TABLE OF CONTENTS

1. INTRODUCTION TO RPA:

Robotic Process Automation (RPA) is an innovative technology that enables organizations to automate repetitive and rule-based tasks using software robots or "bots." RPA bots mimic human interactions with digital systems and applications, performing tasks that were traditionally carried out by humans. This technology allows businesses to streamline their operations, increase efficiency, and reduce costs.

RPA operates by interacting with the user interface of existing software applications, just like a human user would. It can navigate screens, input data, retrieve information, perform calculations, and execute various actions based on predefined rules and logic. RPA bots can handle structured and repetitive processes, allowing employees to focus on higher-value tasks that require critical thinking, problem-solving, and creativity.

Key aspects of RPA include:

Scalability: RPA allows organizations to automate a wide range of processes, from simple and repetitive tasks to more complex and data-intensive workflows. Bots can be easily replicated and scaled up to handle increased workloads, offering flexibility and adaptability to changing business needs.

Integration: RPA can integrate with existing software systems and applications without requiring major changes to the underlying infrastructure. This makes it a non-intrusive technology that can work alongside legacy systems, databases, spreadsheets, and even virtual environments.

Rules-Based Automation: RPA bots follow predefined rules and logic to execute tasks. These rules can be configured by users or developers, allowing bots to make decisions, validate data, and perform specific actions based on defined criteria.

Accuracy and Consistency: RPA eliminates human errors and ensures high accuracy and consistency in performing tasks. Bots follow the same set of rules consistently, reducing the likelihood of mistakes and improving data quality.

Cost and Time Savings: By automating repetitive tasks, RPA reduces manual effort, increases productivity, and saves time. This technology allows organizations to achieve significant cost savings by optimizing resource utilization and achieving faster process execution.

Auditability and Compliance: RPA provides detailed logs and audit trails, allowing organizations to track and monitor each step of the automated process. This enhances transparency, improves compliance with regulations and policies, and facilitates internal and external audits.

RPA can be applied across various industries and departments, including finance, human resources, customer service, supply chain management, and more. It empowers organizations to achieve operational excellence, enhance customer experiences, and focus on strategic initiatives while the bots handle repetitive and mundane tasks efficiently.

2. INTRODUCTION TO UIPATH SOFTWARE:

UiPath is a leading Robotic Process Automation (RPA) software platform that provides a comprehensive set of tools for designing, deploying, and managing automation workflows. It offers an intuitive user interface called UiPath Studio, which is designed to enable both technical and non-technical users to automate their business processes effectively.

Here's a breakdown of the UiPath Studio layout and its various panels:

Ribbon:
The Ribbon is located at the top of the UiPath Studio window and contains tabs with different functionalities. It provides access to various features and tools, such as file operations, activities, debugging, publishing, and managing projects.

Activities Pane:
The Activities Pane is located on the left-hand side of the UiPath Studio window. It houses a wide range of pre-built activities that can be dragged and dropped onto the workflow designer canvas. These activities represent actions that can be performed by the RPA bot, such as data manipulation, file operations, web interactions, and more.

Workflow Designer Canvas:
The Workflow Designer Canvas is the central area of UiPath Studio. It serves as the workspace where users can design their automation workflows. The canvas supports a visual, drag-and-drop approach, allowing users to arrange activities, define their sequence, and establish connections between them to create a workflow.

Properties Pane:
The Properties Pane is located on the right-hand side of the UiPath Studio window. It displays the properties of the selected activity or element on the canvas. Users can configure various settings and parameters for each activity, such as input/output variables, selectors, timeouts, and more.

Output Pane:
The Output Pane is located at the bottom of the UiPath Studio window. It provides real-time feedback and displays the execution output, including error messages, warnings, and information logs. This pane helps users identify and troubleshoot any issues during workflow development and execution.

Variables Pane:
The Variables Pane is a dedicated panel for managing variables used within the automation workflow. It allows users to define and manage variables of different data types, including strings, numbers, arrays, and more. Variables can store values and be used to pass data between activities within the workflow.

Project Panel:
The Project Panel is located on the left-hand side of the UiPath Studio window. It provides a hierarchical view of the project structure, including files, folders, and workflows. Users can organize their automation projects, add new files, manage dependencies, and navigate between different components of the project.

Activities Library Pane:
The Activities Library Pane is a searchable panel that provides quick access to all available activities. It allows users to search for specific activities, browse categories, and access their properties directly from the library. This pane offers a convenient way to discover and utilize the extensive range of pre-built activities in UiPath.

UiPath Studio offers a visually appealing and user-friendly interface that empowers users to create automation workflows efficiently. The layout and panels mentioned above facilitate the smooth development, customization, and management of RPA processes using UiPath.

3. TERMWORKS:

TERMWORK 1

DATE: 22/03/2023

PROBLEM STATEMENT:

Build a workflow that prints "Hello World" in a message box.

THEORY:

Printing "Hello World" using a Message Box is a simple introductory exercise that helped us to understand the basic workflow design and activity usage within the UiPath Studio. The exercise involves displaying a pop-up message with the text "Hello World" using the Message Box activity.

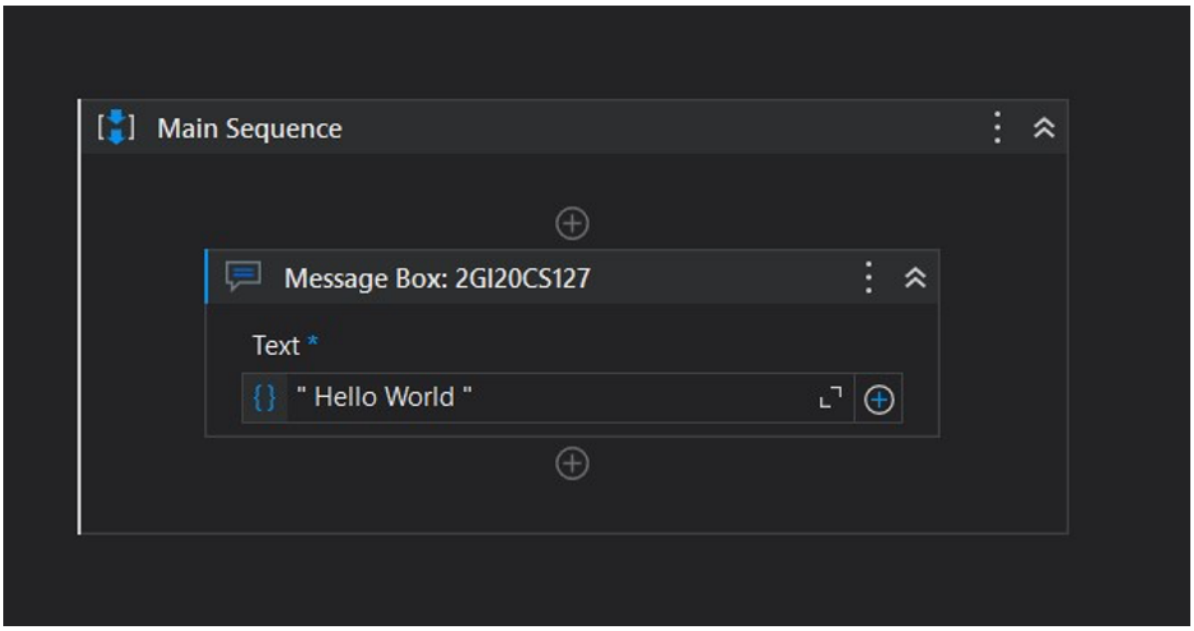
Activities Used:

