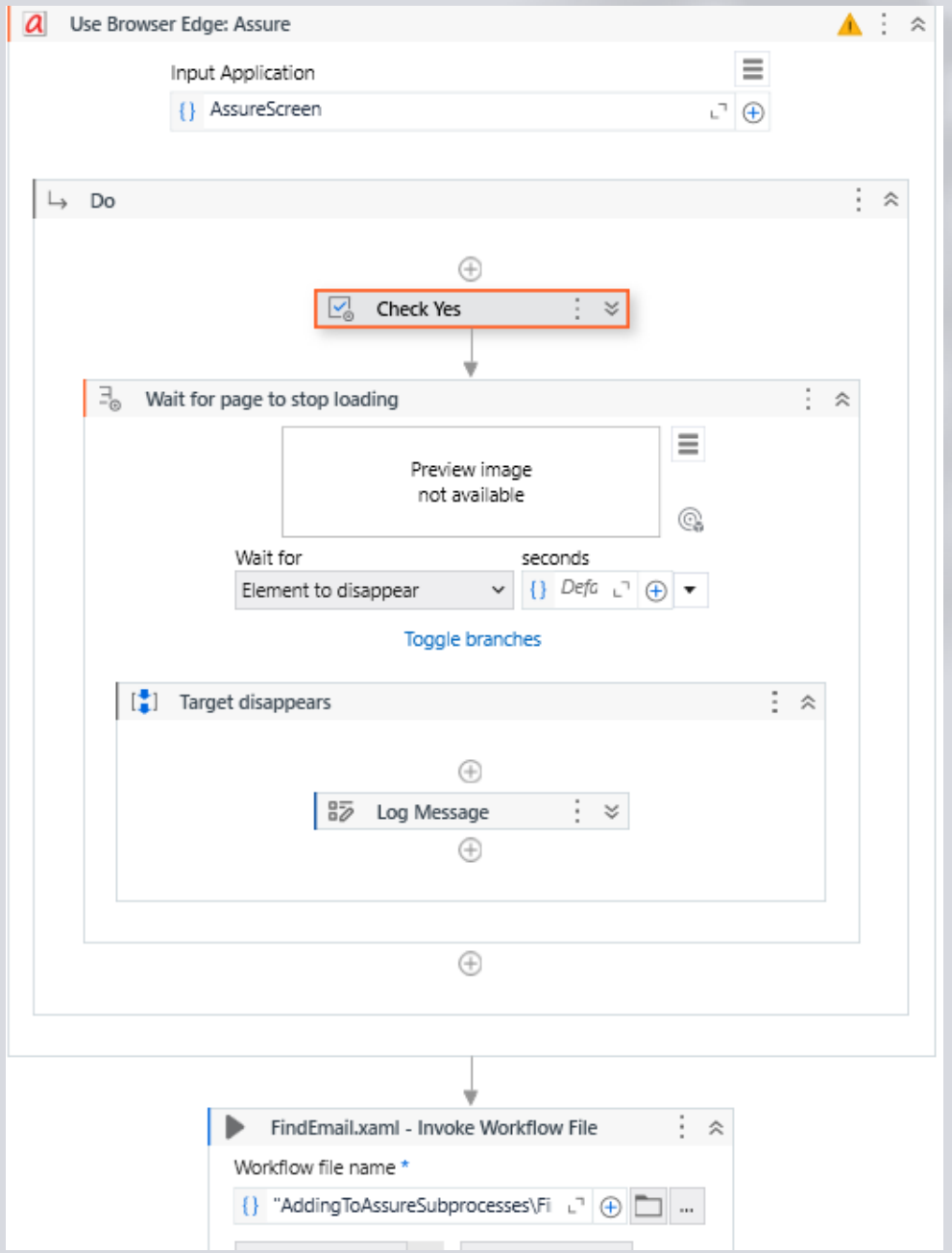
Step 3:



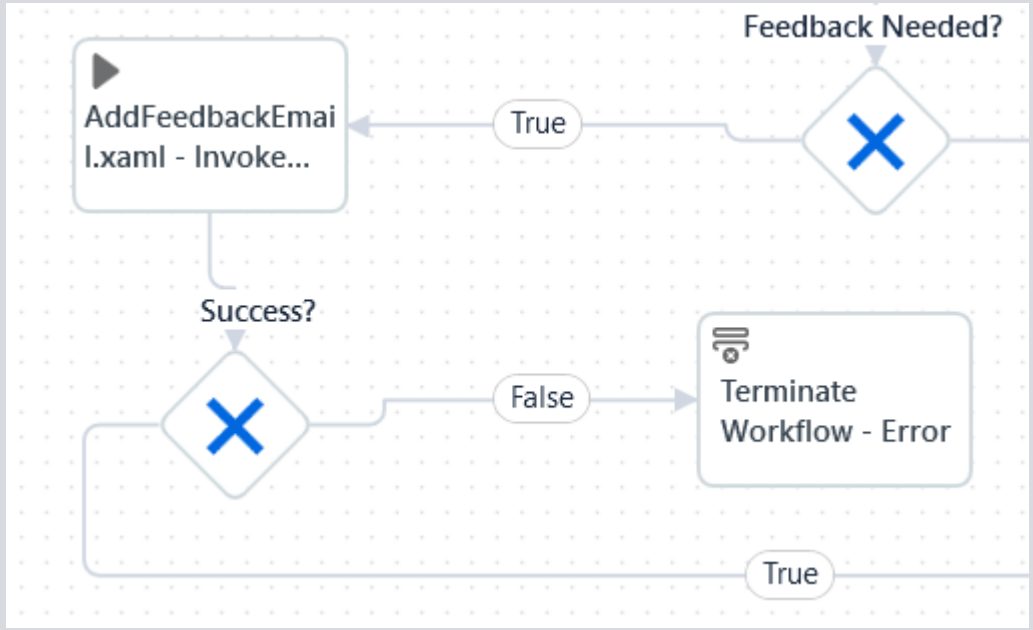
- This checks if feedback is required
- To check, we look at the ‘Feedback Required?’ field in the dictionary
- If not, we go straight to saving the initial report (Slide 100)

Step 3a:

- This step will mark the feedback required box in the Assure report as Yes
- Then we load the same ‘FindEmail’ workflow as in the Relevant Manager Email section (See slides 93-95)



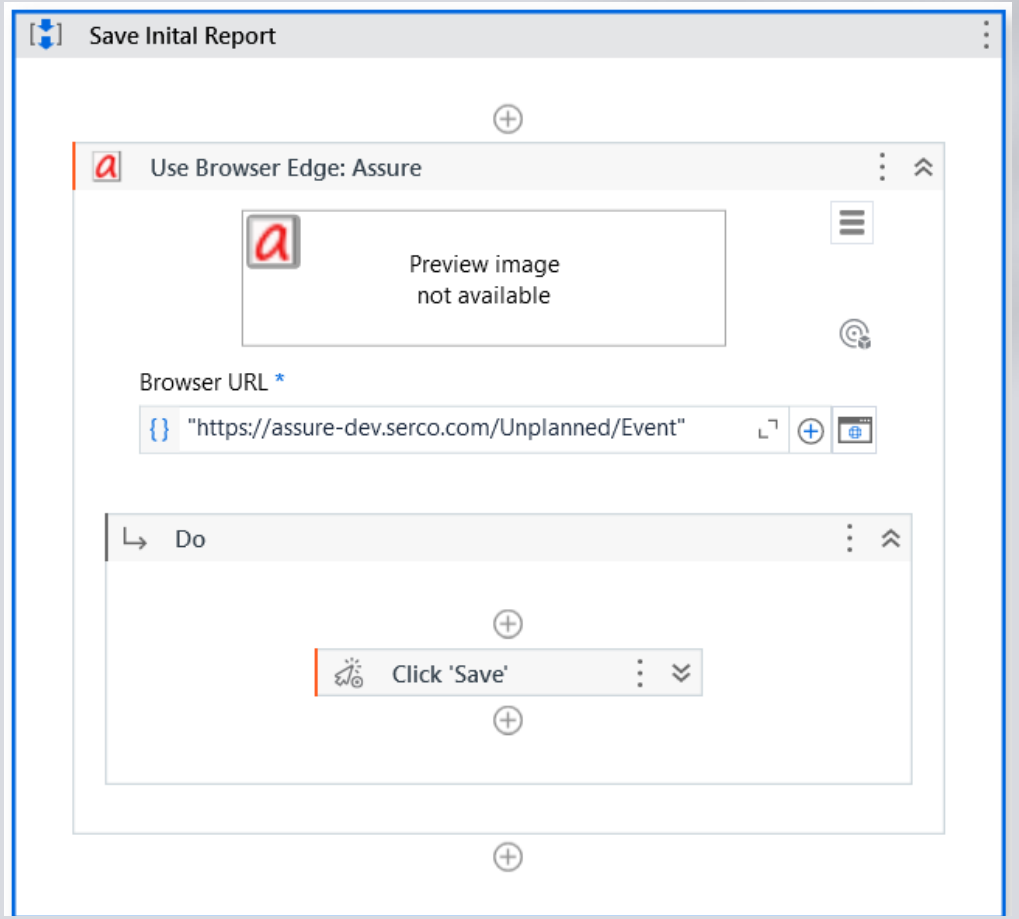
Step 3b:



- Again, if the FindEmail process fails, the workflow is terminated, and an error is thrown and will be handled by the exception handler

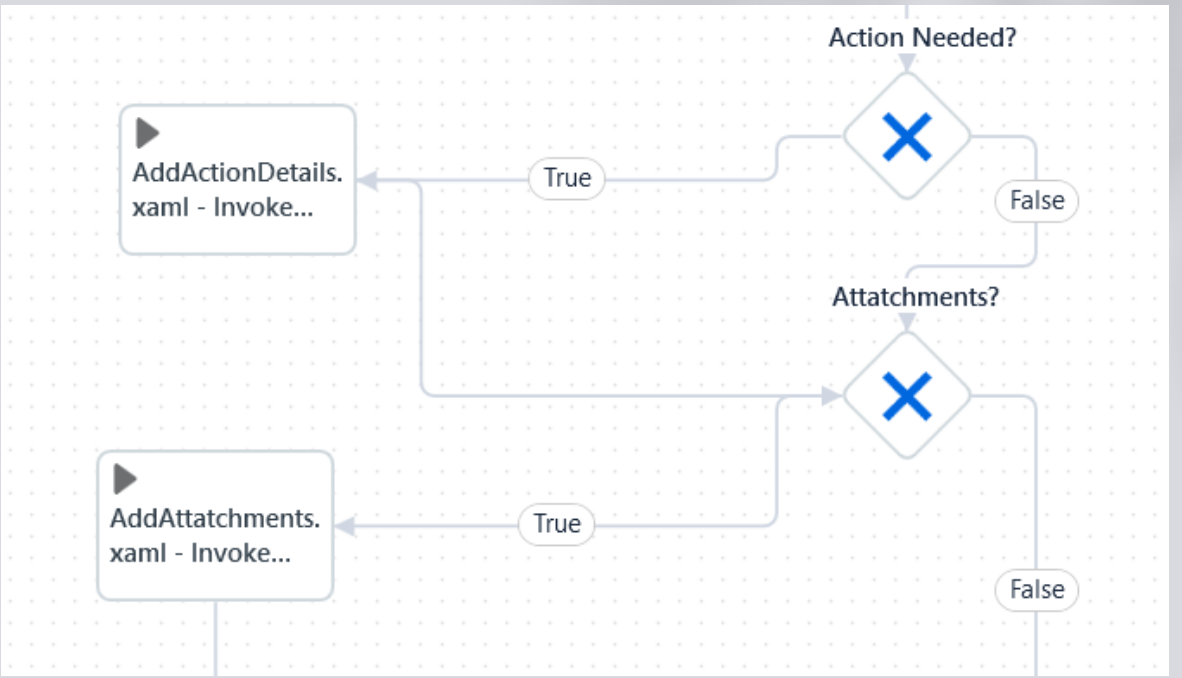
Step 4:

- Then we save the initial report in Assure



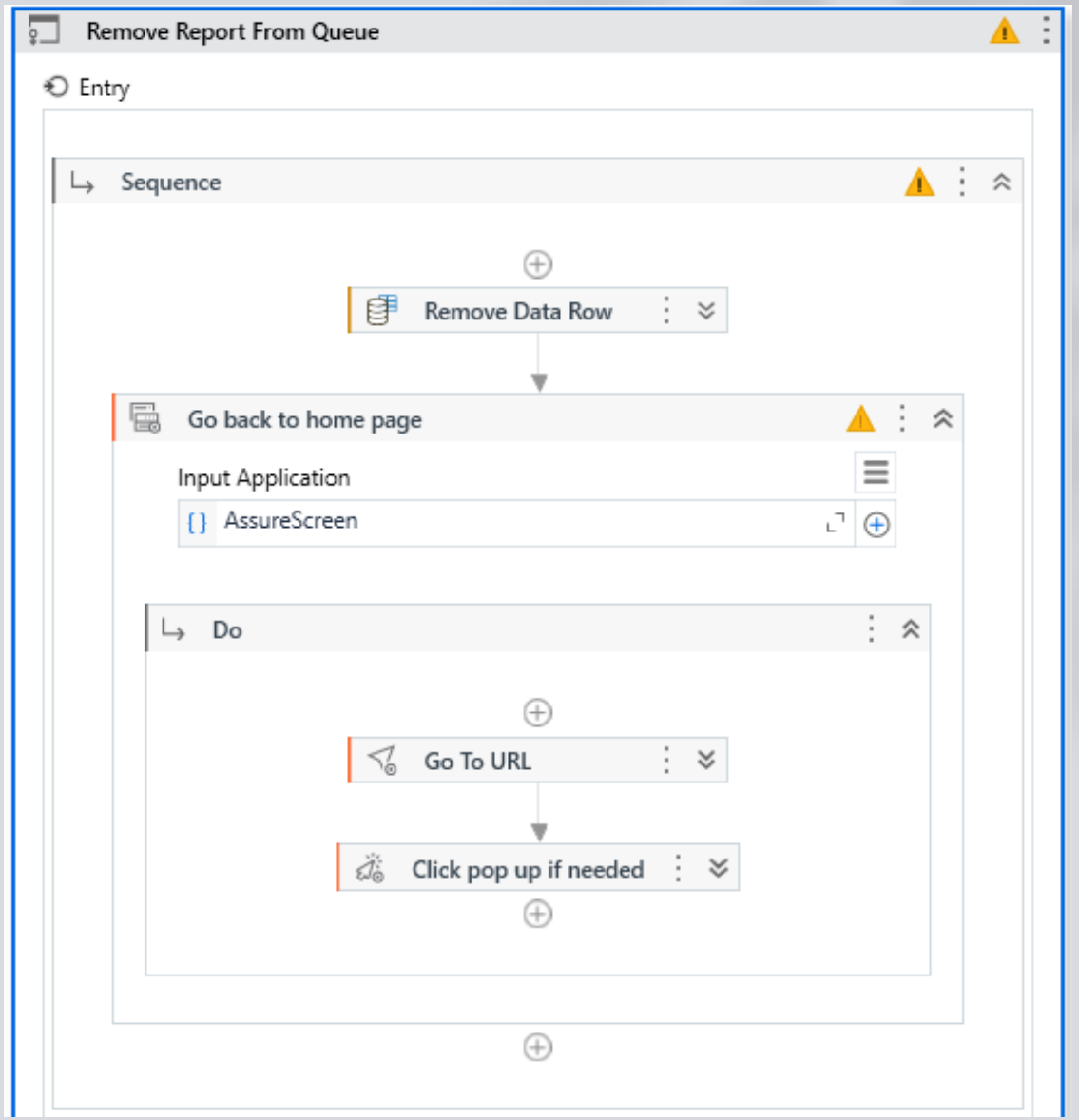
Steps 5 & 6

- Unfortunately, due to limited time and technical difficulties I wasn't able to implement these steps in the process.
- If these had been implemented, they would have added the action details into the Assure report if this was required and it would have added attachments in the correct way.

