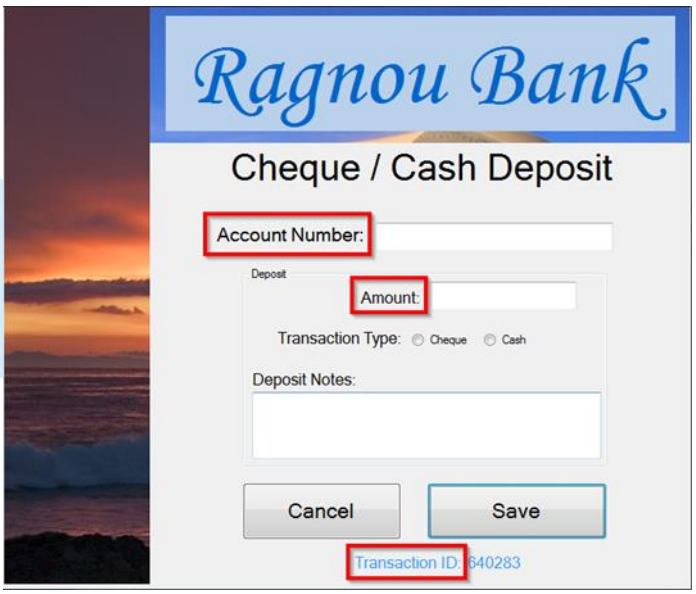
Module 7: Orchestrator CE & Other RPA Tools

Demo 2 – Solution

Demo 2

Remote Data Entry Application

- Create an automation to enter bank details into the "Ragnou Bank" Application. The data entry should be divided among multiple robots. Finally, the bank should also be able to consolidate all the transactions in one file, for audit purposes
    - The input data will be inside a CSV file in the format "Account", "Amount"
    - Use Queues to create the automation
    - The final excel file should be of the format, "Account", "Amount", "TXN IDs"
    - By default, assume all the transactions will be of type "Cheque"



Solution Inputs:

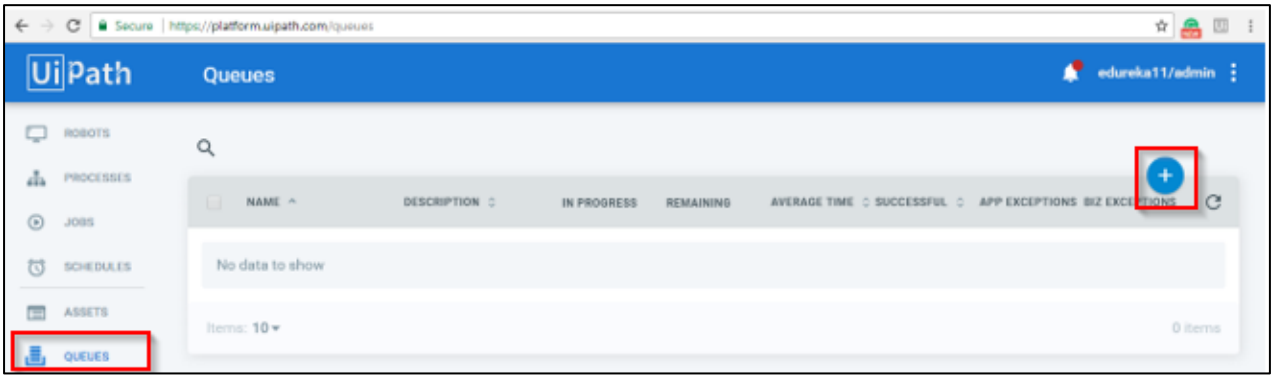
- Consider using Queues in Orchestrator

Link to download the Bank Application:

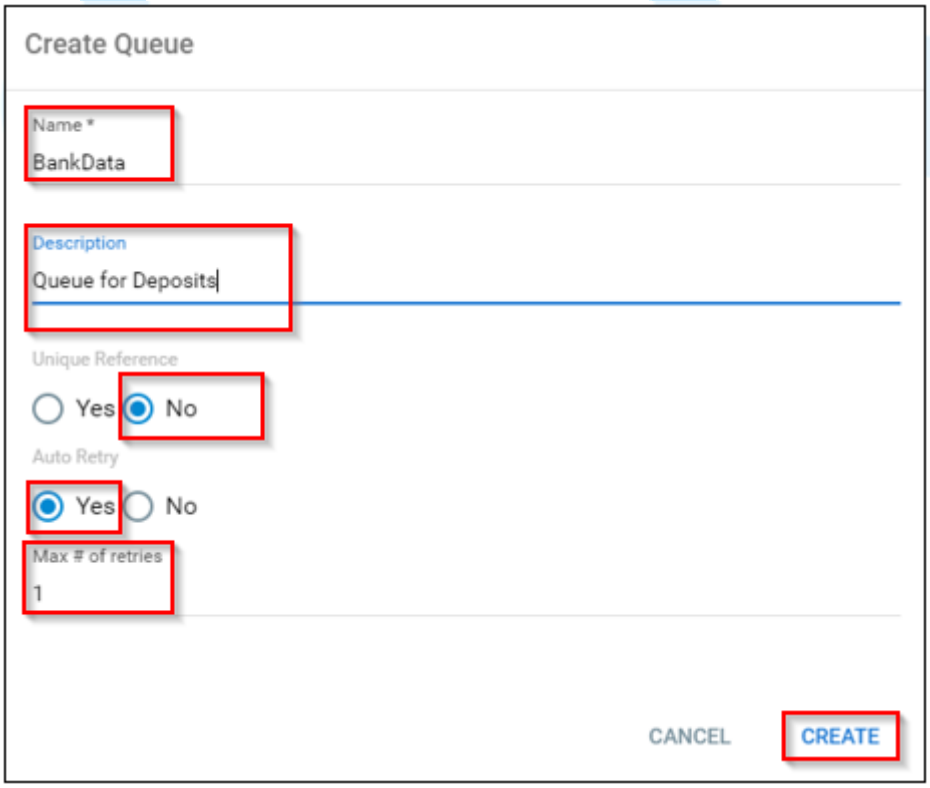
https://edureka.wistia.com/medias/z7gqzr7ouf/download?media_file_id=248534619

Demo 2 – Solution

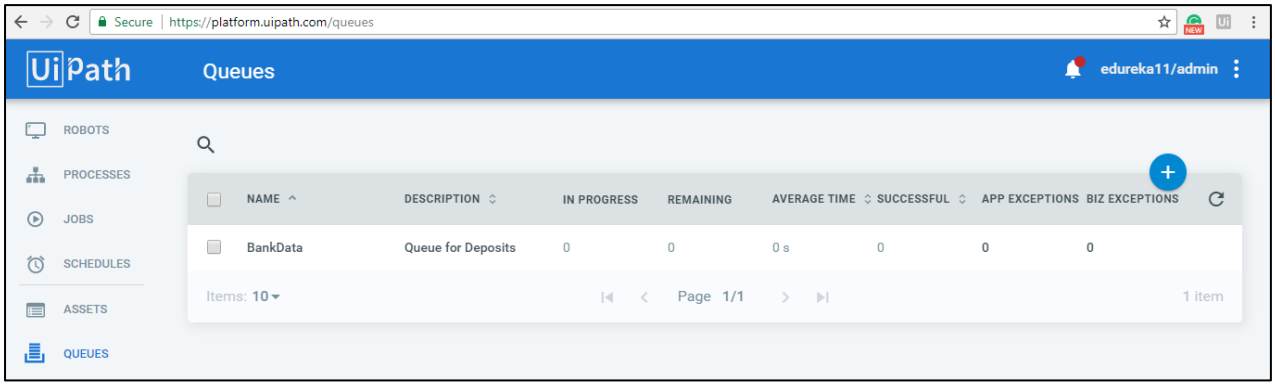
➢ Step 1: Create a queue in the UiPath Orchestrator using the add button in the Queues section



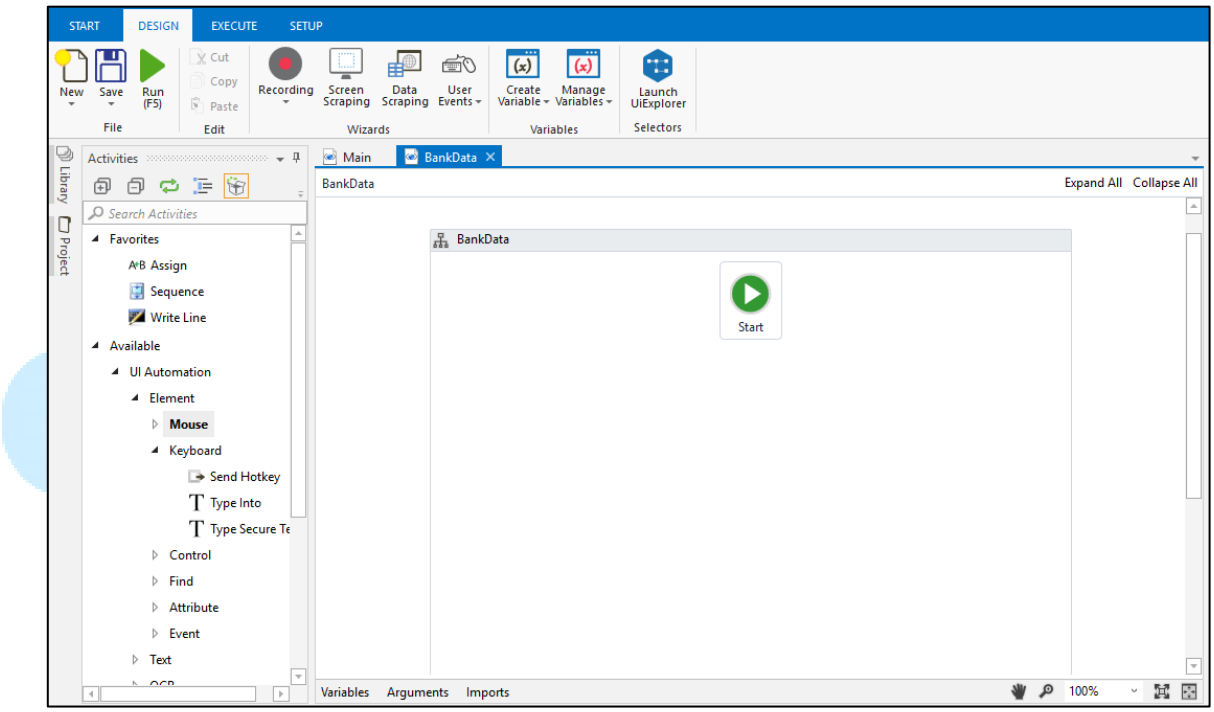
1. Create a queue with the following specifications



