VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Programme	:	B.Tech	Semester	:	Fall 22-23
Course	:	Robotic Process Automation LAB	Code	:	CSE2023
Faculty	:	Sakthivel V	Slot	:	L29+L30

Date: 24-08-2022

Name: P.Je Sai Kailash

Reg Num: 20BRS1208

Assessment 5

Exercise 1 : Loop in Flow chart Activity

Aim

To build a workflow using Loop in Flowchart to ask a number from a user and display if their number is a lottery number.

Process Overview

1. Start the process.
2. Use an input dialog activity within the flowchart to get the name.
3. Use another input dialog activity to get the lottery number.
4. Now use a flow decision to check if the number is greater than 54:
   a. If it's below, then increment the counter by 1.
   b. If it's above, use another flow decision activity to check if it's below 64.
      5. In the second flow decision:
   c. If it's above 64, increment the counter by 1
   d. If its below, use a message box to display "congrats"
5. Use another flow decision if the user has run out of chances:
   a. If yes, then display "sorry, you lost"
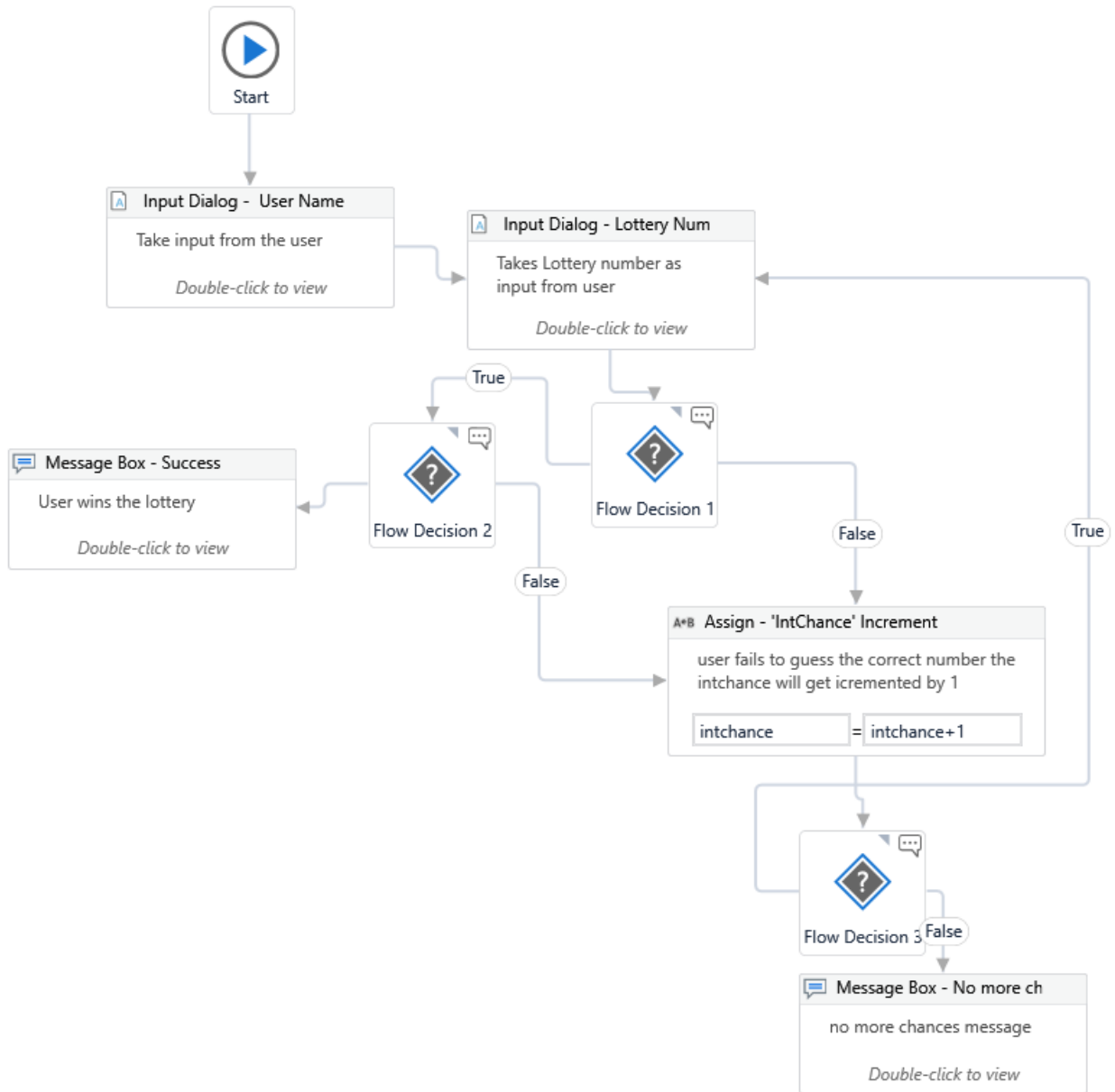6. Stop the process.