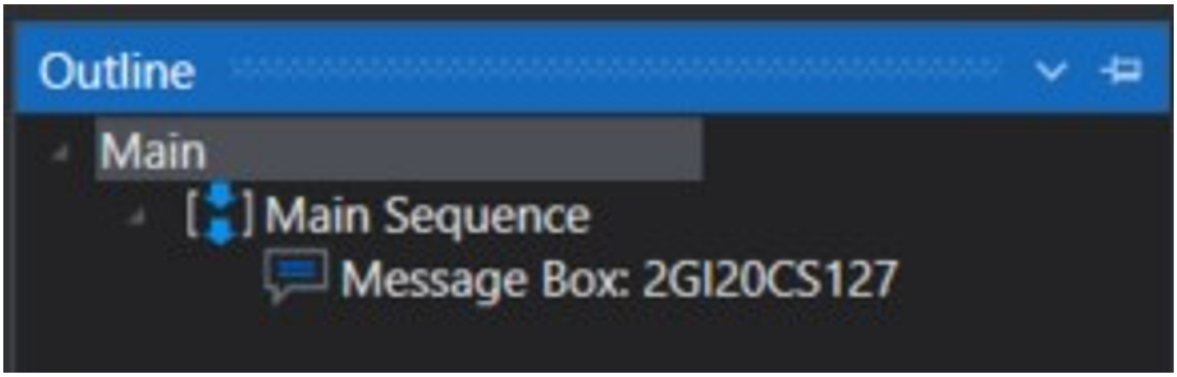
Message Box Activity: The Message Box activity is a UiPath core activity that displays a pop-up message to the user. It is commonly used for displaying information, notifications, or debugging purposes during the automation workflow development. In this case, the Message Box activity will be utilized to display the "Hello World" message.

Packages:

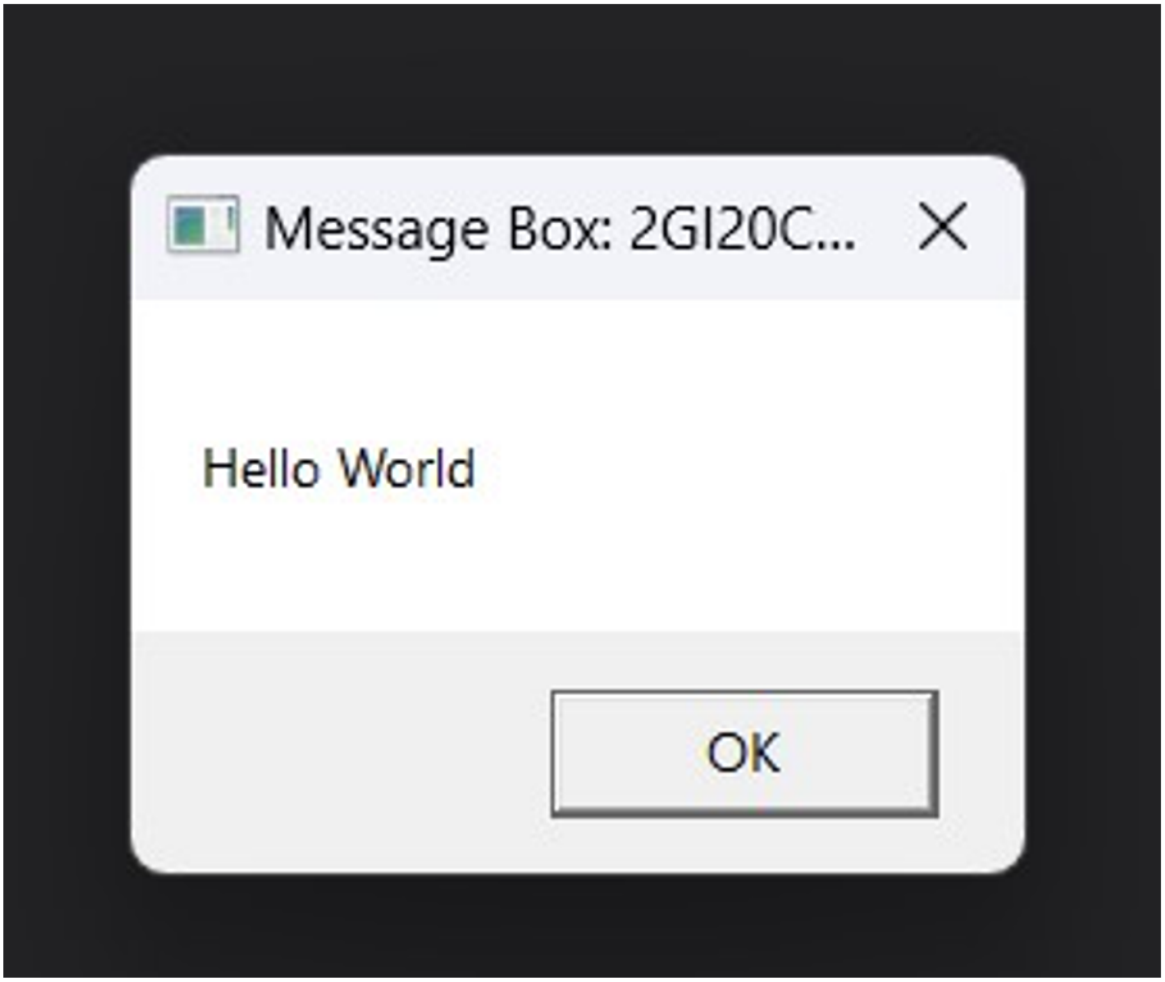
In this specific exercise, you will not require any additional packages as the Message Box activity is a built-in activity within UiPath Studio. However, it's essential to understand that UiPath supports a wide range of packages that extend the functionality and capabilities of the platform. These packages provide additional activities and features for automating various tasks and interacting with different software systems and applications.

WORKFLOW AND OUTLINE:





OUTPUT:



CONCLUSION:

In conclusion, the termwork of printing "Hello World" in UiPath using a Message Box serves as a basic introduction to workflow design and activity usage within UiPath Studio. By completing this exercise, We could gain a practical understanding of how to create a simple automation workflow and utilize the Message Box activity to display information to users.

The outcome of this termwork is the successful execution of an automation process that displays the "Hello World" message in a pop-up message box. This exercise demonstrates the simplicity and effectiveness of UiPath in automating repetitive tasks and interacting with users through intuitive interfaces.