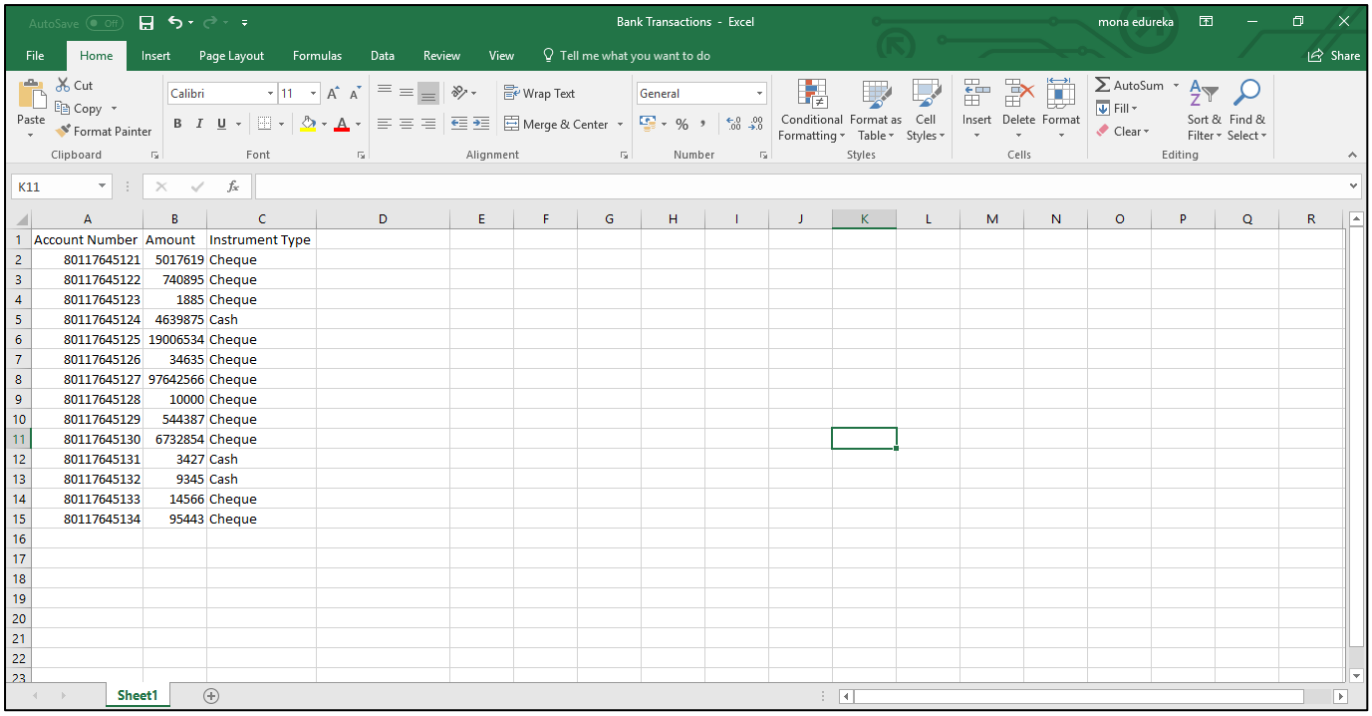
2. We can see that the queue is created



➢ Step 2: Create a new flowchart in UiPath Studio with name **BankData**

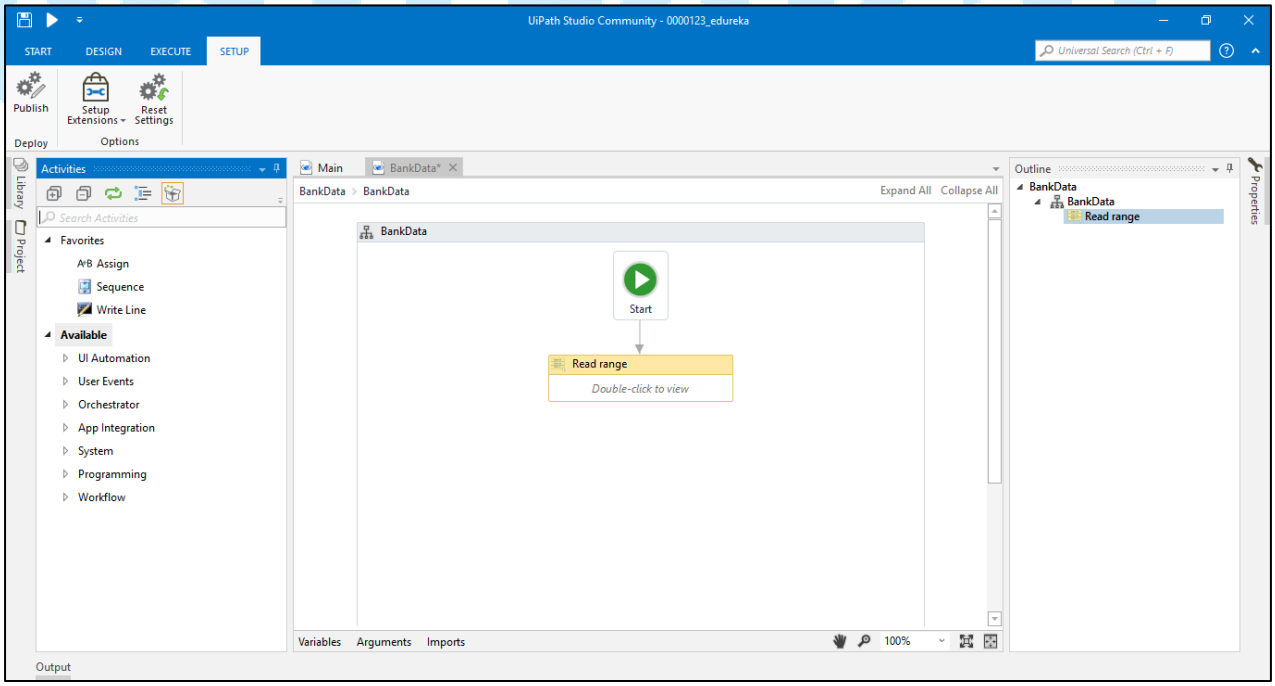


➢ Step 3: We will create a sample excel sheet with name **Bank Transactions** for the bank, which includes **Account Number**, **Amount** and **Transaction Type**



	A	B	C
1	Account Number	Amount	Instrument Type
2	80117645121	5017619	Cheque
3	80117645122	740895	Cheque
4	80117645123	1885	Cheque
5	80117645124	4639875	Cash
6	80117645125	19006534	Cheque
7	80117645126	34635	Cheque
8	80117645127	97642566	Cheque
9	80117645128	10000	Cheque
10	80117645129	544387	Cheque
11	80117645130	6732854	Cheque
12	80117645131	3427	Cash
13	80117645132	9345	Cash
14	80117645133	14566	Cheque
15	80117645134	95443	Cheque

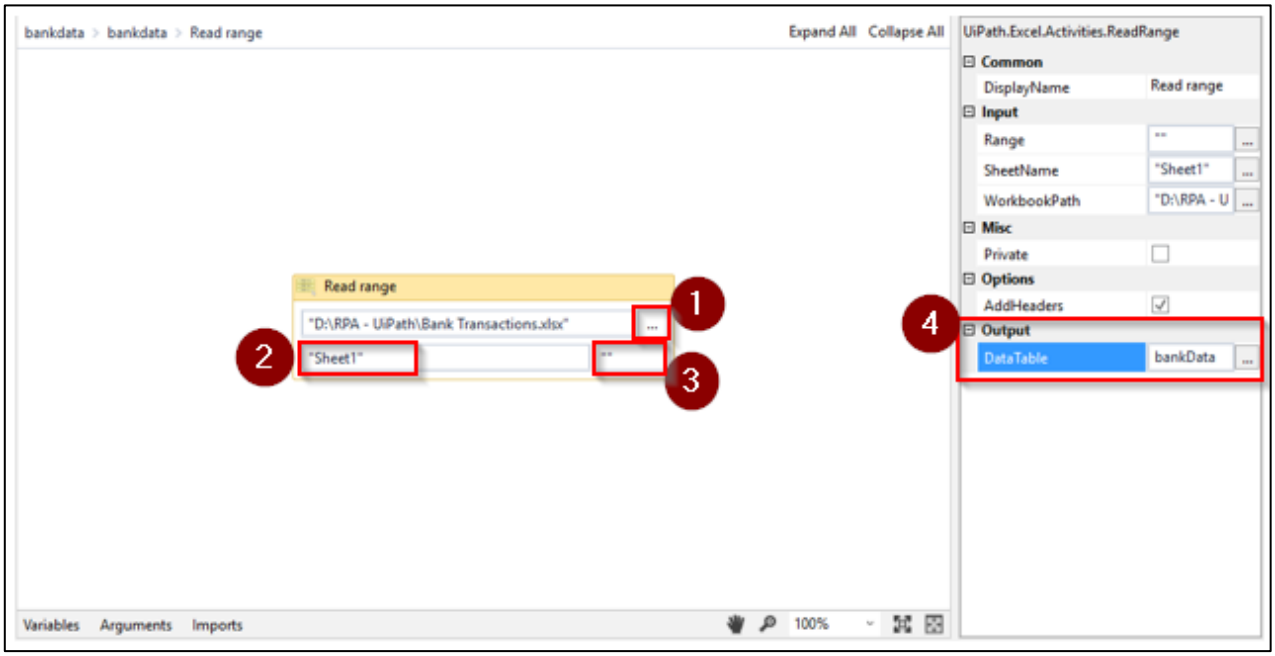
➢ Step 4: Add **Read Range** Activity in the BankData flowchart, which is present under System >File > Workbook > Read Range



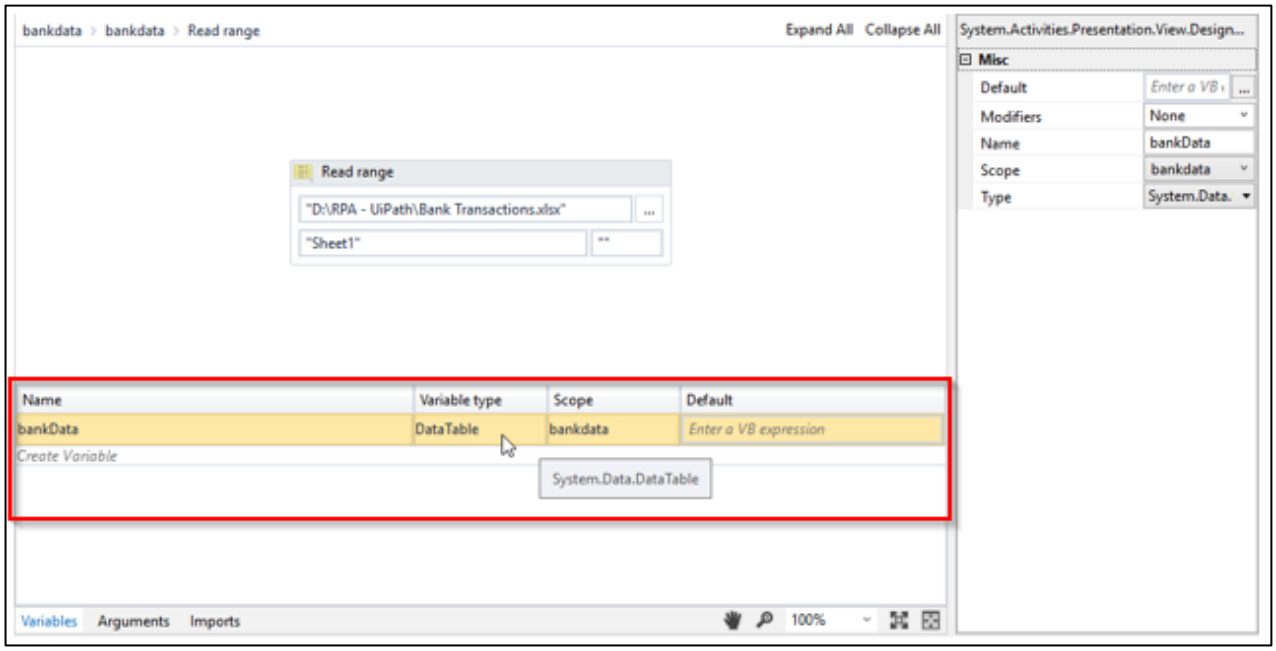
➢ Step 5: Double click the **Read Range** Activity to open it

1. Provide path of the excel sheet inside the activity
2. Select the sheet of the excel file in use

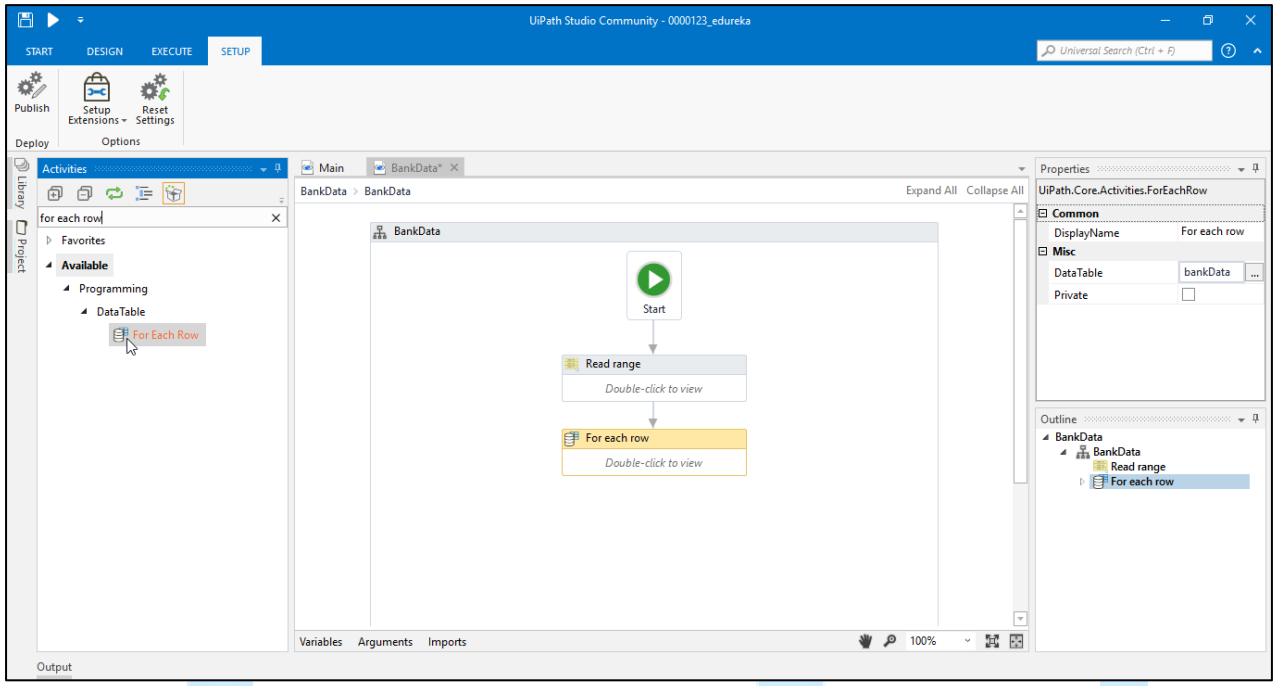
3. Remove the Range given in the box and give blank " ", in order to select the whole range