Procedure

1. Open Uipath and drag a flowchart activity and add it into the workflow space.
2. Insert an input dialog activity into the flowchart.
3. In the dialog title text box of the input dialog activity, enter "name"
4. In the input label text box enter "enter your name" and let the type of input be text box.
5. Now define a few variables that will be used in the flowchart.
6. Define a variable "name" that would store a string data type and put it in the output section of the input dialog activity for name.
7. Define a variable "intChance" that would store an Int32 data type for the counter.
8. Add another input dialog below the previous activity block.
9. In the dialog title text box enter "lottery number" and the input label text box enter " "Enter your lottery number" + (5-intChance).ToString + " Chances remains."
10. Define a variable "intEnteredNum" to store the lottery number given by the user.
11. Connect a flow decision to the lottery number input dialog activity and enter the condition "54 < intEnteredNum".
12. Connect another flow decision to the true node of the previous flow decision and enter the condition "intEnteredNum < 64".
13. At the true node of the above flow decision add a message box that should display " "Congratulations!" + name + "you won the lottery!".
14. Insert an assign activity and connect it to the false node of both the flow decision activities.
15. In the assign activity, in the to text box enter intChance and in the value text box enter "intChance + 1"
16. Now insert another flow decision and connect it to the assign activity and enter the condition "intChance < 5".
17. Connect the true node to the input dialog which takes the lottery number. Connect a message box to the false node that should display a message "Sorry, you lost. No more chances remaining"
18. End the process. Save, debug and run the file.

Activity Box

Flowchart- Loop in Flowchart

this code demonstrated a workflow that asks user for his name and two-digit lottery number and displays if he is a winner

Start

Input Dialog - User Name

Take input from the user

Double-click to view

Input Dialog - Lottery Num

Takes Lottery number as input from user

Double-click to view

True

Flow Decision 2

Flow Decision 1

Message Box - Success

User wins the lottery

Double-click to view

False

False

True

Assign - 'IntChance' Increment

user fails to guess the correct number the intchance will get icremented by 1

| intchance | = | intchance+1 |

Flow Decision 3 False

Message Box - No more ch

no more chances message

Double-click to view

Variables Used

Name	Variable type	Scope	Default
name	String	Flowchart- Loop i...	*Enter a VB expression*
intchance	Int32	Flowchart- Loop i...	*Enter a VB expression*
intenterednumber	Int32	Flowchart- Loop i...	*Enter a VB expression*

Output

Exercise 2 : Try Catch Activity

Aim

To create a workflow using Try and Catch activity to do the following:
- Take Name, Gender, and Age as the user input
- Subtract current year with Age value to get the Year of Birth
- Handle an error that occurs due to a reckless user input of an incorrect age containing the 11-digit number
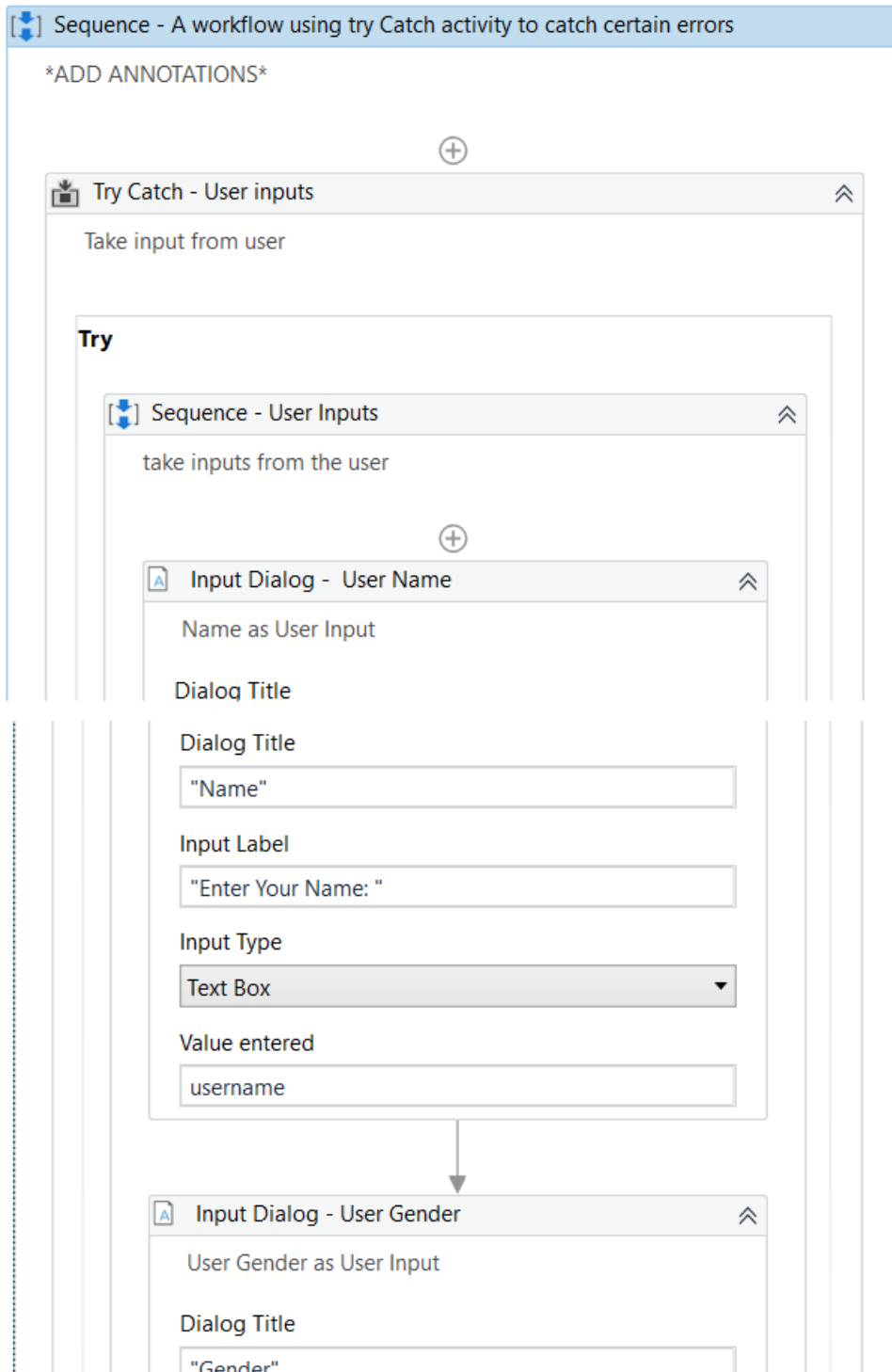- Continue the process to display the Name, Gender, and Year of Birth of the user in a message box