TERMWORK 2

DATE: 29/03/2023

PROBLEM STATEMENT:

Build a workflow that swaps the values of two variables using a third variable.

THEORY:

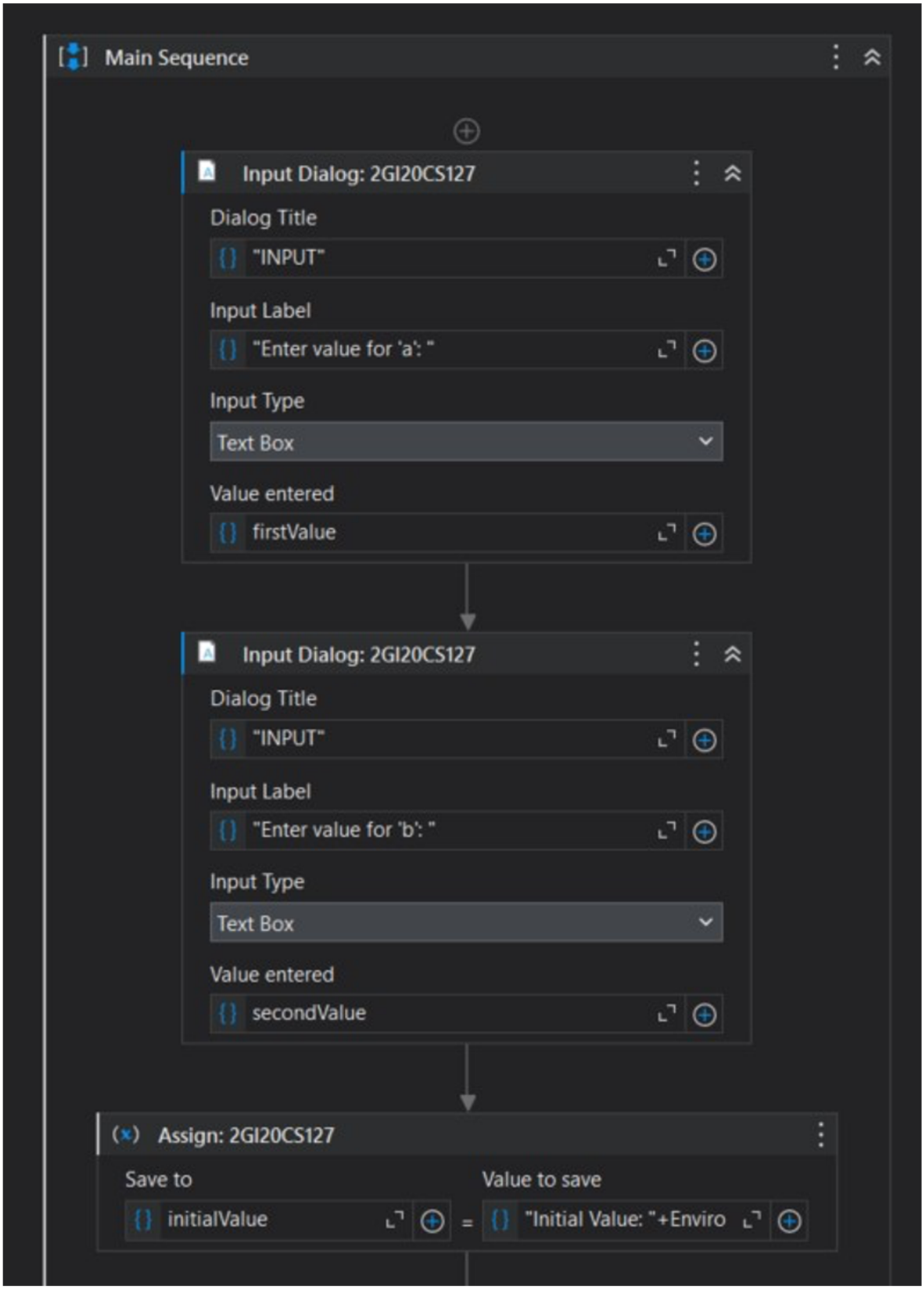
This termwork focuses on demonstrating how UiPath's activities can be used to facilitate user input, variable assignment, and displaying results through message boxes.

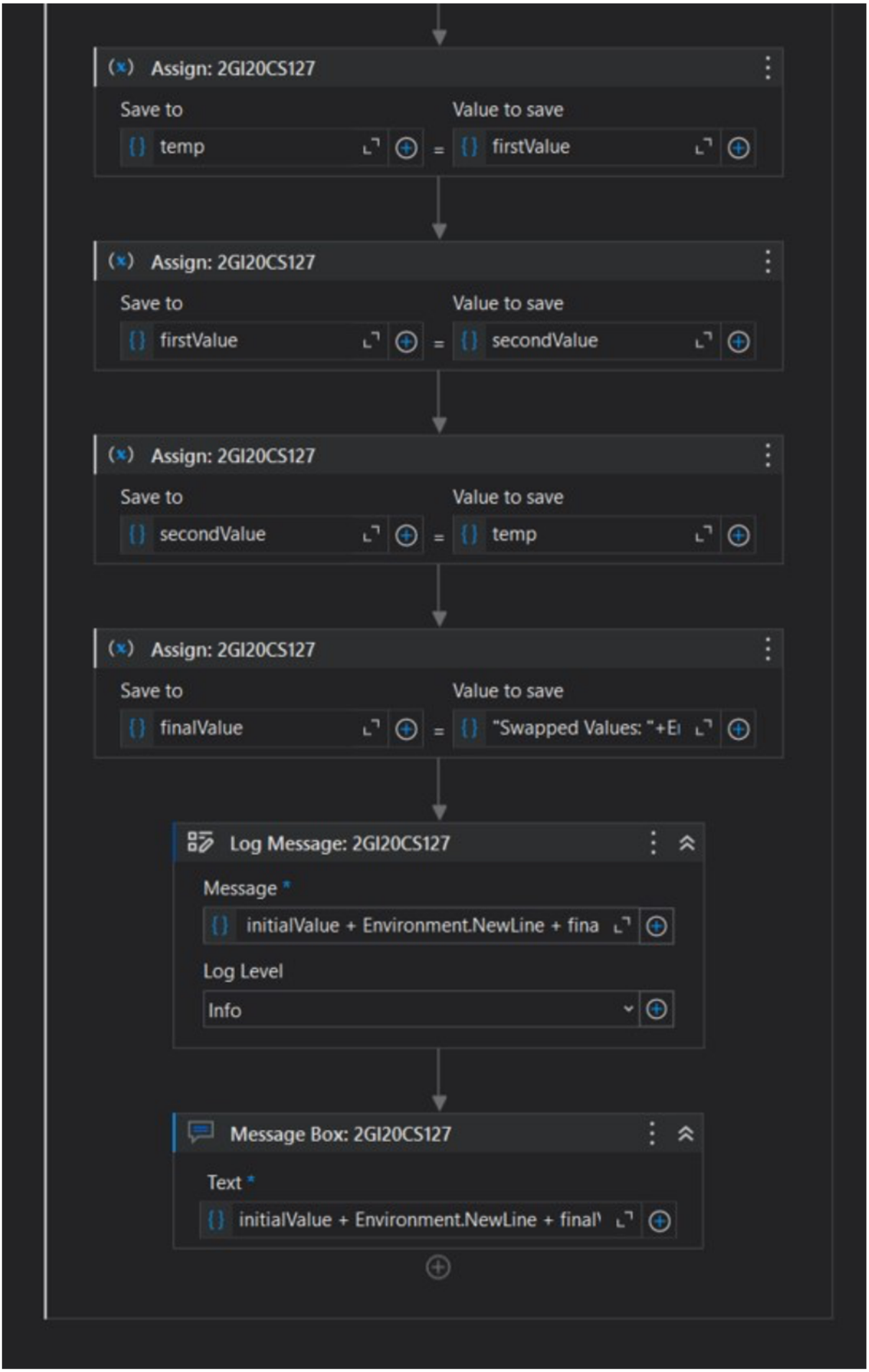
Input Dialog Activity: The Input Dialog activity is utilized to prompt the user for input during the automation process. It displays a dialog box where the user can enter values for the variables to be swapped. In this case, the input dialog is used to obtain the initial values of the variables from the user.