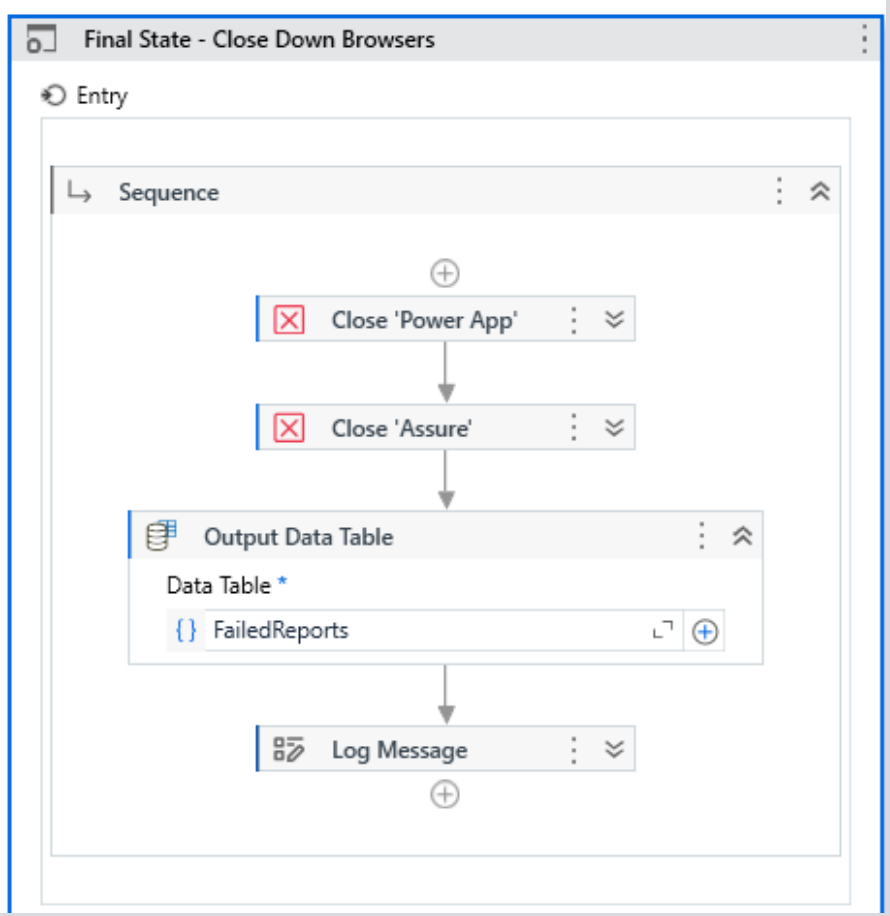


Remove report from queue

- If no errors were thrown, then we go to this state
- This will remove the completed report from the data table
- Then return to the assure home page.
- This state will also check if there are more reports left in the data table. If no reports are left, it moves to the final state. Otherwise, these steps are repeated for the next report
  - Get Report Data
  - Add Data to Assure
  - Remove report from Queue