Process Overview

1. Start the process
2. Using the Input DIalog activity ask the user for Name,Gender and Age.
3. Use an assign activity to subtract the age from the current year to get the year of birth.
4. Use exception type in the catches section and store the error in a string variable.
5. Use a message box to display all the information along with the error, if any.
6. End the process

Procedure

1. Open Uipath.
2. Now drag a sequence activity into the workflow.
3. Now add a Try catch activity into the sequence.
4. Now add another sequence within the try section of the try catch activity.
5. Add an input dialog activity within the sequence where the title is "Name" , input label is "enter your name" and the input type is a text box.
6. Add another input dialog activity below the previous input dialog activity and add the title as "gender" and the input label as "enter your gender" which should store the input type as text box.
7. Add another input dialog activity below the gender input dialog activity and add the title as "age" with the input label as "enter your age" and the input type should be a text box.
8. Now define variables to store the input given by the user.
9. "username" to store the name of the user in the input dialog for name activity.
10. "userGender" to store the gender of the user in the input dialog for gender activity.
11. "intUserAge" to store the age of the user.
12. "intYearofBirth" to store the year of birth. "inputError" to store an error message, if any.
13. Now insert an assign activity below the input dialog for age activity.
14. In the to text box enter "intYearofBirth" and in the value text box enter "Date.Today.Year() – intUserAge"
15. In the catches section of the try catch activity, select "System.Exception"
16. Add an assign activity within the catches section where the value stored is "exception.Message" into the variable "inputError"
17. Insert a message box below the try catch activity to finally display all the information along with the error, if any.
18. End the process.

Activity Box

Sequence - A workflow using try Catch activity to catch certain errors

ADD ANNOTATIONS

⊕

Try Catch - User inputs ⌃

Take input from user

Try

Sequence - User Inputs ⌃

take inputs from the user

⊕

Input Dialog - User Name ⌃

Name as User Input

Dialog Title

Dialog Title

"Name"

Input Label

"Enter Your Name: "

Input Type

Text Box ▼

Value entered

username

Input Dialog - User Gender ⌃

User Gender as User Input

Dialog Title

"Gender"

Input Label

"Enter your Gender: "

Input Type

Text Box ▼

Value entered

usergender

⊕

↓

🅰 Input Dialog - User Age ⌃

User Age as User Input

Dialog Title

"Age"

Input Label

"Enter your Age"

Input Type

Text Box ▼

Value entered

intuserage

A+B Assign- Year of birth

present year-User's year of birth
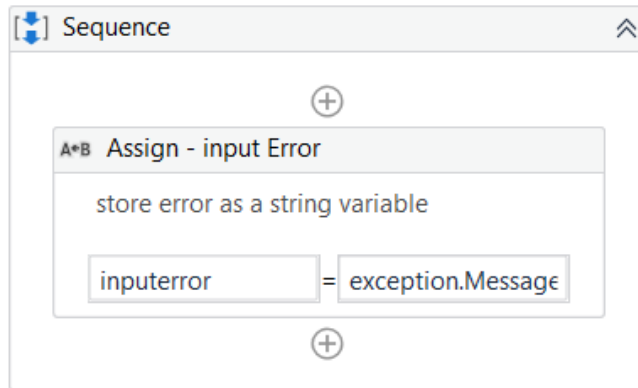
intyearofbirth = Date.Today.Year()-i