4. Create a DataTable variable (name - **bankData**) which will store the data of all the transactions made from the bank's application form along with the Transaction IDs. Add this variable inside the Output section of the Read Range Property

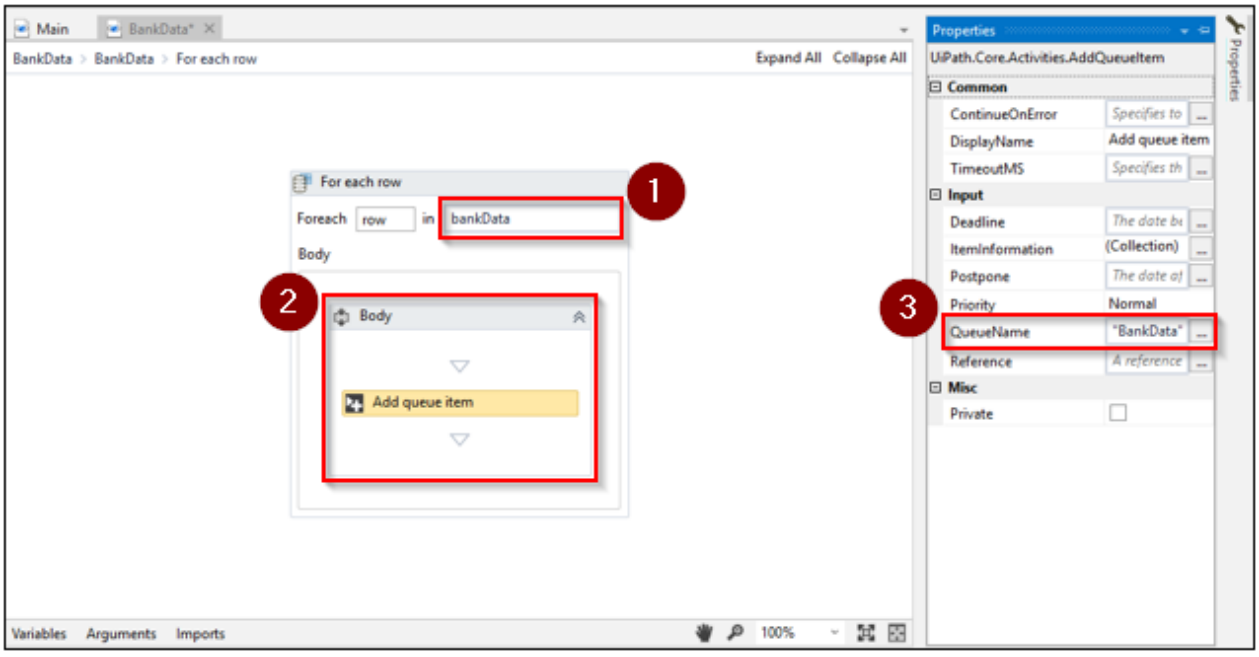


➢ Step 6: Add **For Each Row** Activity to add the data of each row into the queue

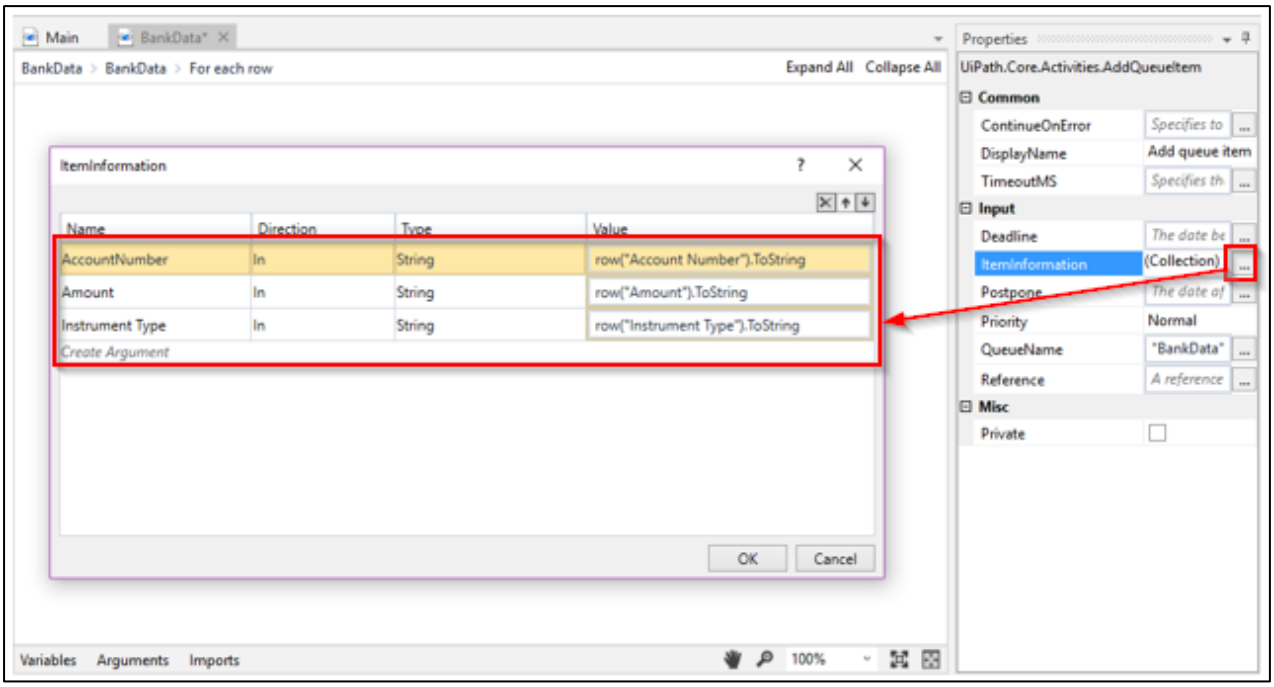


➢ Step 7: Double Click the Activity to open it

1. Write bankData in the box
2. In the body section add **Add Queue Item** Activity
3. Add the QueueName that we have provided in the Orchestrator

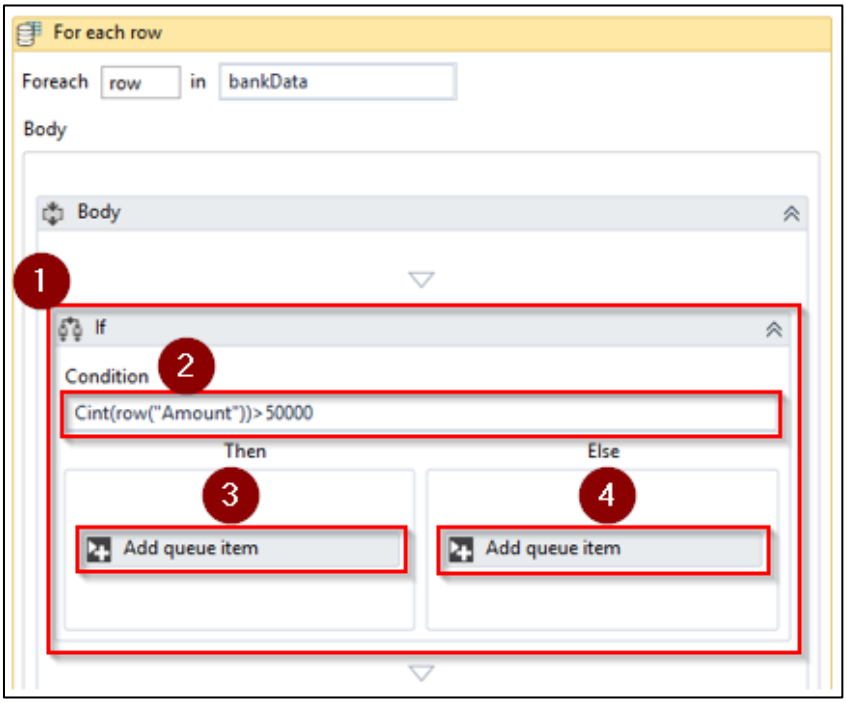


➢ Step 8: Edit the **ItemInformation** section from the properties of **Add Queue Item** and add the following variables, to create row in bankData Data Table

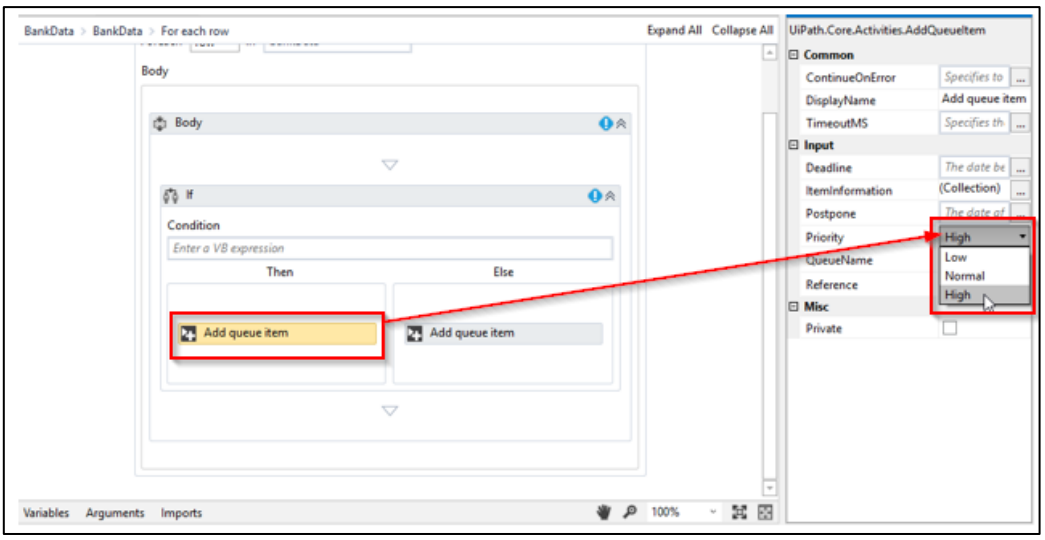


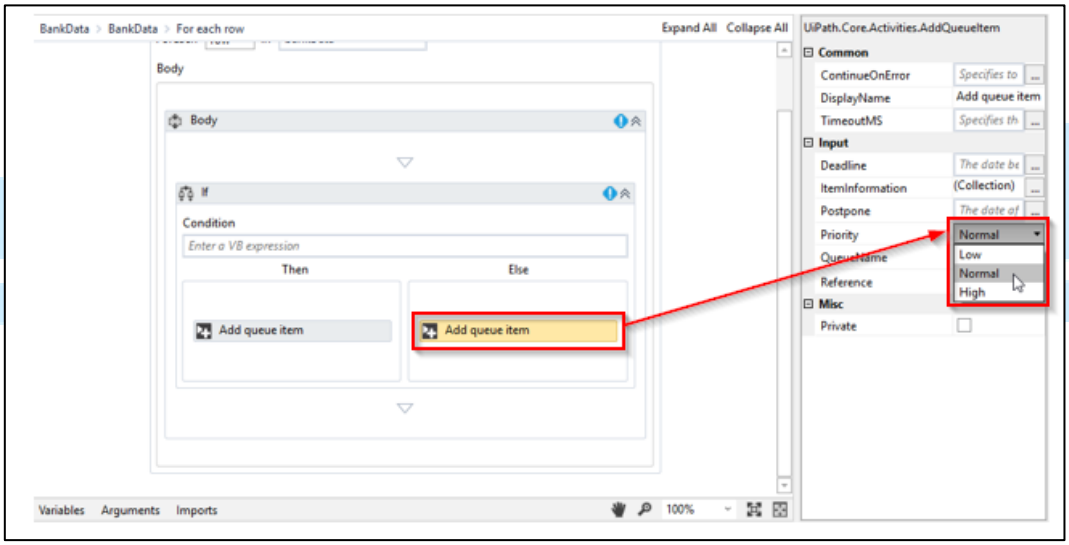
➢ Step 9: Configure the Body section **For Each Row** Activity as Below