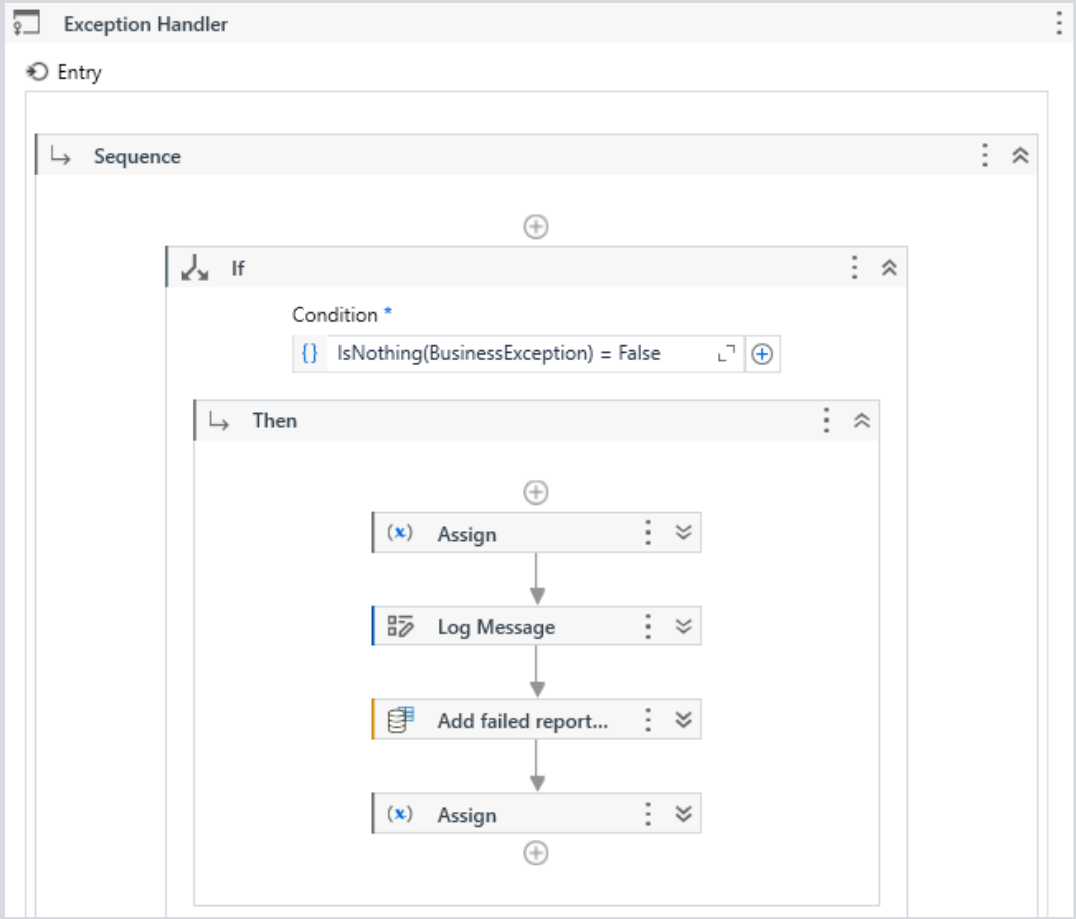


Final State



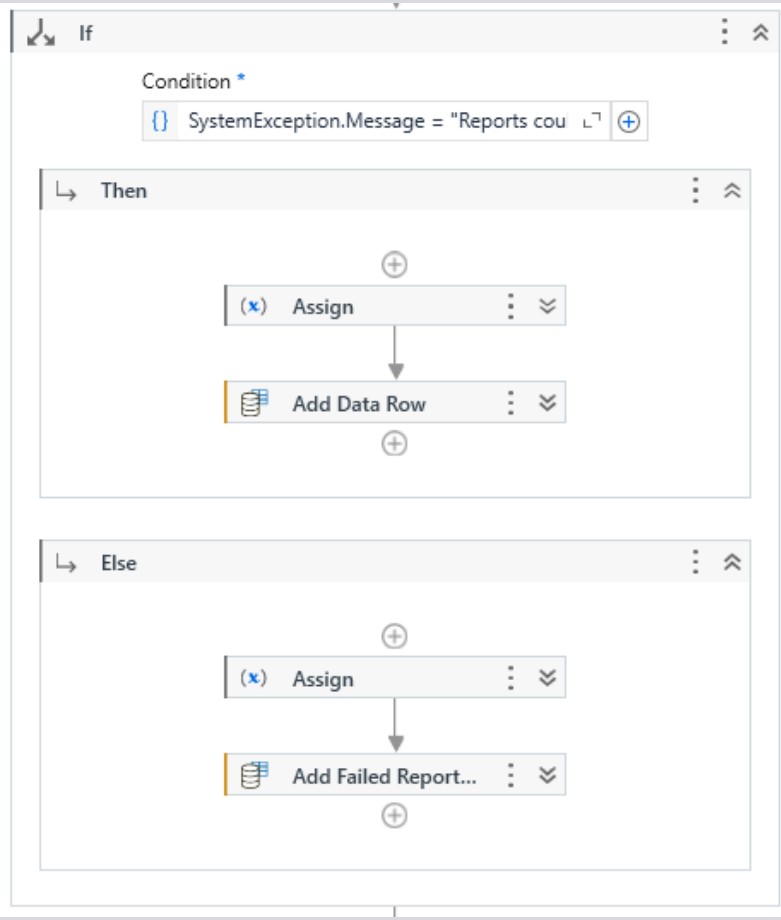
- The final state closes down both of the applications and logs the table of failed report numbers so these can be looked at manually