⊕

Catches

Exception   *Sequence*

Add new catch

Catches
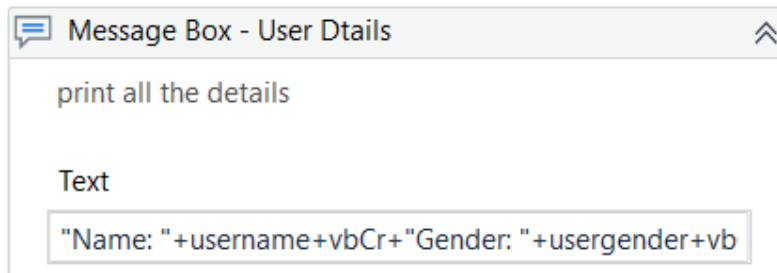
Exception   exception

[↕] Sequence   ⌃

                          ⊕

A•B Assign - input Error

store error as a string variable

inputerror   =   exception.Message

                          ⊕

Add new catch

Finally   *Message Box - User Dtails*

Finally

💬 Message Box - User Dtails   ⌃

print all the details

Text

"Name: "+username+vbCr+"Gender: "+usergender+vb

Output

Name   ✕

Enter Your Name:

```
Kailash
```

[Ok]

Gender   ✕

Enter your Gender:

```
Male
```

[Ok]

Age   ✕

Enter your Age

```
19
```

[Ok]

Message Box - User ...   ✕

Name: Kailash
Gender: Male
Year of Birth: 2003
Error:

[OK]

Exercise 3 : Data Conversions

Aim

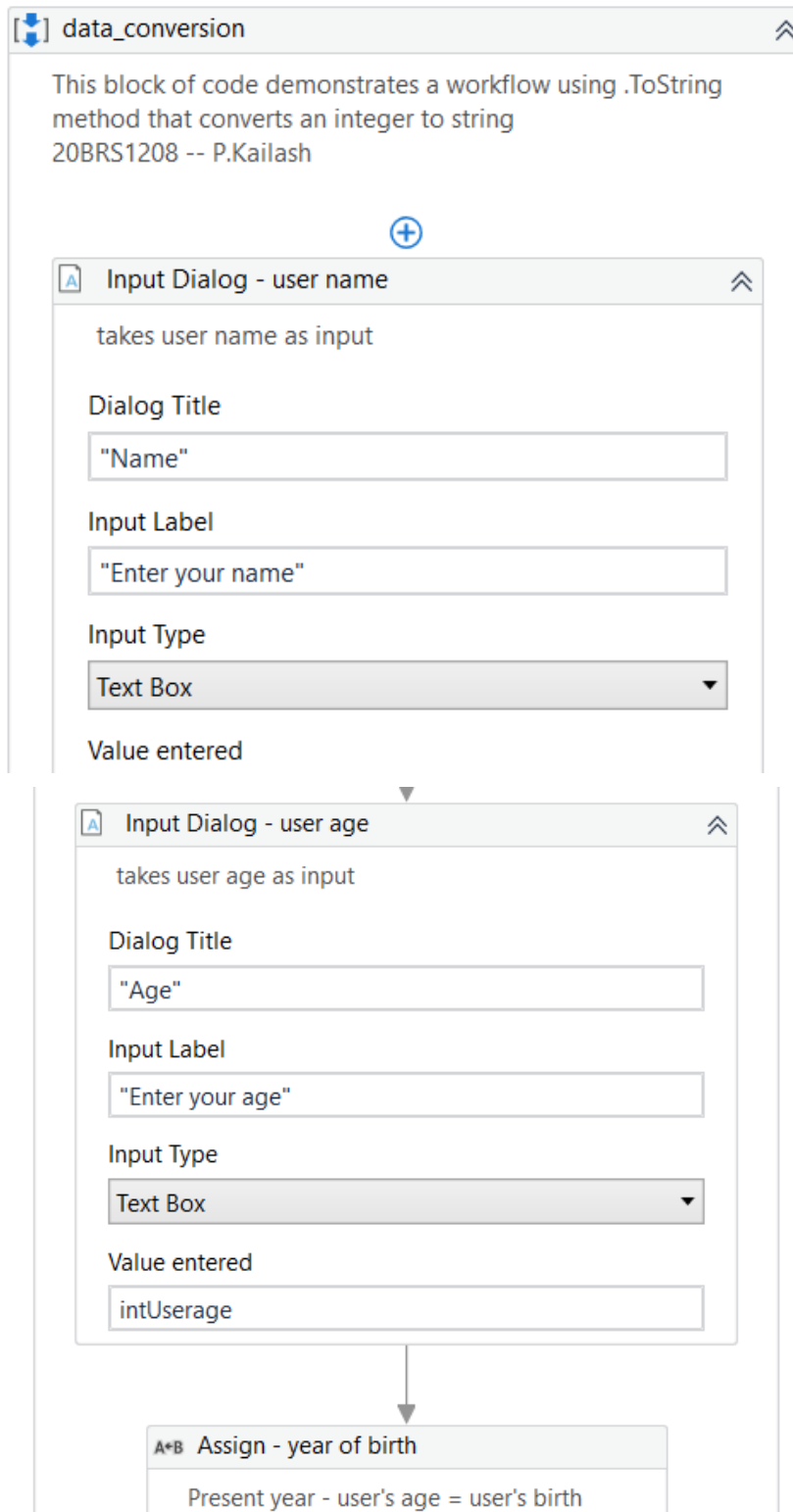To build a workflow using the. To String method that converts an integer to a string.
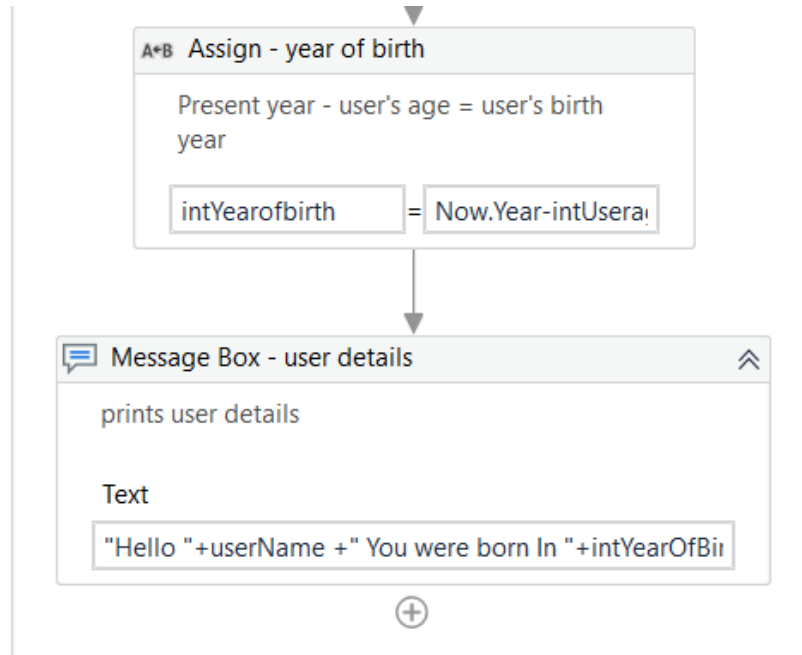
Process Overview