1. Insert If Condition
2. Provide Condition as: Cint(row("Amount"))>50000
3. Now place the **Add Queue Item** Activity created before, inside the **Then** section
4. Copy the same **Add Queue Item** and place inside the **Else** Section too

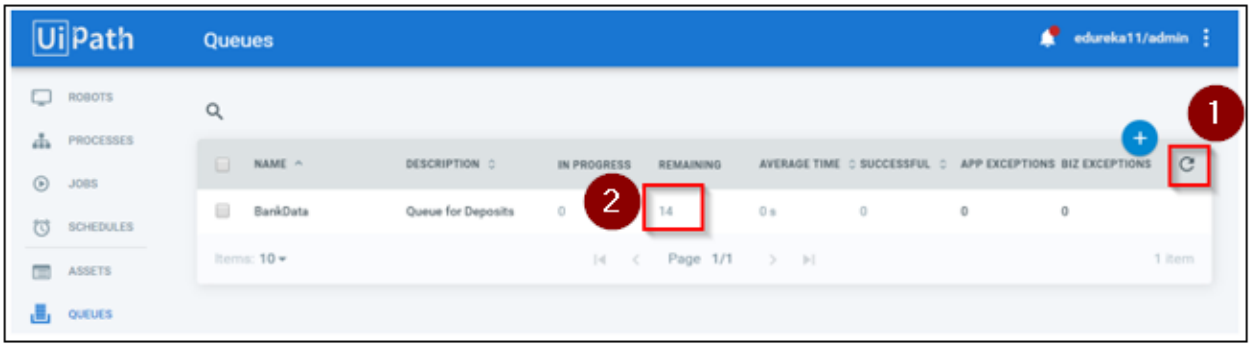


- Step 10: In the Properties of **Add Queue Item**, which is in **Then** part make priority as **High**. For **Add Queue Item** in **Else** part make priority as **Normal**

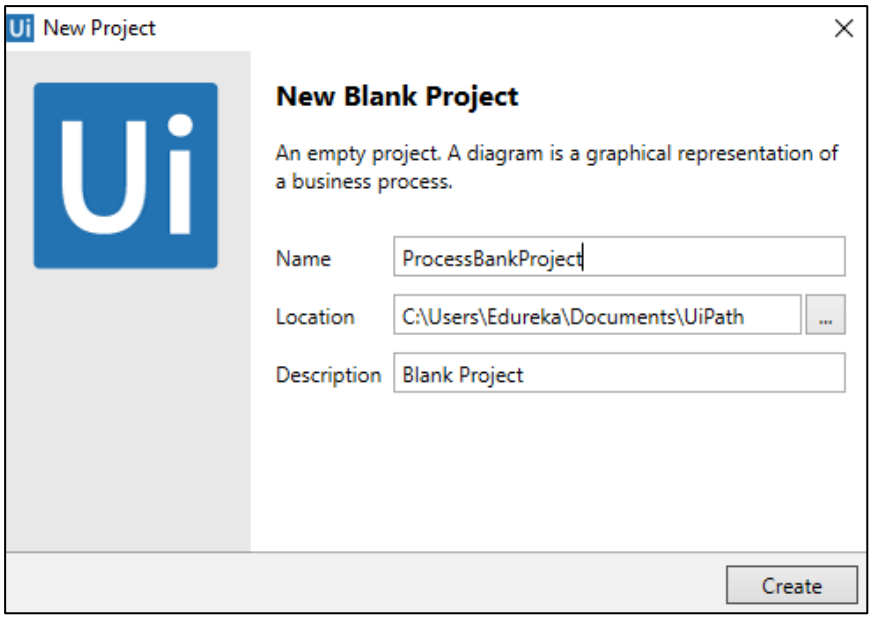




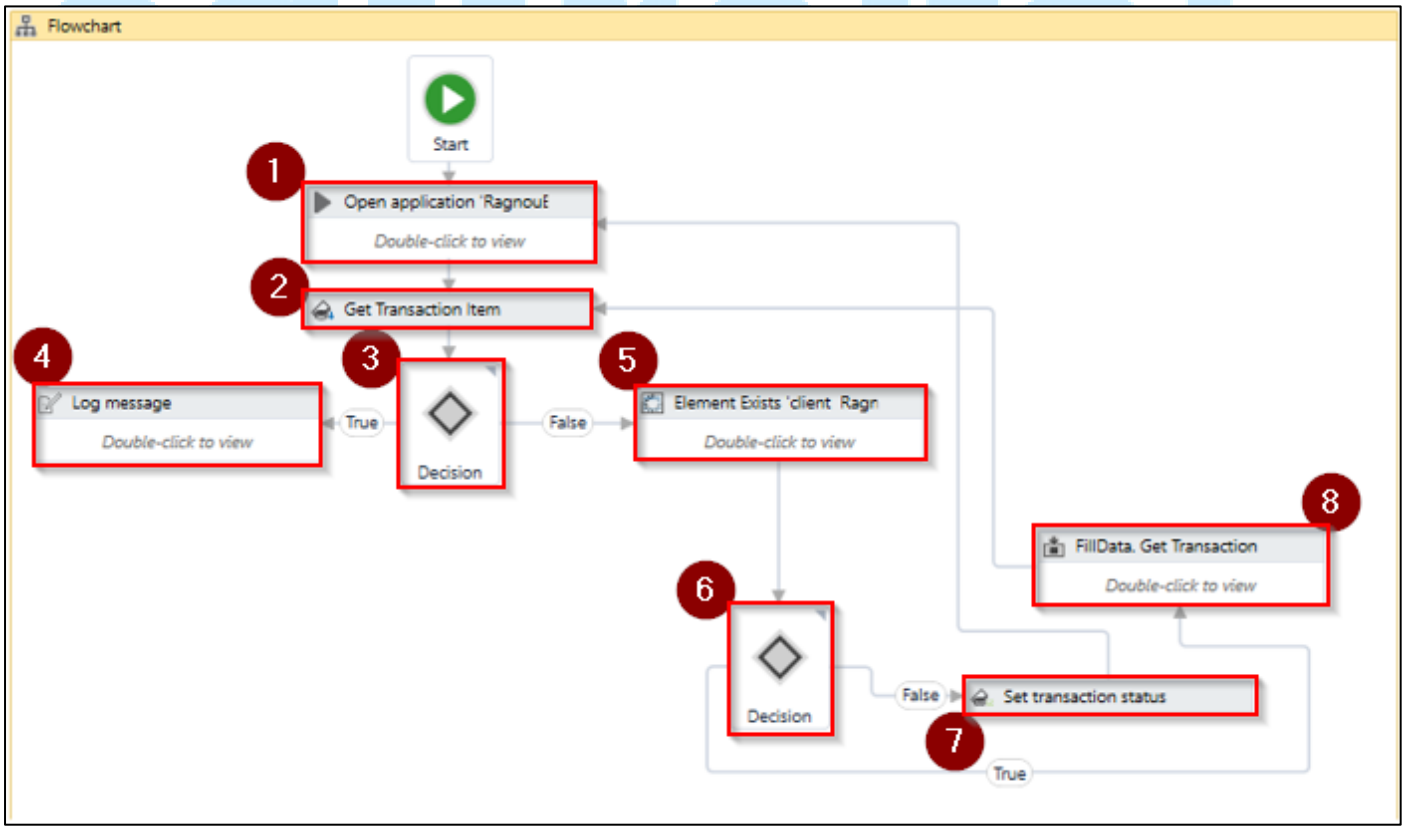
- Step 11: Run the project
  1. Go to the Orchestrator and refresh the Queue
  2. Once the execution is done we can see that the data is added to the Queue



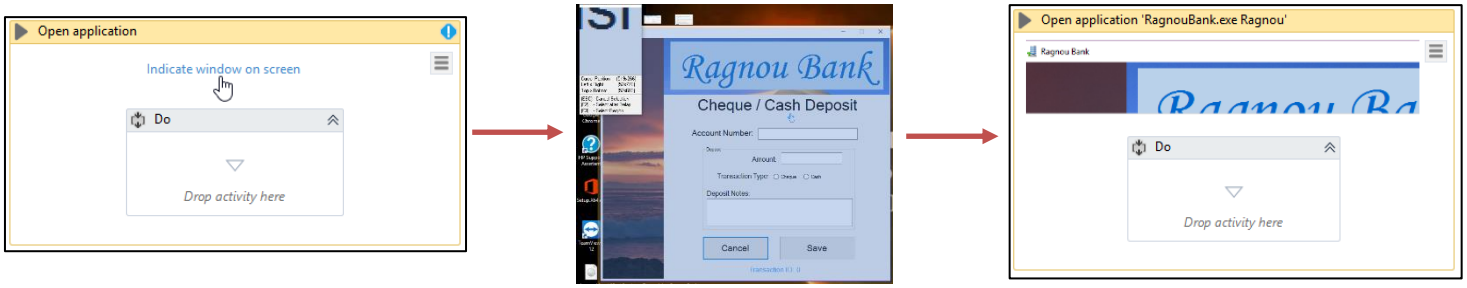
- Step 12: Again create a new Blank Project in UiPath Studio with name **ProcessBankProject**

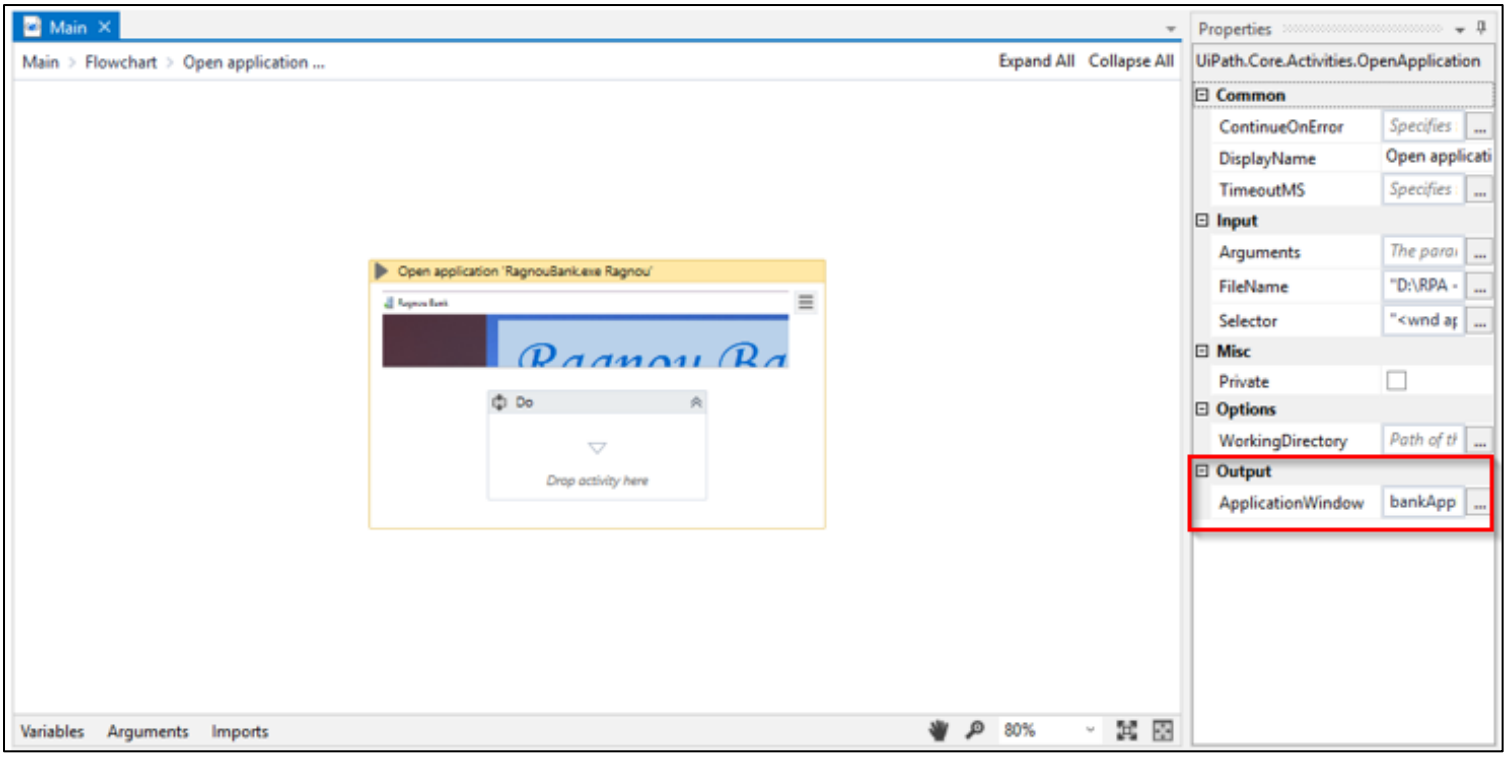


- Step 13: Create a Flowchart in that project as shown below, we will discuss each step one by one