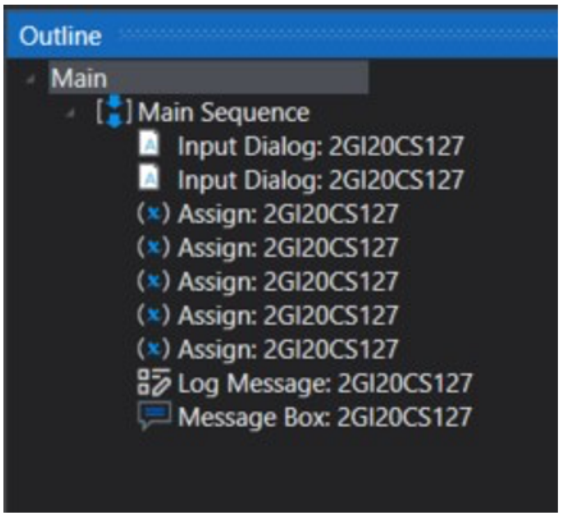
Assign Activity: The Assign activity is employed to assign values to variables within the workflow. It allows users to manipulate variables and perform operations on them. In the context of the variable swap termwork, the Assign activity is used to assign the input values to the respective variables and carry out the swapping operation using a temporary variable.

Message Box Activity: The Message Box activity is used to display information or output during the automation process. In the variable swap termwork, the Message Box activity is utilized to showcase the swapped values of the variables to the user.

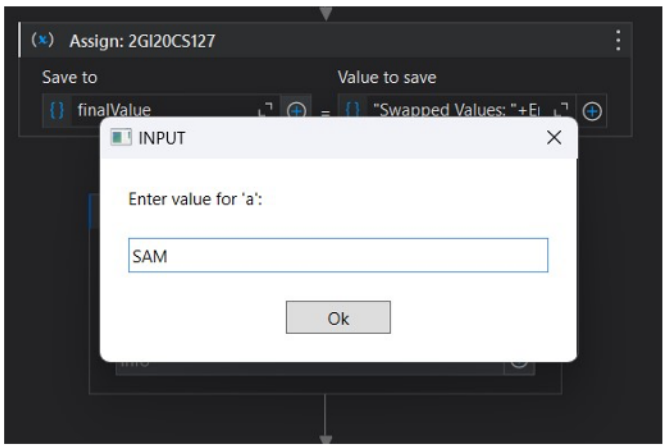
WORKFLOW AND OUTLINE:

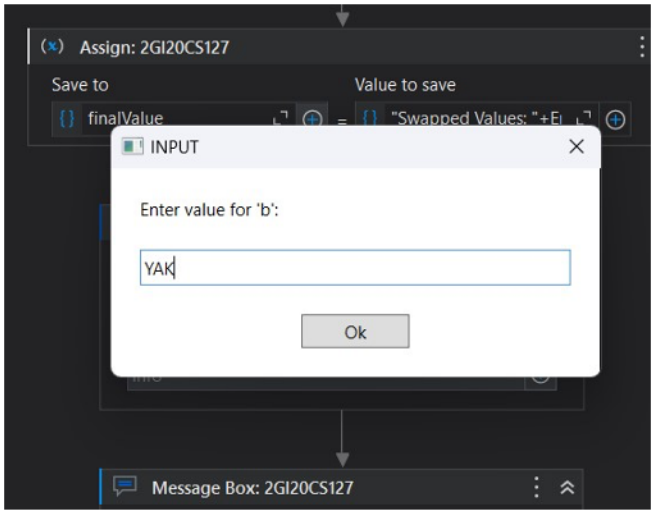


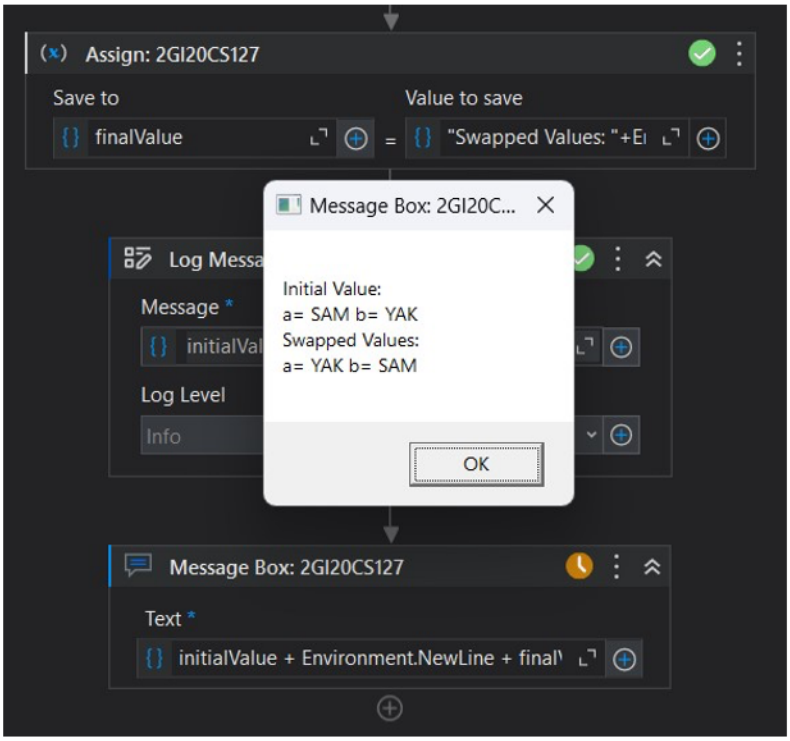

Assign: 2GI20CS127
Save to
temp
Value to save
firstValue
Assign: 2GI20CS127
Save to
firstValue
Value to save
secondValue
Assign: 2GI20CS127
Save to
secondValue
Value to save
temp
Assign: 2GI20CS127
Save to
finalValue
Value to save
"Swapped Values: "+E
Log Message: 2GI20CS127
Message *
initialValue + Environment.NewLine + fina
Log Level
Info
Message Box: 2GI20CS127
Text *
initialValue + Environment.NewLine + final