1. Start the process.
2. Use an input dialog activity to get the name and age of the user.
3. Now use an assign activity to find the birth year of the user.
4. Use a message box to display the year of the birth to the user.
5. End the process.

Procedure

1. Open Uipath.
2. Now add a sequence activity into the workflow
3. Now add an input dialog activity within the sequence where the title is "Name" and the input label as "enter your name" and the input type should be a text box.
4. Now add another input dialog activity below the previous input dialog activity and enter the title as "age" and the input label as "enter your age" and let the input type be a text box.
5. Now let us create variables to store the data given by the user.
6. Create "userName" and enter it to the output property of the name input dialog activity.
7. Create "intUserAge" and enter it to the output property of the age input dialog activity.
8. Create another variable "intYearofBirth" to store the birth year.
9. Now add an assign activity below the input dialog activity and in the to text box enter "intYearOfBirth" and in the value section enter "Now.Year – intUserAge"
10. Now add a message box to display all the information.
11. End the process.

Activity Box

[⇕] data_conversion ≪

This block of code demonstrates a workflow using .ToString method that converts an integer to string
20BRS1208 -- P.Kailash

⊕

[A] Input Dialog - user name ≪

takes user name as input

Dialog Title

"Name"

Input Label

"Enter your name"

Input Type

Text Box ▼

Value entered

▽

[A] Input Dialog - user age ≪

takes user age as input

Dialog Title

"Age"

Input Label

"Enter your age"