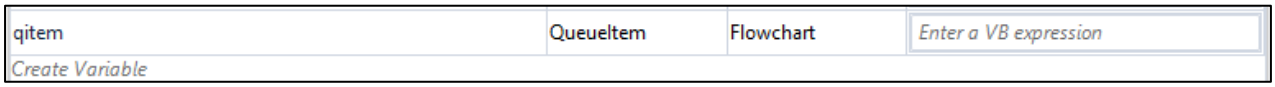


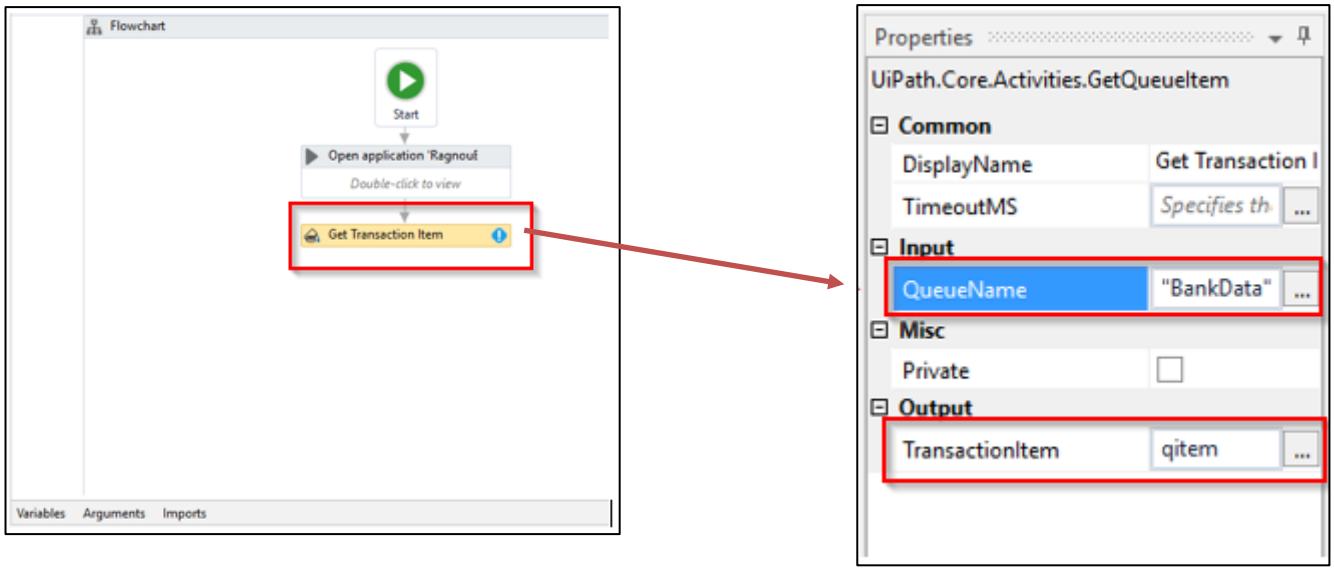
1. Create an **Open Application** Activity to open the Ragnou Bank application. Double click the Activity to open, click “indicate window on screen” to add the application window. In the Properties panel, create a variable named as **bankAppl** to save this Application window for future use



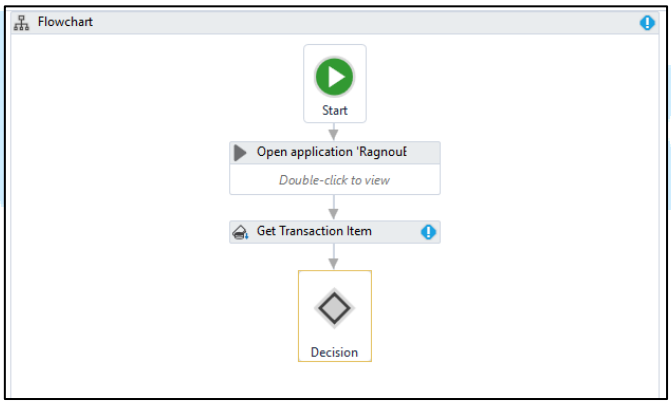


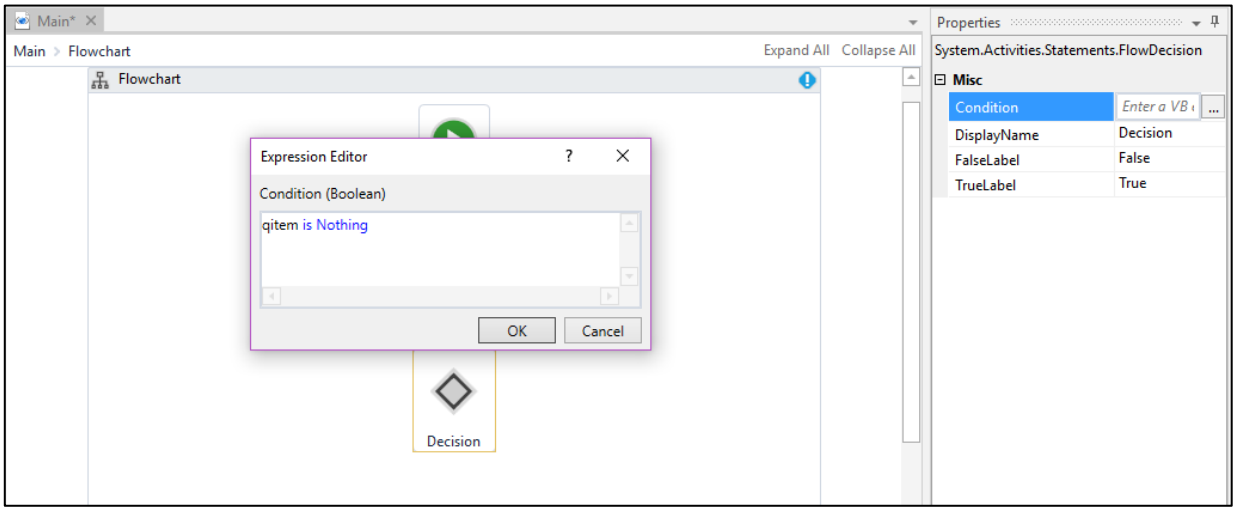
2. Choose Activity **Get Transition Item**, which will add items to queues. Configure the properties by providing the **QueueName** (as we have given in Orchestrator). Create a variable **qitem** for the items in the queue, put this variable in **Output** section



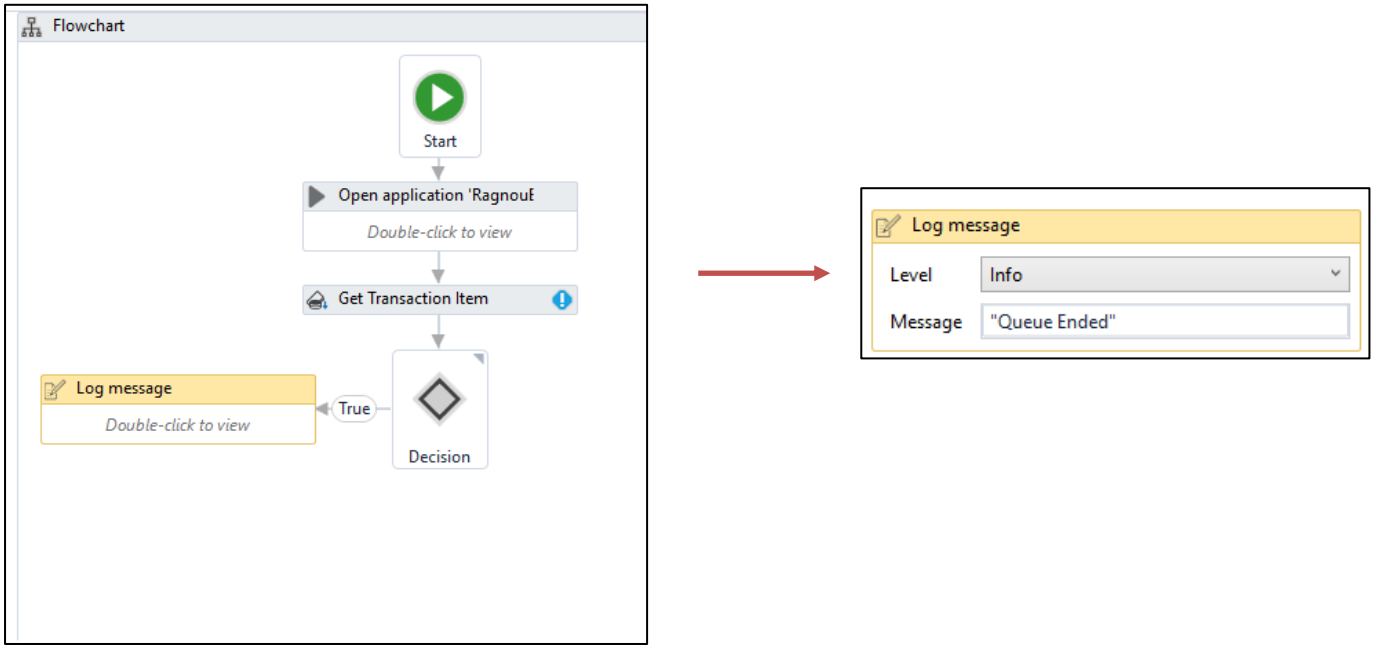


3. Add a decision box and inside the Condition box add “qitem is Nothing” (Nothing is a Keyword)

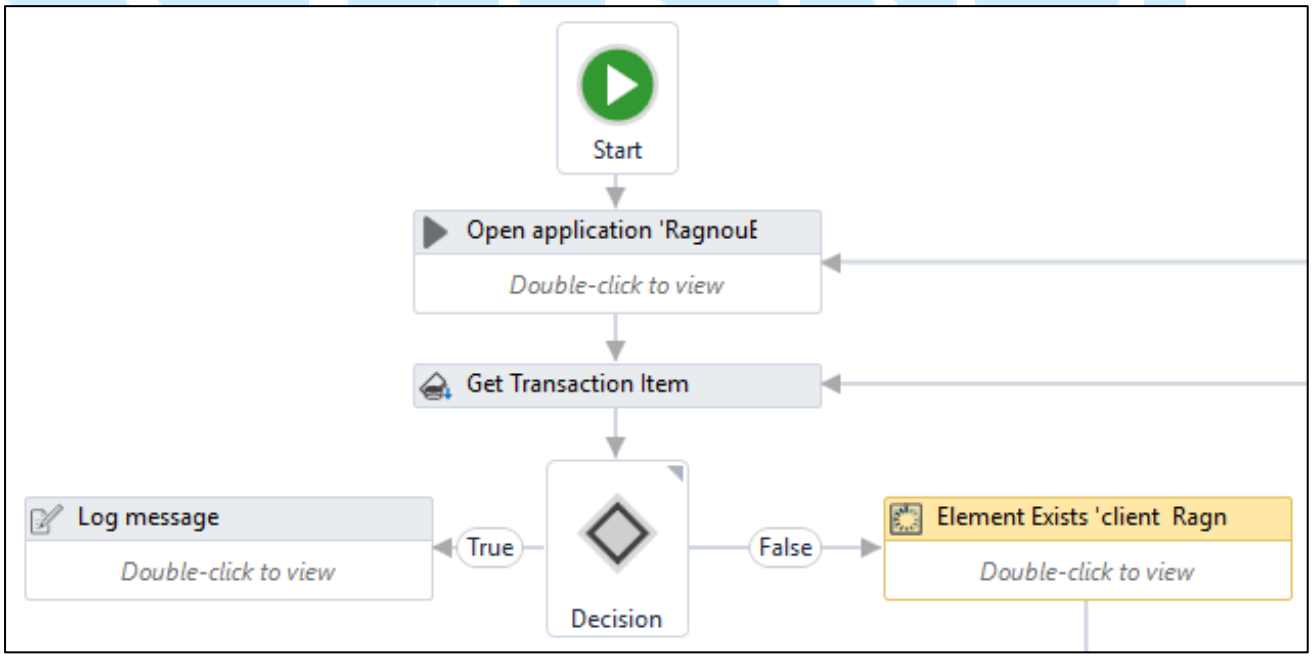




4. If the condition will be true i.e. if no item is present in the queue, then the process will end. For this add a **Log Message** Activity. Put a message inside the activity