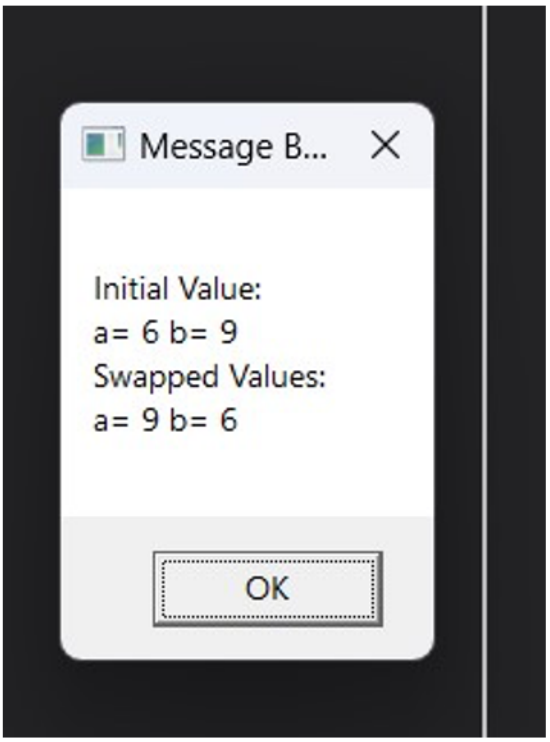

Outline
Main
Main Sequence
Input Dialog: 2GI20CS127
Input Dialog: 2GI20CS127
Assign: 2GI20CS127
Assign: 2GI20CS127
Assign: 2GI20CS127
Assign: 2GI20CS127
Assign: 2GI20CS127
Log Message: 2GI20CS127
Message Box: 2GI20CS127

OUTPUT(s):


Assign: 2GI20CS127
Save to
Value to save
finalValue
INPUT
Enter value for 'a':
SAM
Ok


Assign: 2GI20CS127
Save to
Value to save
finalValue
INPUT
Enter value for 'b':
YAK
Ok
Message Box: 2GI20CS127


Assign: 2GI20CS127
Save to
Value to save
finalValue
Message Box: 2GI20C...
Initial Value:
a= SAM b= YAK
Swapped Values:
a= YAK b= SAM
OK
Log Level
Info
Message Box: 2GI20CS127
Text *
initialValue + Environment.NewLine + final


Message B...
Initial Value:
a= 6 b= 9
Swapped Values:
a= 9 b= 6
OK

CONCLUSION:

In conclusion, involving the swapping of variables using UiPath's Input Dialog, Assign, and Message Box activities provides a practical application of workflow design and activity usage within UiPath Studio. By completing this exercise, individuals gain hands-on experience in building automation workflows that involve user input, variable assignment, and displaying results.

The outcome of this termwork is the successful execution of an automation process that prompts the user for input, swaps the values of two variables using a temporary variable, and displays the swapped values to the user through a message box. This exercise reinforces the understanding of fundamental programming concepts such as variable manipulation and the use of temporary variables.

By working on the variable swap termwork, we could become proficient in utilizing UiPath activities for input, assignment, and output but also develop problem-solving skills by applying these activities to achieve the desired result.

TERMWORK 3

DATE: 05/04/2023

PROBLEM STATEMENT:

Build a workflow that uses different Input Methods to input data in a Notepad.

i. Typeinto
ii. SimulateClick
iii. SendWindowMessages methods.

THEORY:

The workflow aims to demonstrate different input methods in automating data input in an application, specifically Microsoft Word instead of Notepad due to limitations encountered with Windows 11 Notepad automation.
The input methods employed in this workflow are Type Into, Simulate Click, and SendWindowMessages. By utilizing these methods, the workflow showcases how data can be entered, modified, and formatted within the Word application.

Activities Used:

Word Application Scope: The Word Application Scope activity is used to establish a connection with the Microsoft Word application. It allows the subsequent activities to interact with Word.

Type Into Activity: The Type Into activity is used to input text into the Word document. In this workflow, it is utilized to type the phrase "automation makes life easier" into the document.

Simulate Click Activity: The Simulate Click activity is employed to simulate a mouse click on a specific element within the Word application. Here, it is used to maximize the Word window, ensuring the entire document is visible.

SendWindowMessages Activity: The SendWindowMessages activity is used to send keyboard input directly to the Word application window. It is used to input the phrase "welcome to the world of automation" into the document.

Get Text Activity: The Get Text activity is used to extract the current font size of the selected text in the Word document. It captures the font size for further manipulation.

Assign Activity: The Assign activity is used to store the font size retrieved from the Get Text activity into a variable. It allows for further calculations and modifications.

Type Into Activity (for updating font size): The Type Into activity is employed to update the font size by adding 5 to the retrieved font size value. It is used to change the font size of the selected text.

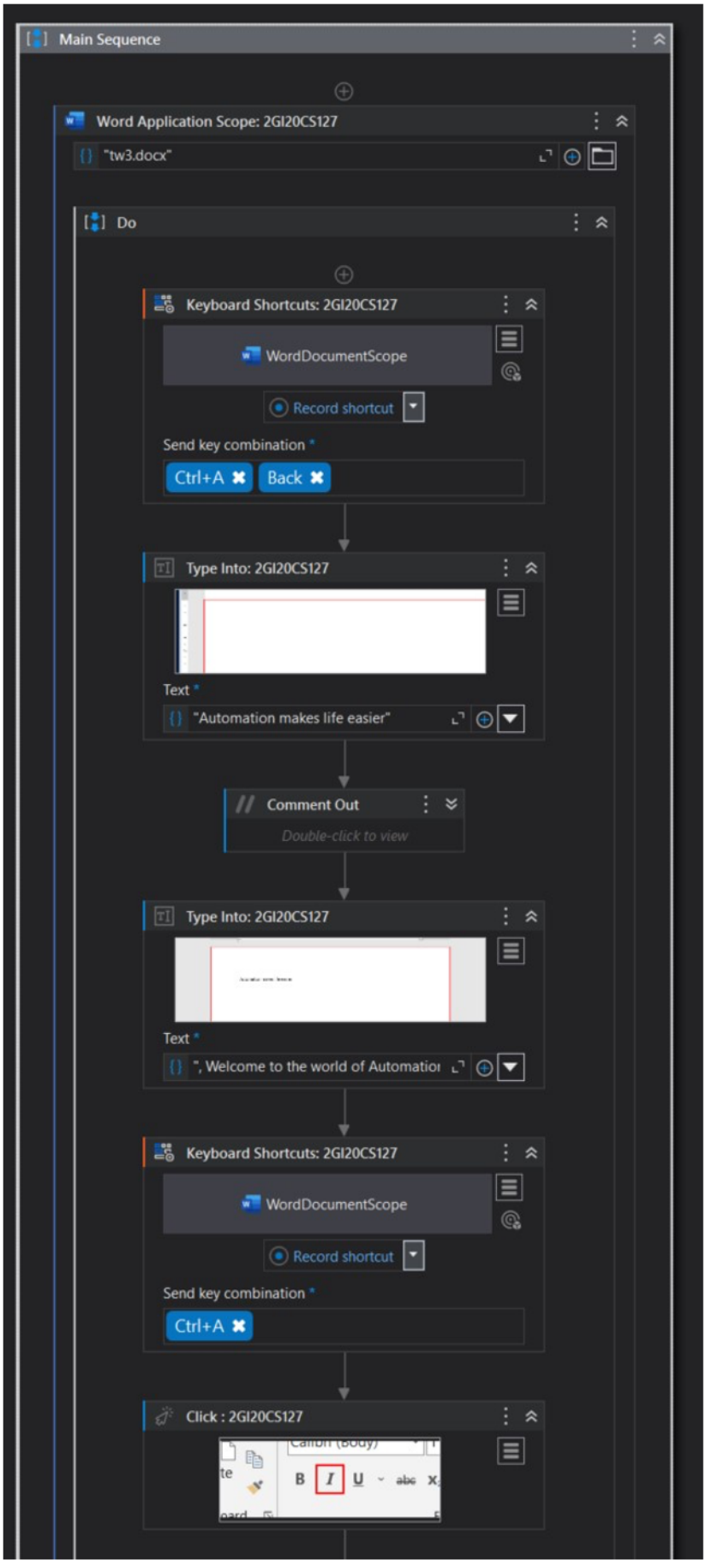
Send Hotkey Activity: The Send Hotkey activity is used to send a keyboard shortcut, specifically the "Enter" key. It is utilized to apply the updated font size to the selected text.