Error Handler



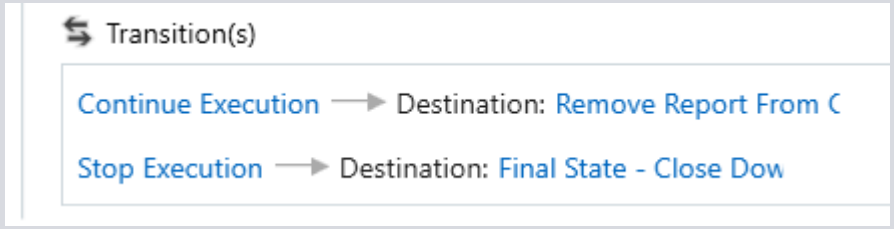
- The exception handler deals with 2 types of exceptions
  - Business Exceptions
  - System Exceptions
- This part of the exception handler deals with business exceptions
- The handler will check if it's a business exception. If it is then it will log the error message to the log and add the failed report number to the failed report number's table
- Then it will reset the business exception variable to null

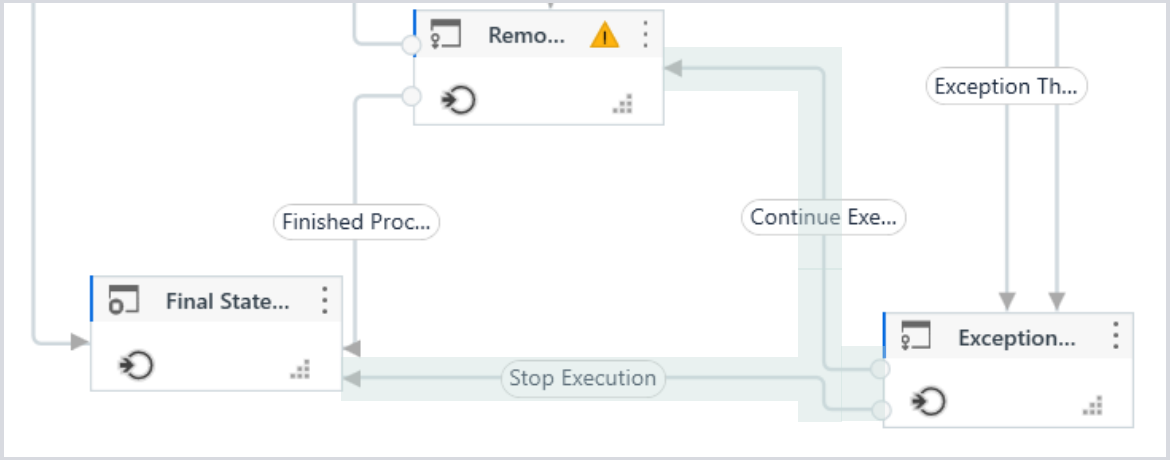
Error Handler



- This part deals with system exceptions
- It checks to see what type of system exception it is.
  - If the exception was thrown before the report numbers were collected, we add a line to the failed reports table saying no reports could be collected.
  - Otherwise, we only add the failed report number to the table

Error Handler





- The exception handler has 2 transitions
- If there was a system exception that couldn't find any reports the process cannot continue so we follow the transition to stop the execution (to the final state)
- Otherwise, we continue the process and just delete the failed report from the queue of reports to be processed.
- The program will then loop around again (if there are more reports to process) and continue attempting to process the remaining reports

Limitations of the current solution

- This solution is not fully complete
- With further work, this solution would have been able to send emails to a staff member with the failed reports and reason for failure
- It would also have been able to correctly implement the adding action information and attachment processes in Assure
- Due to limited time and technical issues with Assure, this was as far as I was able to program during this work experience program
- This solution took approximately 4 days to implement

- I used an agile method of programming and initially implemented a basic version of this code that often failed and had no exception handling. The code itself also had no structure to it so was hard to read and understand.
- Therefore, I took the time to improve my solution by restructuring the code using state machines, flowcharts, and individual workflows.
- I also added error handling and ways for the program to automatically retry certain steps of the process which often caused the failures. This meant that my final solution run much more reliably and could be more easily understood by the main developer of this process.