Input Type

Text Box ▼

Value entered

intUserage

▽

A+B Assign - year of birth

Present year - user's age = user's birth

A+B Assign - year of birth

Present year - user's age = user's birth year

| intYearofbirth | = | Now.Year-intUsera(|

Message Box - user details ⌃

prints user details

Text

"Hello "+userName +" You were born In "+intYearOfBi|

⊕

▪▮ Message Box - user details   ✕

Hello Kailash You were born In 2003

OK

Exercise 4 : String manipulations - 1

Aim

To create a workflow using Format, Join, IndexOf, Split, and Substring methods that extract key information from a text and prints in a different format
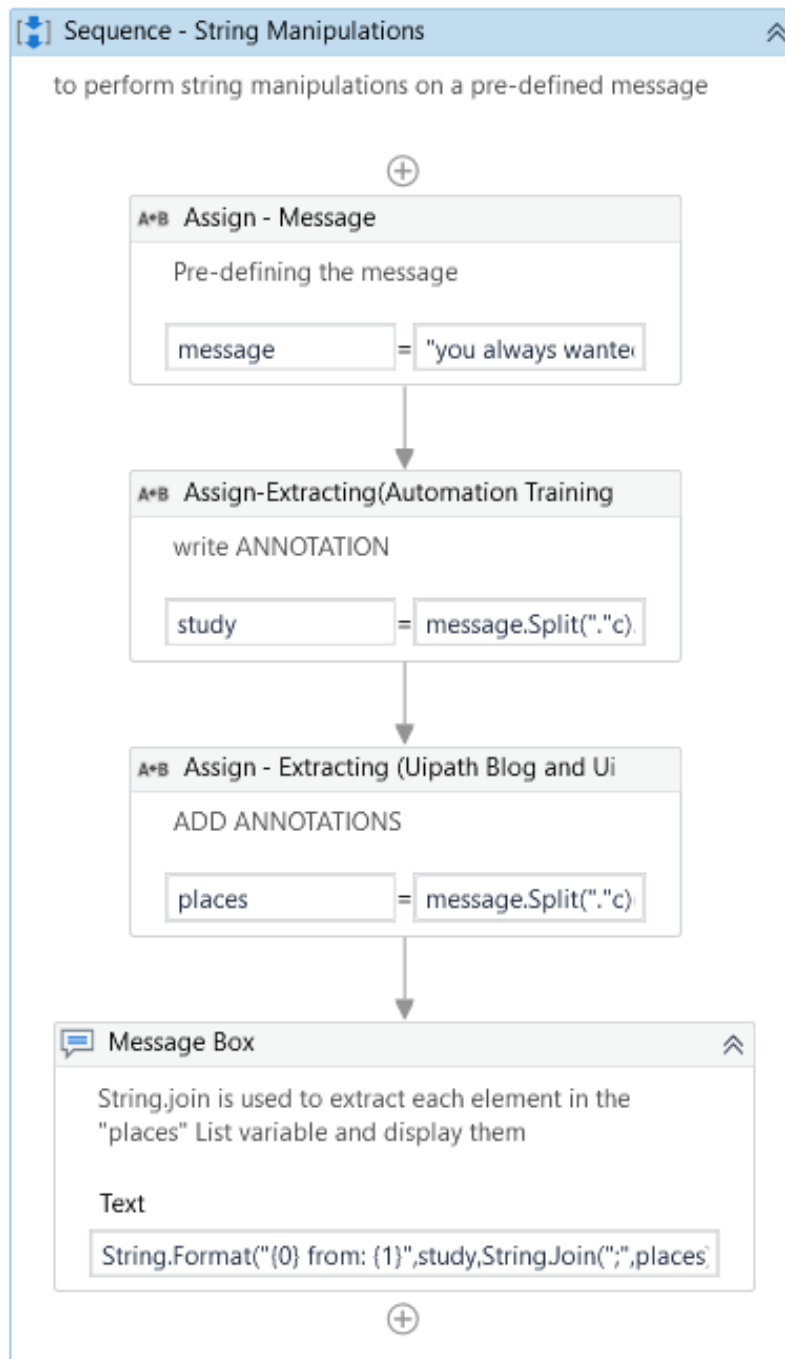
Process Overview

1. Start a process
2. Use assign activity to give the initial string message "You always wanted to study Automation Training. The materials are available in the following places: UiPath Blog, UiPath Academy."
3. Create a new String variable study and use a succession of String methods to assign the course from the query:
   message.Split(".".c).First.ToString.Substring(message.LastIndexOf("study"))]
   a. Split(".".c).First.ToString extracts the first sentence of the String and converts it to a String
   b. Substring(message.LastIndexOf("study")) extracts the Substring starting from "get"
4. Create a new List variable places and use a succession of String methods to assign the places from the query:
   message.Split(".".c)(1).ToString.Split(":".c).Last.ToString.Split(",".c).ToList
   a. message.Split(".".c)(1).ToString extracts the second sentence of the String and converts it to a String.
   b. Split(":".c).Last.ToString splits the remaining string and keeps only the last part of it.
   c. Split(",".c).ToList takes each string separated by comma and adds it as an element in the List variable.
5. Use a Message Box activity to display output using this expression:
   String.Format("{0} from: {1}", study,String.Join(";", places))
   a. String.Join is used to extract each element in the "places" List variable and display them.
6. End the process

Procedure

1. Open UiPath.
2. Add a sequence activity into the workflow.
3. In the variables panel, create three variables.
4. Create "message" with data type as string.
5. Create "study" with a data type as string
6. And create "places" with a data type as List<System.String>
7. Now add an assign activity within the sequence.
8. In the assign activity, add "message" variable in the to text box and the text " You always wanted to study automation training. The materials are available in the following places: UiPath Blog, UiPath Academy." In the value text box.
9. Now insert another assign activity below the previous assign activity.
10. Add "study" variable in the to text box and " message.split(".")c).First.ToString.Substring(message.LastIndexOf(("study")) " in the value text box.
11. Now add another assign activity below the previous assign activity.
12. In the assign activity, ass "places" variable in the to text box and " message.Split(".")c)(1).ToString.Split(":"c).Last.ToString.Split(","c).ToList " in the value text.
13. Now add a message box activity and in the activity text box enter " String.Format("{0} from: {1}", study ,String.Join(";", places)) ".
14. End the process
15. Save, debug and run the file.