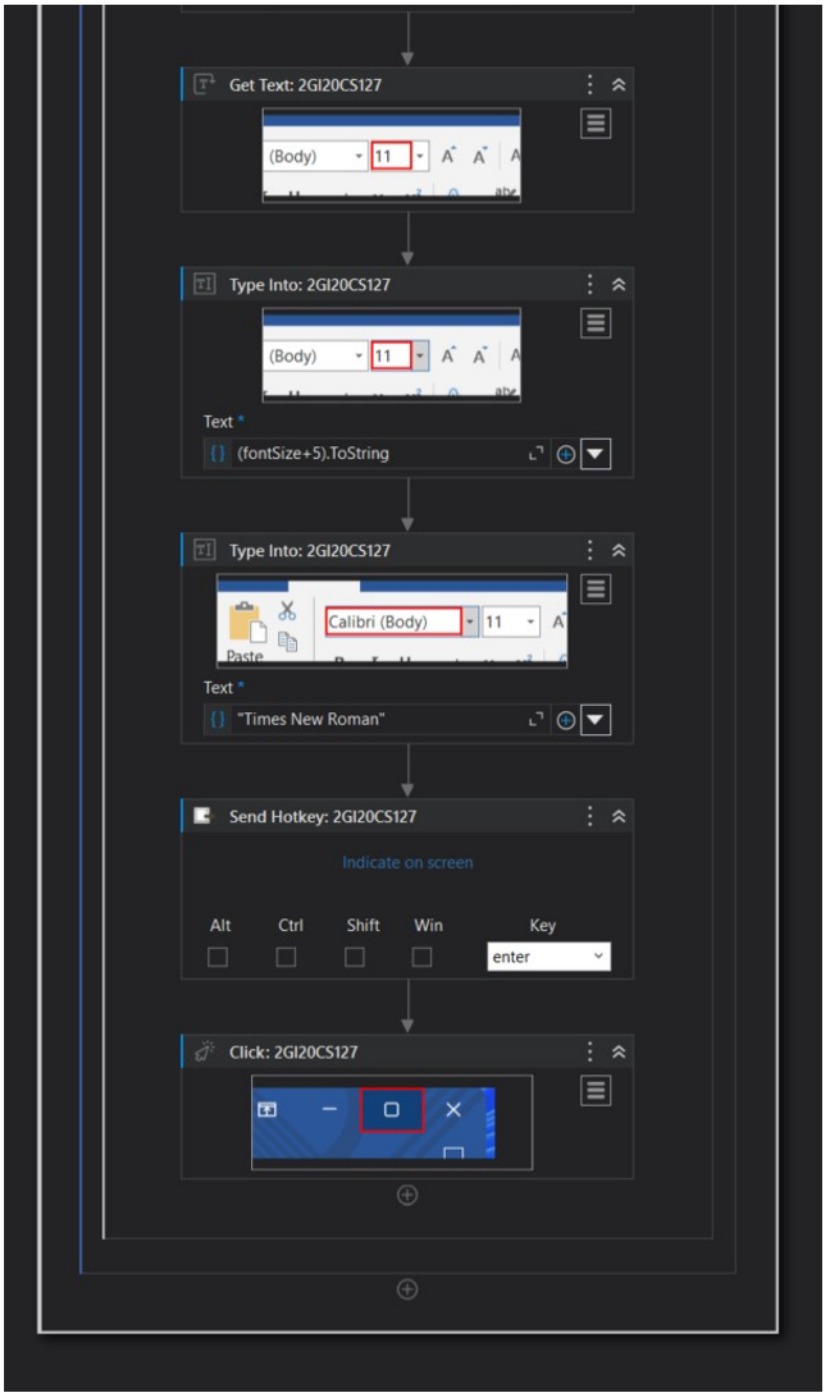
Packages:

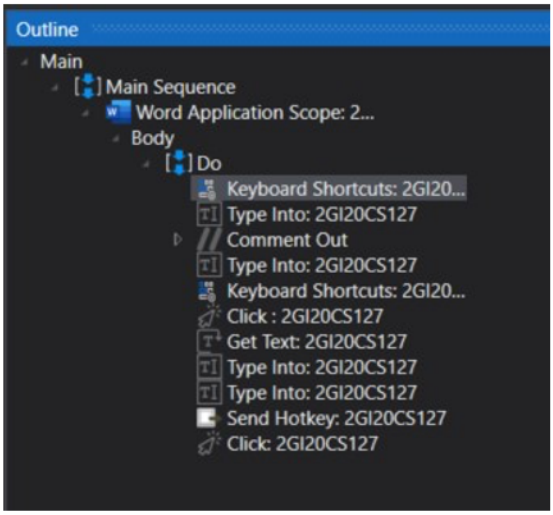
The workflow may not require additional packages beyond the core UiPath activities. However, to interact with Microsoft Word effectively, I used the following package:

UiPath.Word.Activities: This package provides a set of activities specifically designed for automating tasks within the Microsoft Word application. It offers activities for document manipulation, formatting, content extraction, and more.