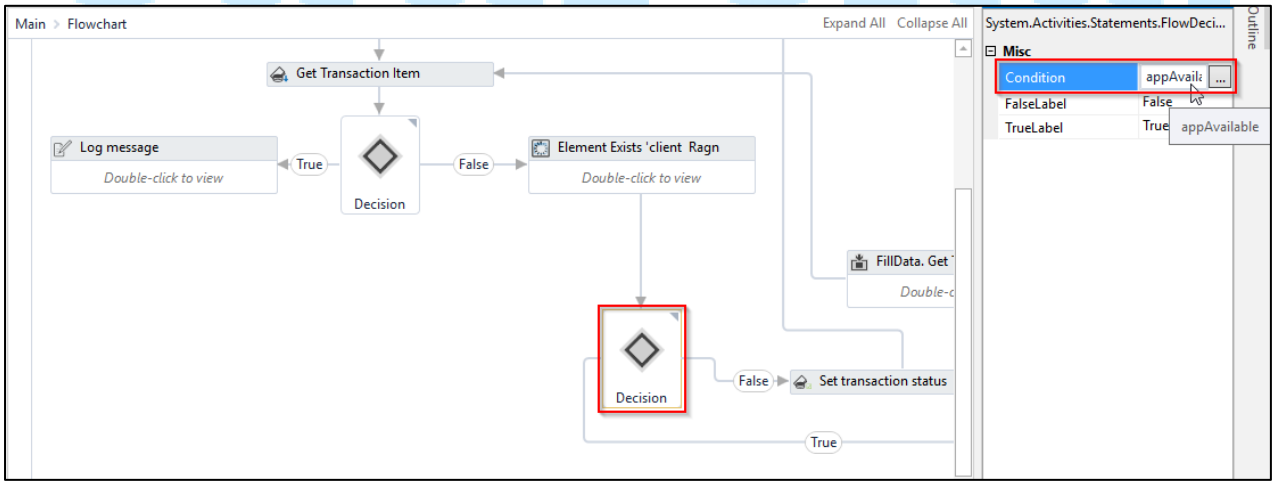


5. If the condition is true, add an **Element Exists** Activity which will find the bank Application and open for further processes

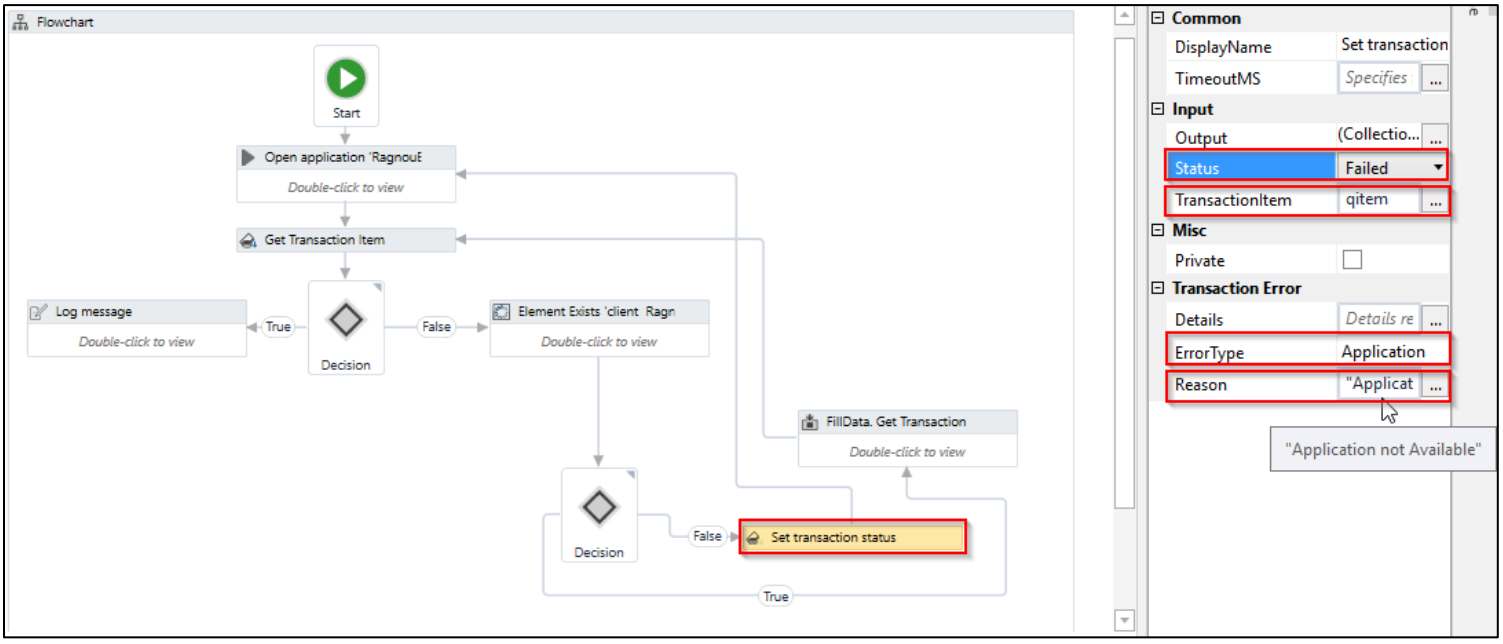


Select the application page using “indicate on screen”. Create a variable in the output section using ctrl +k, provide name as **appAvailable**

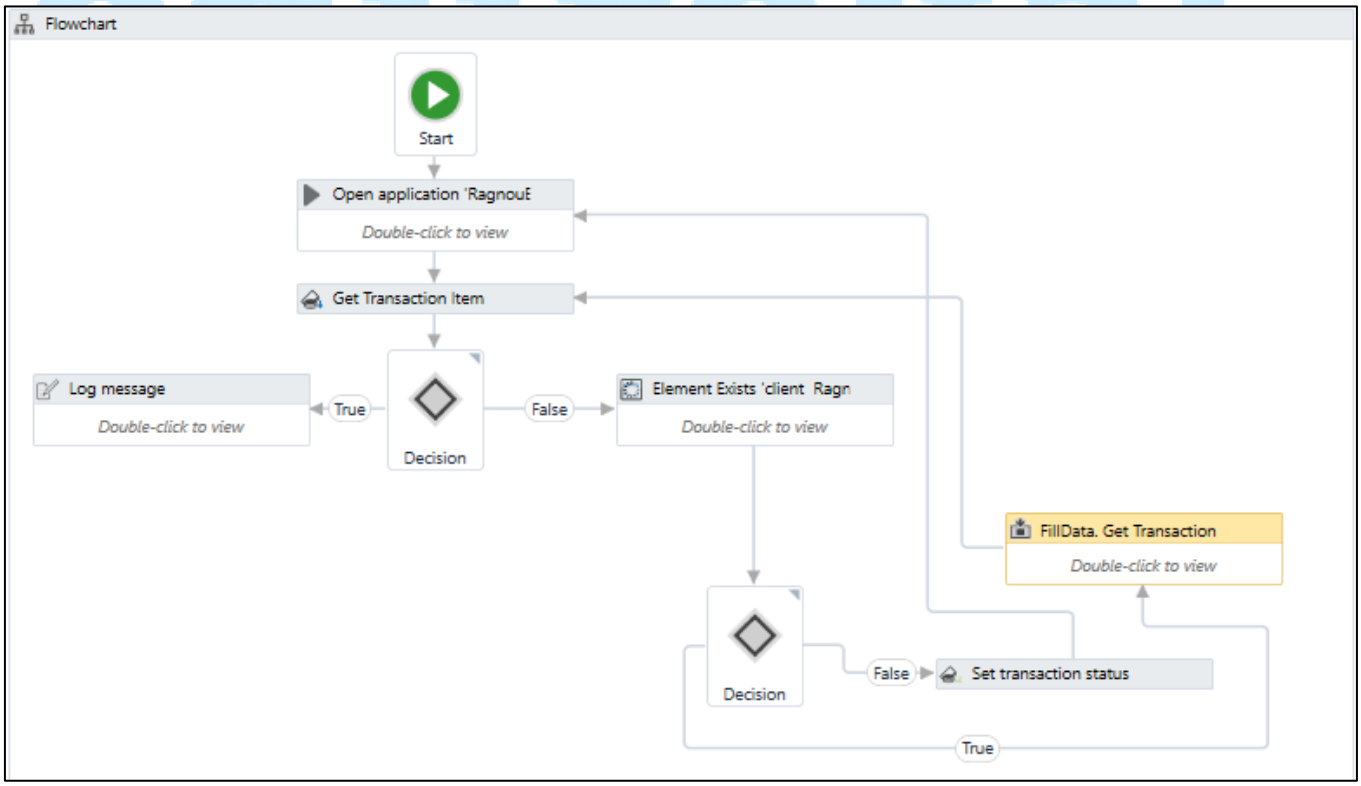
6. Attach a decision box with the above activity. In the condition box add the variable **appAvailable**, this is because if the application will be available then we will decide what we have to do further



7. If the condition is False i.e. if the application is not available, add a **Set Transaction Status**
   Configure the Properties panel by providing **Status** as **Failed**, **TransactionItem** as **qitem**, **Error** type as **Application** and give a **Reason** like “**Application not Available**”
   Connect this Activity back to **Open Application,** to start the process again

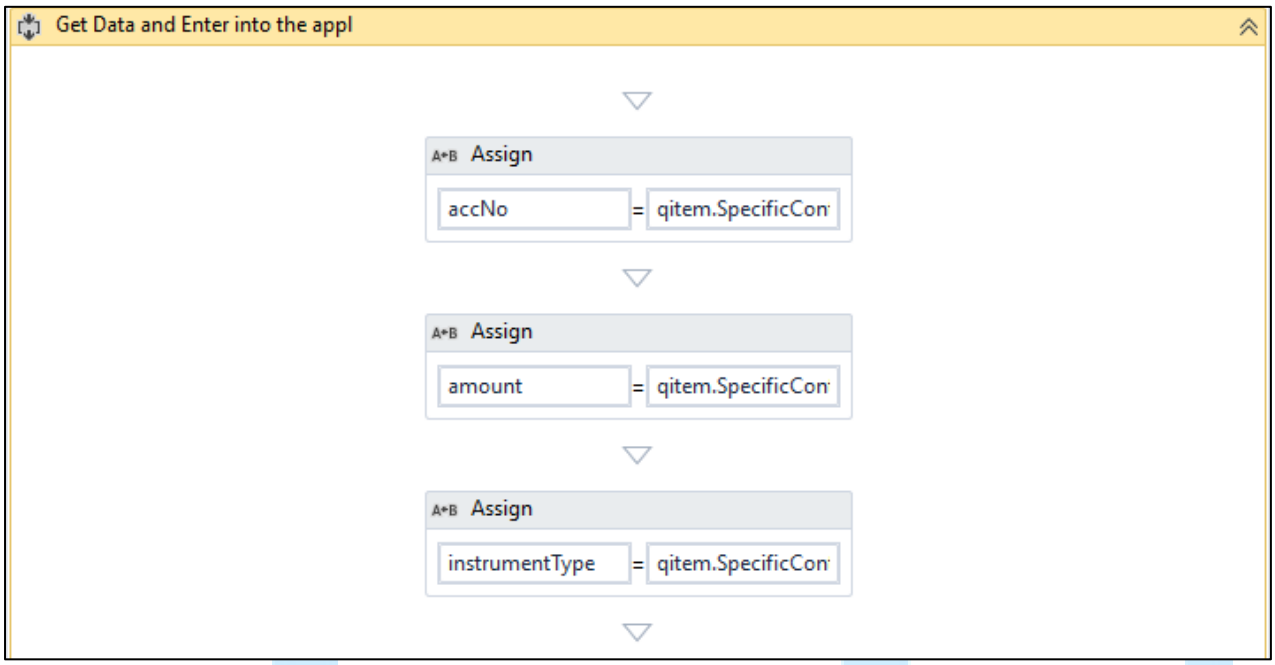


8. If the Condition is True i.e. if the application is available, then add a **TryCatch** Activity, name the Activity as **FillData. Get Transaction**



Open this Activity and in the **Try** Section add a **Sequence**, name the sequence as **Get Data and Enter into the appl**