Activity Box

Sequence - String Manipulations

to perform string manipulations on a pre-defined message

⊕

A•B Assign - Message

Pre-defining the message

| message | = | "you always wante |

↓

A•B Assign-Extracting(Automation Training

write ANNOTATION

| study | = | message.Split("."c) |

↓

A•B Assign - Extracting (Uipath Blog and Ui

ADD ANNOTATIONS

| places | = | message.Split("."c) |

↓

💬 Message Box

String.join is used to extract each element in the
"places" List variable and display them

Text

| String.Format("{0} from: {1}",study,StringJoin(";",places |

⊕

Variables Used

Name	Variable type	Scope	Default
message	String	Sequence - String...	*Enter a VB expression*
study	String	Sequence - String...	*Enter a VB expression*
places	List<String>	Sequence - String...	*Enter a VB expression*

Output

Message Box ✕

studyautomation training from: the materialss are in UiPath Blog; UiPath Academy

OK

Exercise 5 : String manipulations - 2

Aim

To create a workflow using Split and Contains methods that extract sentences containing "RPA" from a paragraph.

Process Overview

1. Start the process.
2. Use an Assign activity to store a paragraph in a string variable called new Text.
3. Use newText.Split(".".c) and store all sentences in an array called newSentence
4. Use a For Each activity and iterate through each item in the array newSentence
5. Use an If activity within the For Each activity to identify sentence that contains the string "UiPath in it. Use item.Contains("RPA") as condition
6. Use a Type Into activity to store result in an MS Word file
7. Stop the process.

Procedure

1. Open a new MS word file.
2. Now open UiPath.
3. Now add a sequence activity within the workflow.
4. Now create two variables in the variables panel.
5. Create "newText" variable containing variable type string.
6. Create "newsentence" variable containing variable type String[].
7. Now, add an assign activity within the sequence.
8. In the assign activity, enter the variable "newText" within the to text box and the following text in the value text box:
" Robotic process automation (RPA) is a software technology that makes it easy to build, deploy, and manage software robots that emulate human actions interacting with digital systems and software. RPA streamlines
workflows, which makes organizations more profitable, flexible, and responsive.
RPA is noninvasive and can be rapidly implemented to accelerate digital

transformation. It also increases employee satisfaction, engagement, and productivity by removing mundane tasks from their workdays"

9. Now add another assign activity below the previous assign activity.

10. Enter newSentence variable in the to text box and the "newText.Split(","c) in the value text box.

11. Now add a for each activity. And in the first box enter item and in the second box enter newSentence and change the type argument as string.

12. In the body section of the for each activity, add an if activity and enter the condition as item.contains("RPA")

13. Add an attach window and place it in the then section of the if activity.

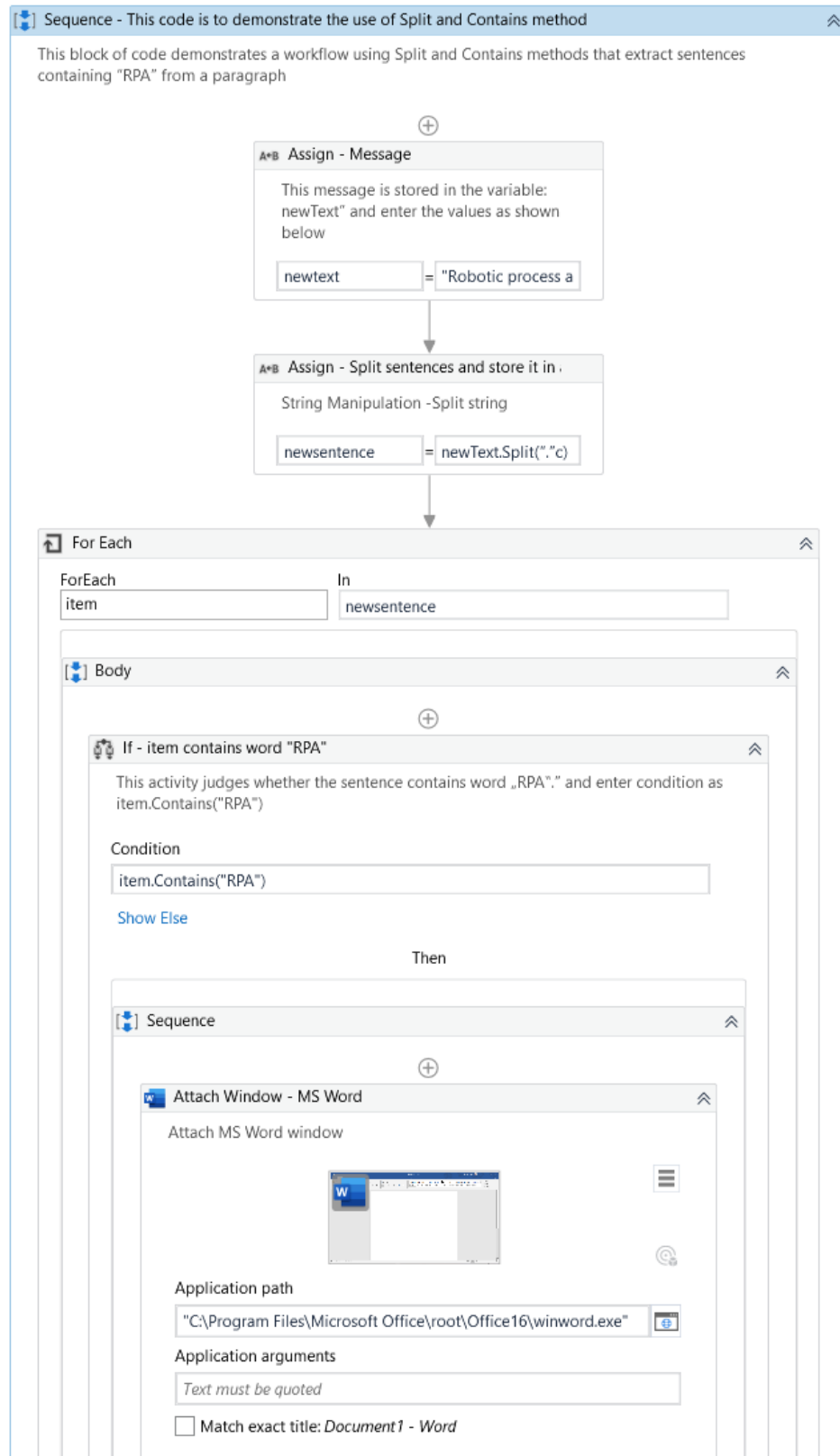14. Click on indicate element on screen and indicate it to the MS word window.