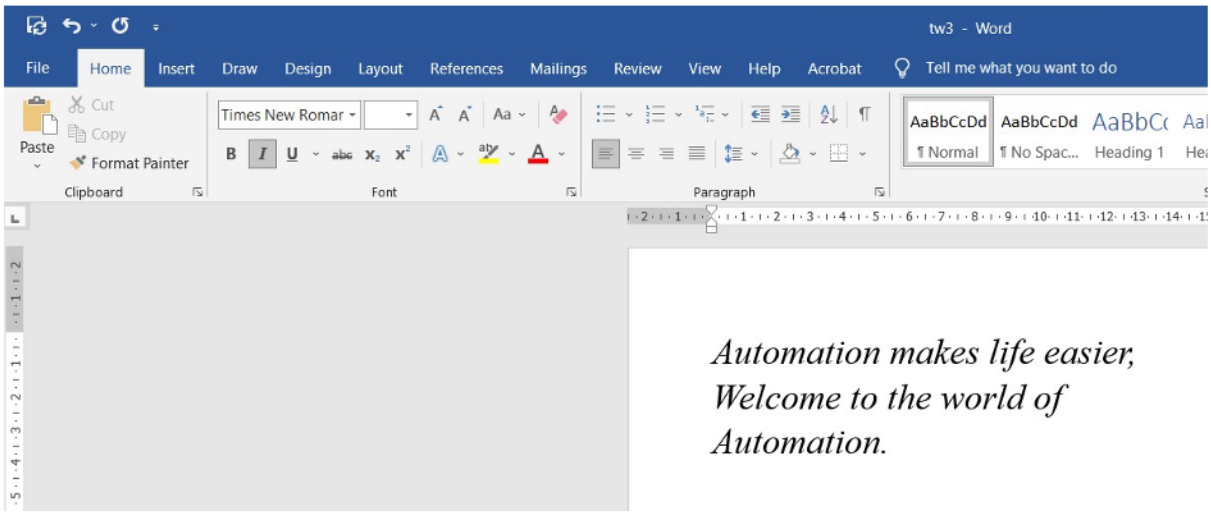
WORKFLOW AND OUTLINE:




Get Text: 2GI20CS127
(Body)
11
Type Into: 2GI20CS127
(Body)
11
Text *
(fontSize+5).ToString
Type Into: 2GI20CS127
Calibri (Body)
11
Paste
Text *
"Times New Roman"
Send Hotkey: 2GI20CS127
Indicate on screen
Alt
Ctrl
Shift
Win
Key
enter
Click: 2GI20CS127


Outline
Main
Main Sequence
Word Application Scope: 2...
Body
Do
Keyboard Shortcuts: 2GI20...
Type Into: 2GI20CS127
Comment Out
Type Into: 2GI20CS127
Keyboard Shortcuts: 2GI20...
Click : 2GI20CS127
Get Text: 2GI20CS127
Type Into: 2GI20CS127
Type Into: 2GI20CS127
Send Hotkey: 2GI20CS127
Click: 2GI20CS127

OUTPUT:



Automation makes life easier,
Welcome to the world of
Automation.

CONCLUSION:

In conclusion, this termwork involving the automation of data input and formatting in Microsoft Word using different input methods, such as Type Into, Simulate Click, and SendWindowMessages, demonstrates the power and flexibility of UiPath Studio in automating tasks within applications. Despite the change from Notepad to Word due to limitations in automating Windows 11 Notepad, the workflow effectively showcases the utilization of input methods to interact with and manipulate data in the Word application.

The outcomes of this termwork are as follows:

Successful Data Input: The workflow demonstrates the successful input of text into the Word document using the Type Into activity. The phrase "automation makes life easier" is efficiently entered into the document.

Window Maximization: The Simulate Click activity effectively maximizes the Word application window, ensuring the entire document is visible for further operations.

Input Using SendWindowMessages: The SendWindowMessages activity showcases an alternative input method by directly sending keyboard input to the Word application window. The phrase "welcome to the world of automation" is successfully inputted into the document.

Font Size Manipulation: The workflow extracts the font size of the selected text using the Get Text activity, stores it in a variable using the Assign activity, and then updates the font size by adding 5 using the Type Into activity. The updated font size is applied to the selected text using the Send Hotkey activity.

TERMWORK 4

DATE: 12/04/2023

PROBLEM STATEMENT:

Build a workflow that fills the form on RPAChallenge.com website with organized data from an excel file.

THEORY:

This termwork involves building a workflow that automates the process of filling out a form on the RPAChallenge.com website using organized data from an Excel file. The workflow utilizes UiPath activities to access the Excel file, read the data, interact with the website, and populate the form fields dynamically.

Activities Used:

Excel Application Scope: The Excel Application Scope activity establishes a connection with the Excel application and enables the subsequent activities to access and manipulate the data in the Excel file.

Read Range Activity: The Read Range activity is used to read the data from the Excel file into a DataTable. It retrieves the organized data required for populating the form on the RPAChallenge.com website.

For Each Row Activity: The For Each Row activity enables the iteration over each row of the DataTable, allowing the workflow to process the data for filling out the form.

Click Activity: The Click activity is used to initiate the form-filling process on the RPAChallenge.com website. It interacts with the "Start" button on the website to begin filling the form.

Parallel Activity: The Parallel activity is employed to execute multiple actions simultaneously. In this workflow, it contains six Anchor Base activities that target different text boxes in the website's form. The Anchor Base activities use Find Element activities to set anchors and Type Into activities to input the data from the Excel sheet into the respective text boxes.

Anchor Base and Find Element Activities: Within each Anchor Base activity, the Find Element activity is used to set an anchor element, which provides a stable reference point for locating the corresponding text box. The Find Element activity ensures that the workflow accurately identifies the target text box on the webpage.

Type Into Activity: Inside the Anchor Base activities, the Type Into activity is used to input the data from the Excel sheet into the respective text boxes on the website. The Type Into activity populates the form fields dynamically with the values from the Excel file.

Click Activity (Submit Button): After the Parallel activity, a Click activity is used to interact with the "Submit" button on the website, completing the form-filling process.