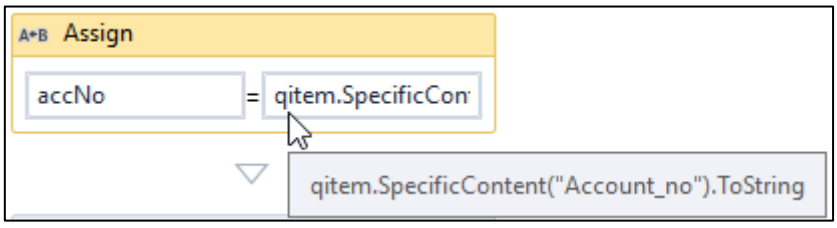
Inside the Sequence add 3 **Assign** Activities

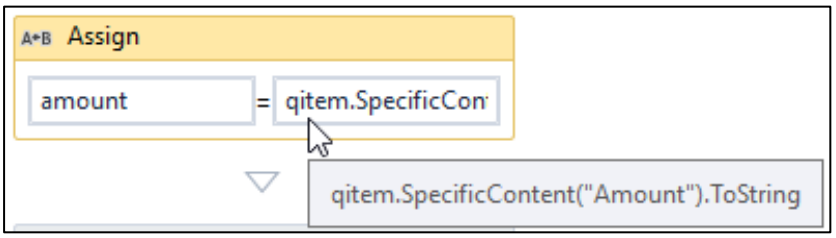


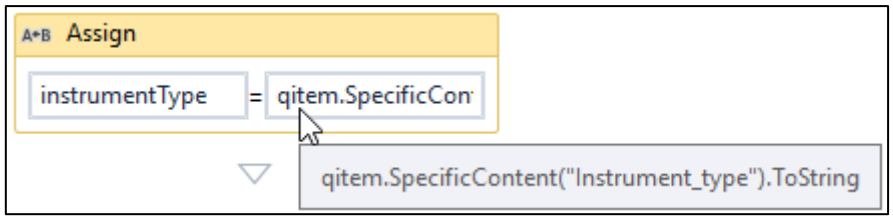
Create three variables as shown below:

Name	Variable type	Scope	Default
accNo	String	Get Data and Ente...	*Enter a VB expression*
amount	String	Get Data and Ente...	*Enter a VB expression*
instrumentType	String	Get Data and Ente...	*Enter a VB expression*

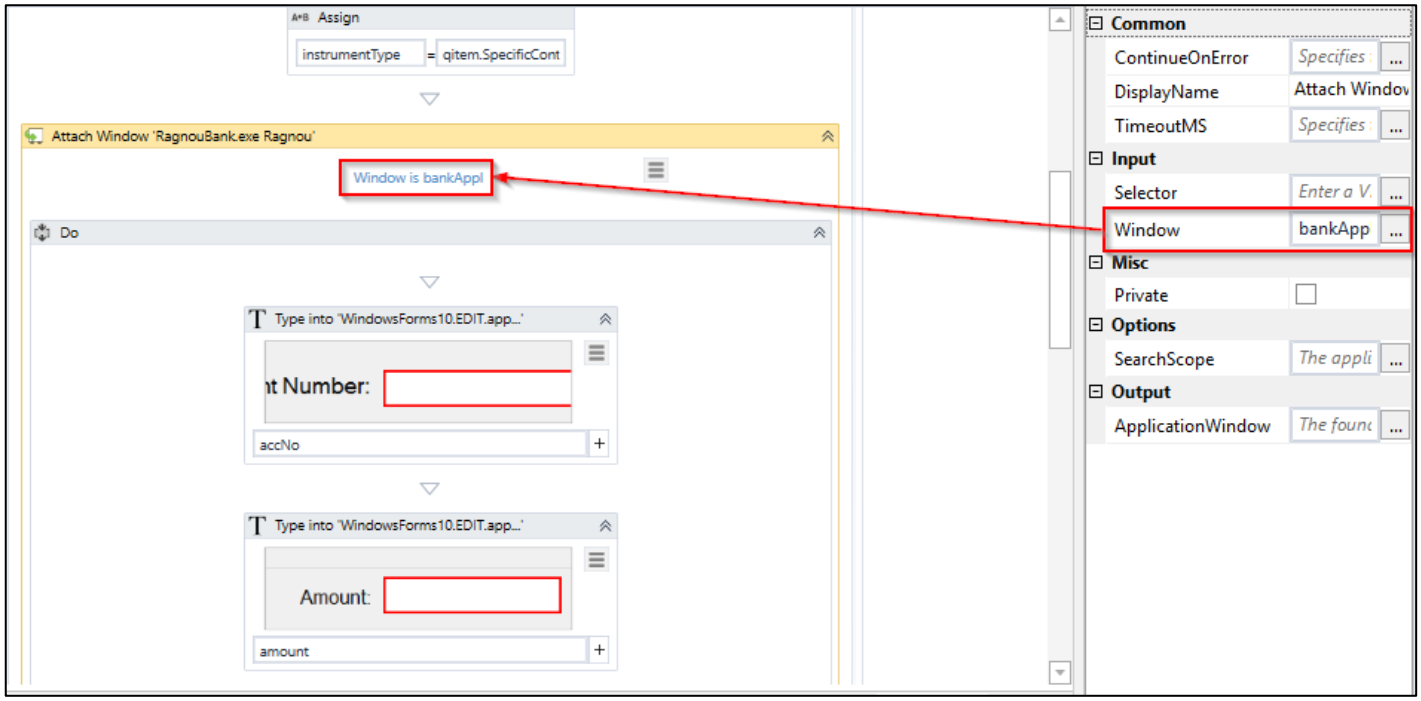
Configure these 3 **Assign** activities as follows:



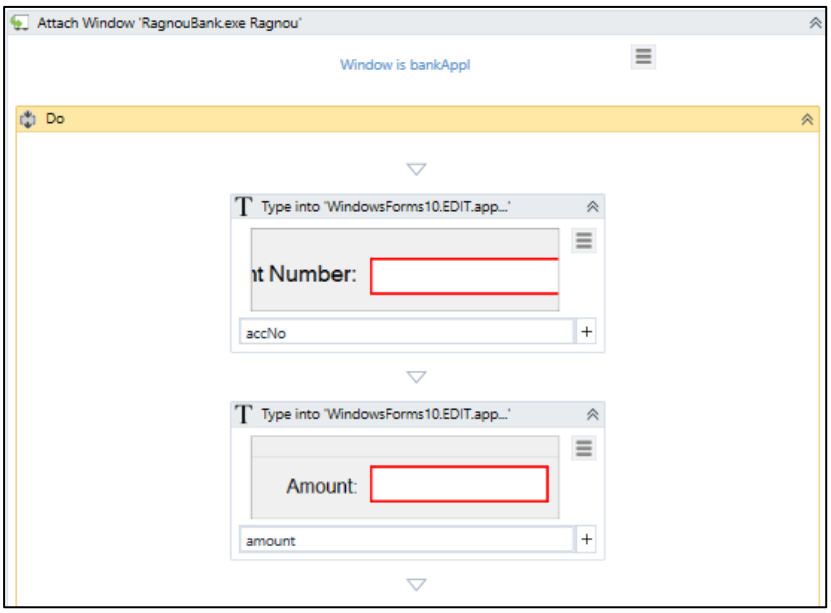




Below the **Assign** activities (in the same sequence), add **Attach Window** activity. In its properties panel, initialize **Window** as **bankAppl,** this is done to prevent the bank application from opening more than once



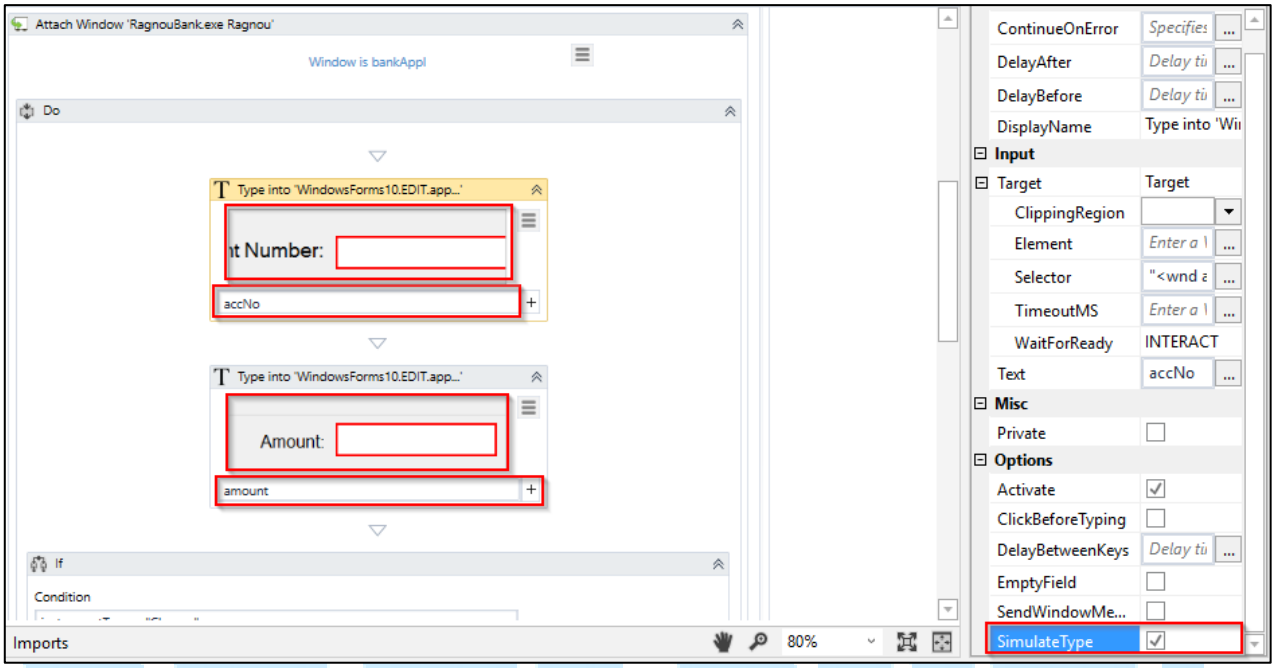
Inside the **Do** Section add 2 **Type into** activities



In the first Type into activity click the “indicate on the screen” and select the Account Number box from the bank application window. In the text box, add variable **accNo**.

Check the **SimulateType** box inside the properties panel

Similarly do for second **Type into** activity. In this activity indicate the amount box in the bank Application window, text to be mentioned as **amount**. Check the **Simulate Type** box

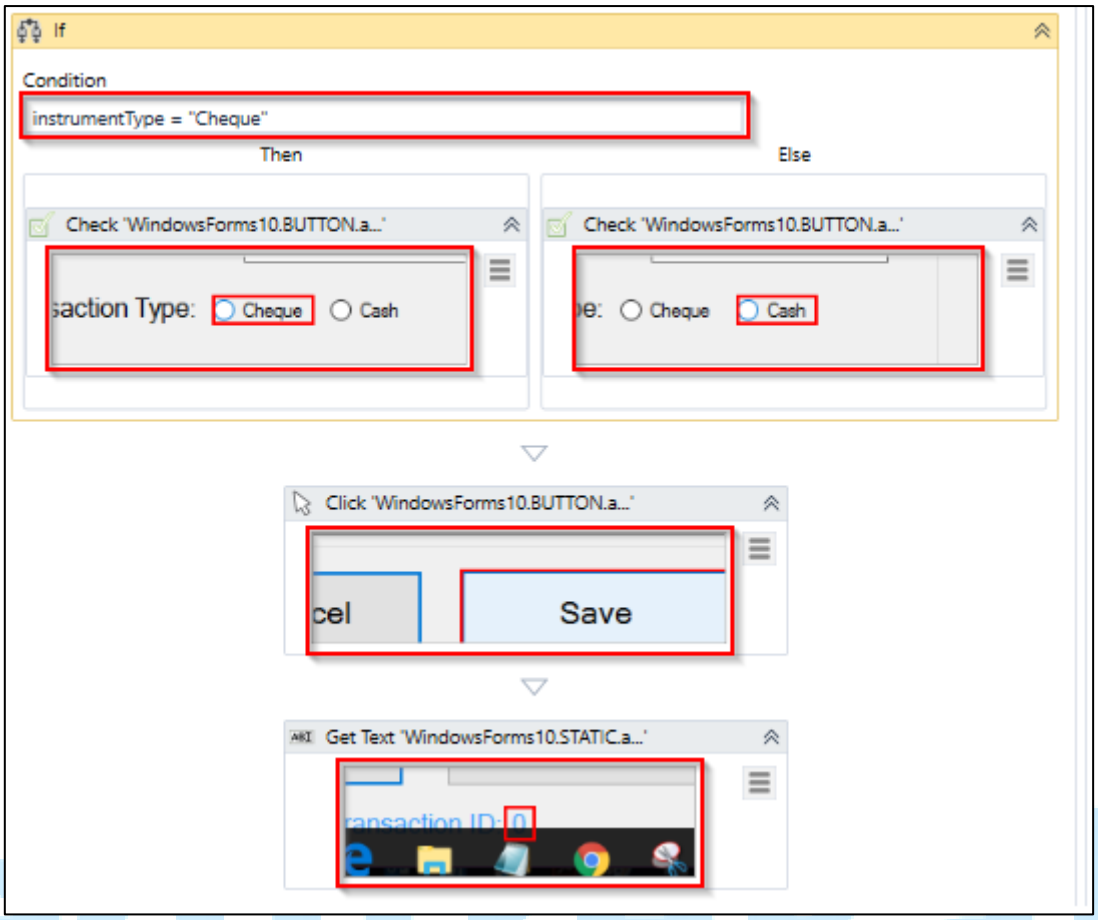


Insert an **If** condition activity just below the **Do** Section. Condition is given as:

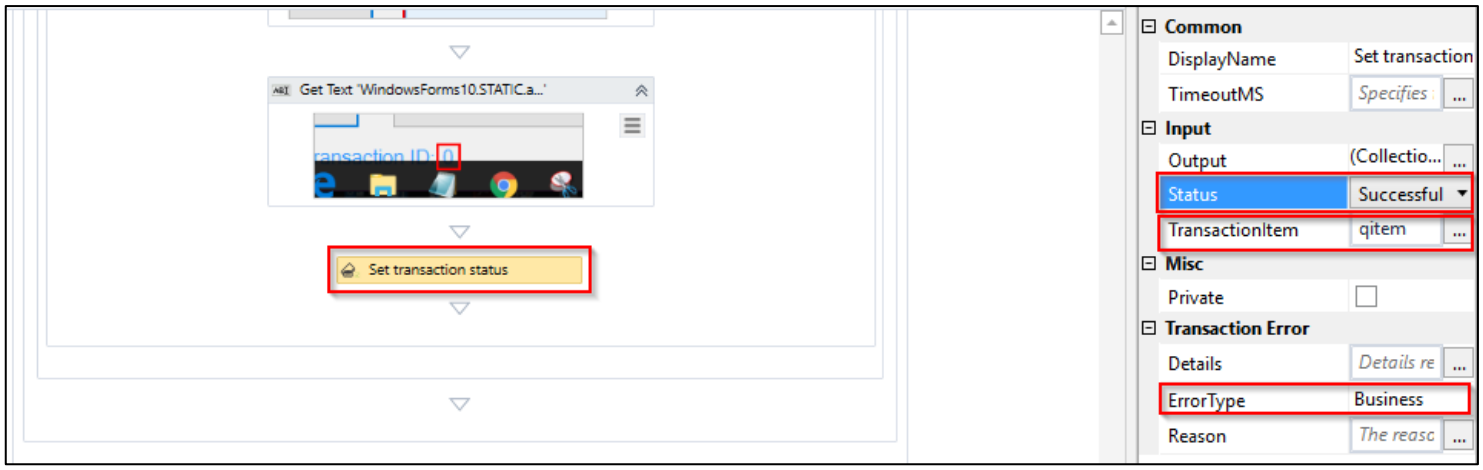
If condition is **True,** then we should check on the Cheque radio button else on Cash button. For this use **Check** activity. In the **Then** section indicate the Cheque button from the bank application window and same way indicate the Cash button in the **Else** part

After this use **Click** activity to click on the **Save** button of the bank application. Check the **SimulateType** box inside the properties panel of this activity

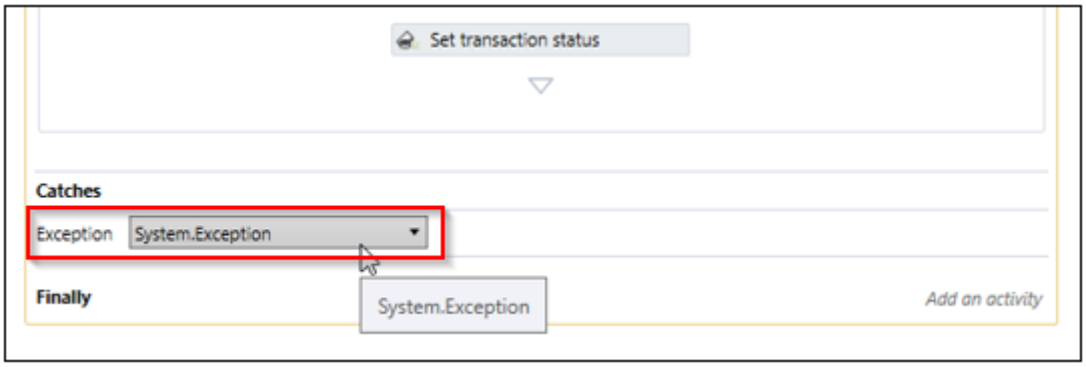
We also require the Transaction Id which can be extracted using Get Text activity. Create a variable txnId to save the Transaction Id, mention this variable inside the Output section of its properties panel



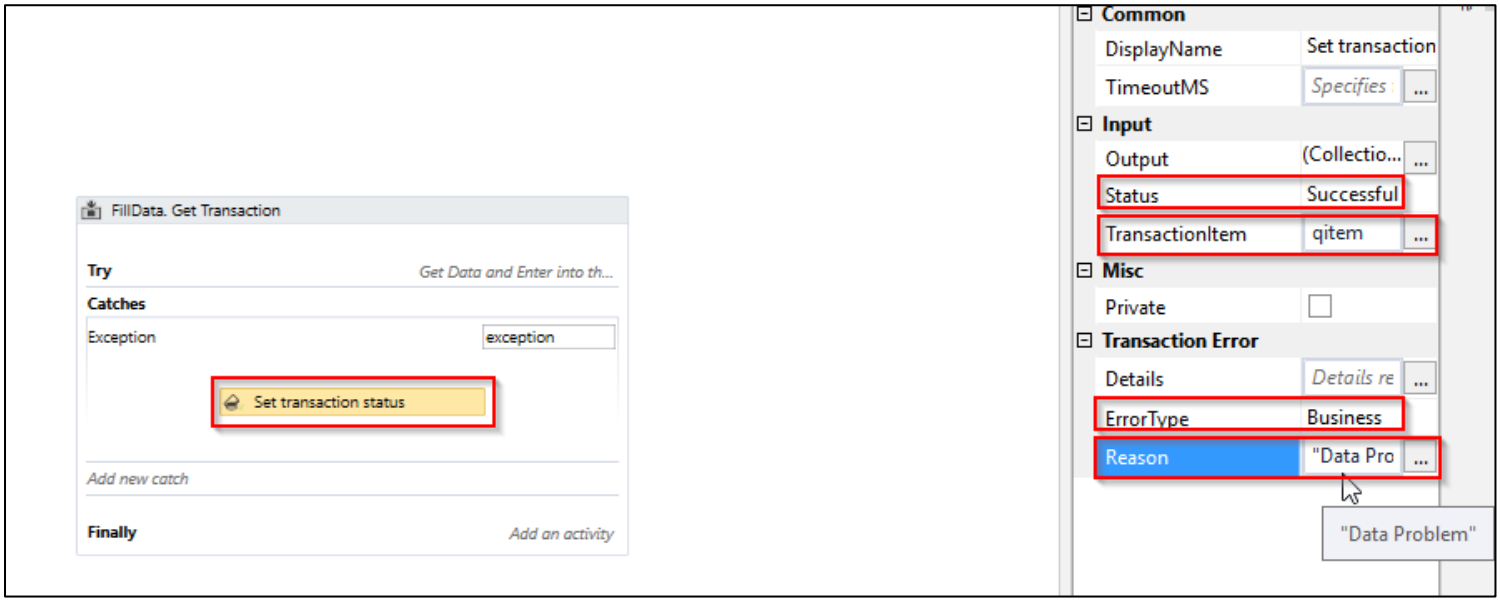
Use **Set transaction status** activity to know the status of the qitem. Configure its Properties panel as shown below:



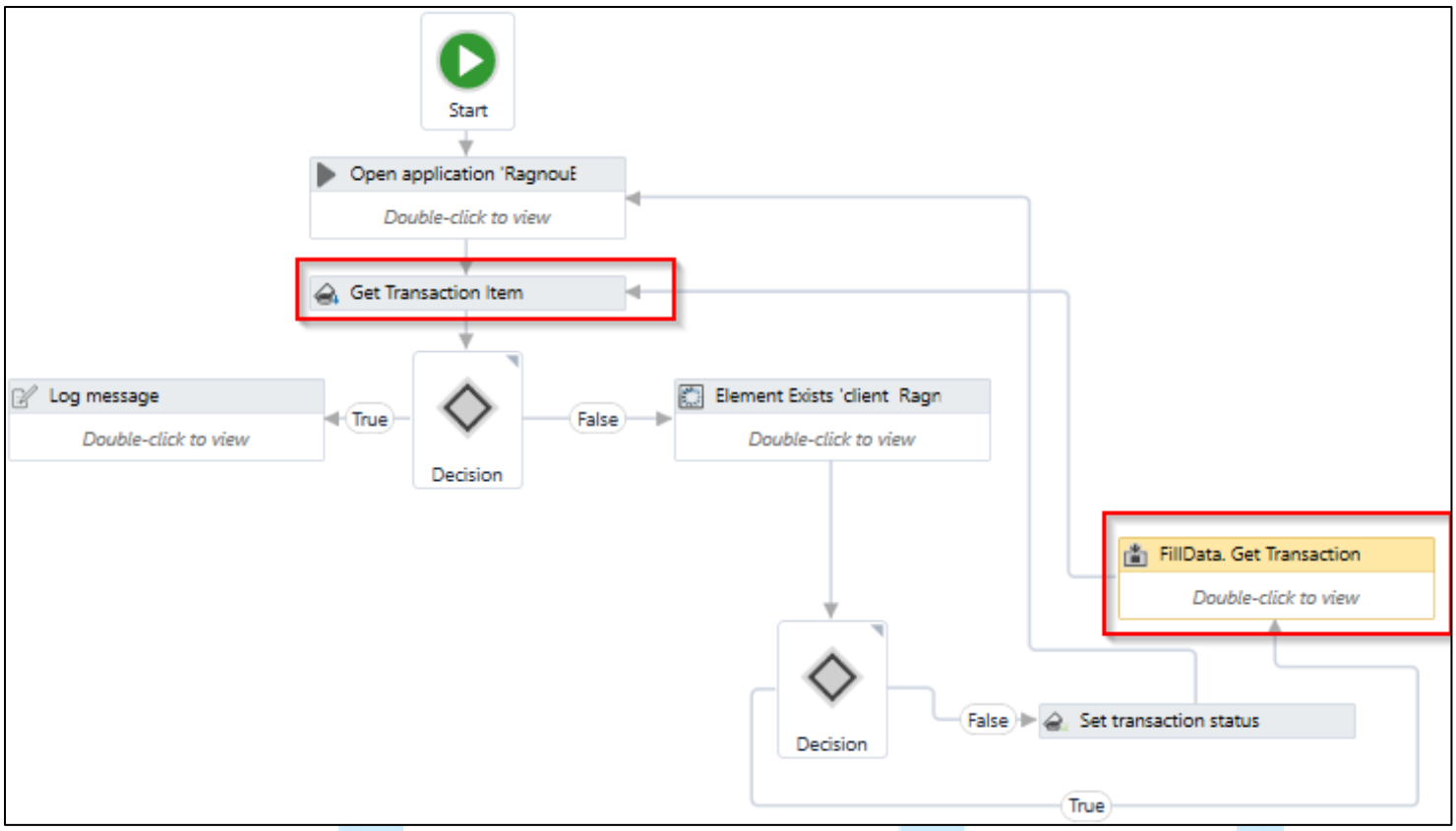
In the Catches section add new catch as **System.Exception**.



Double click the Exception to expand the activity. Add the **Set transaction status** activity to get the status again. Configure the properties panel as shown below:



Come back to the Flowchart window. Connect the **FillData. Get Transaction** box to the **Get Transaction Item** in order to create a loop for fetching data from the excel sheet


Start
Open application 'Ragnout
Double-click to view
Get Transaction Item
Log message
Double-click to view
True
Decision
False
Element Exists 'client Ragn
Double-click to view
FillData. Get Transaction
Double-click to view
Decision
False
Set transaction status
True