15. Now drag and drop a type into activity in the do container of the attach window activity and indicate it to the editor area of the MS word.

16. In the text area type into activity enter " item.ToString+"[k(enter)]"

17. End the process.

18. Save, debug and run the file.

Activity Box

Sequence - This code is to demonstrate the use of Split and Contains method

This block of code demonstrates a workflow using Split and Contains methods that extract sentences containing "RPA" from a paragraph

⊕

A•B Assign - Message

This message is stored in the variable: newText" and enter the values as shown below

| newtext | = | "Robotic process a |

A•B Assign - Split sentences and store it in

String Manipulation - Split string

| newsentence | = | newText.Split("."c) |

For Each

ForEach	In
item	newsentence

Body

⊕

If - item contains word "RPA"

This activity judges whether the sentence contains word „RPA"." and enter condition as item.Contains("RPA")

Condition

| item.Contains("RPA") |

Show Else

Then

Sequence

⊕

Attach Window - MS Word

Attach MS Word window

Application path

| "C:\Program Files\Microsoft Office\root\Office16\winword.exe" | ⊕ |

Application arguments

| *Text must be quoted* |

☐ Match exact title: *Document1 - Word*

[↕] Do ⌃

⊕

TI₀ Type Into - MS Word ⌃

Types each item in the word file

Type this   ◉ Standard ○ Secure

`item.ToString+ "[k(enter)]"` 👁̸ ▾

Empty field before typing   **Click before typing**

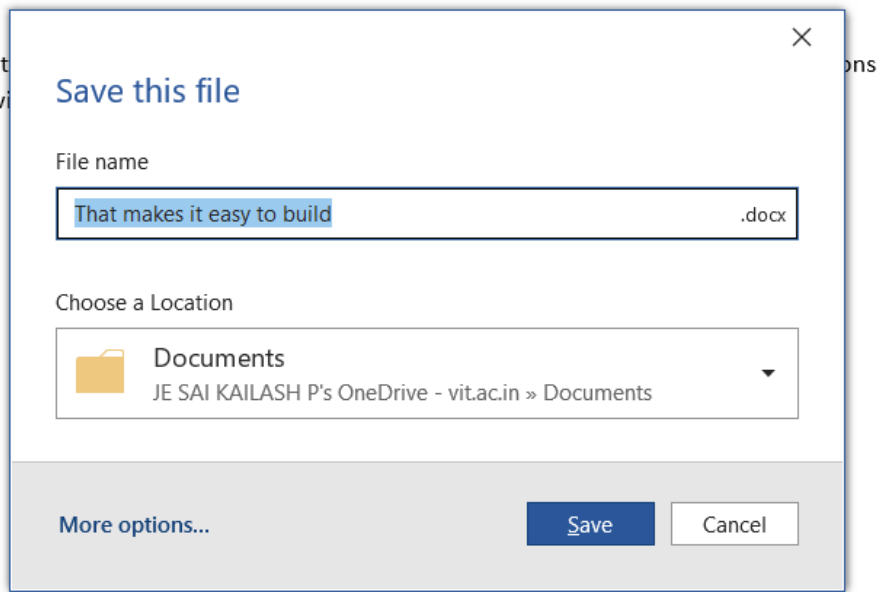Single line [End, Shift+H ▾   Single   ▾

⊕

Output

That makes it easy to build, deploy, and manage software robots that emulate human actions interacting with digital systems and software

That makes it [Save this file dialog] ons
interacting wi

Save this file ✕

File name

That makes it easy to build   .docx

Choose a Location

📁 Documents
JE SAI KAILASH P's OneDrive - vit.ac.in » Documents ▾

More options...   **Save**   Cancel

Result

Hence, we have successfully created workflows that Loop in Flow chart Activity, Try Catch Activity, Data Conversions, String manipulations.