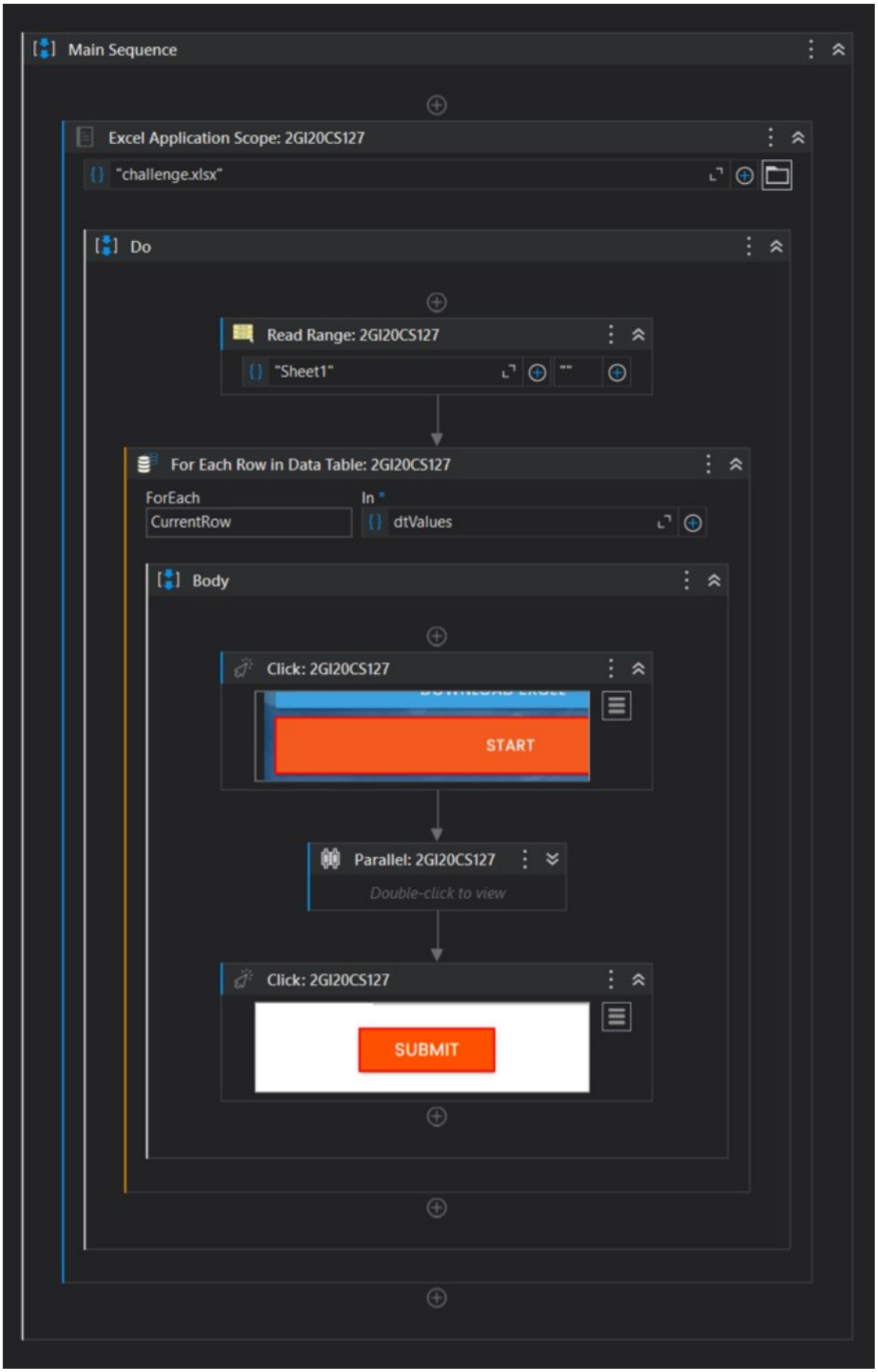
Packages:

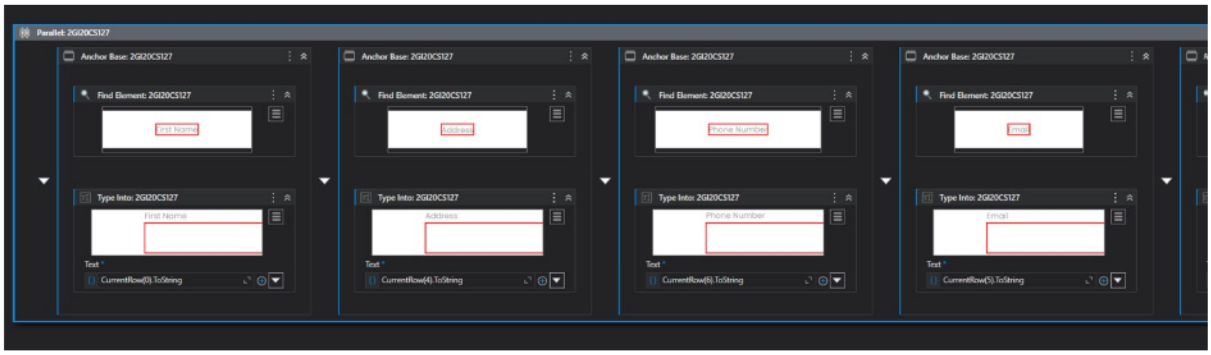
Following packages are used, depending on the specific needs and activities utilized:

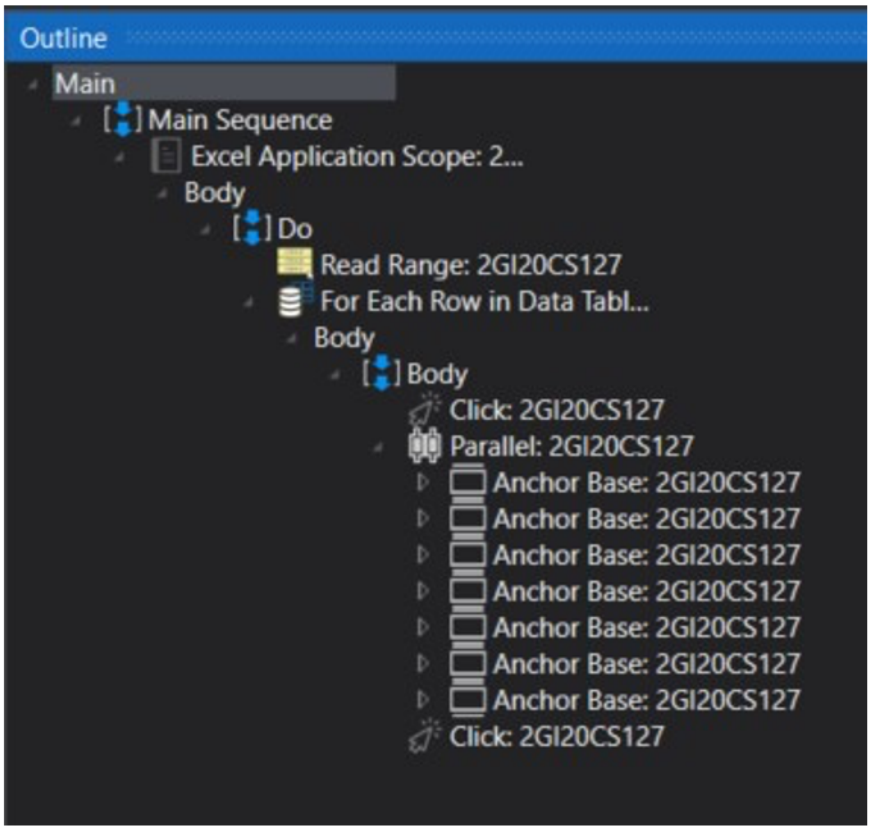
UiPath.Excel.Activities: This package provides activities for interacting with Excel files, such as reading data from worksheets, writing data to worksheets, and performing various Excel-related operations.

UiPath.UIAutomation.Activities: This package offers activities for automating interactions with the user interface of applications and web browsers. It includes activities for clicking, typing, finding elements, and more.

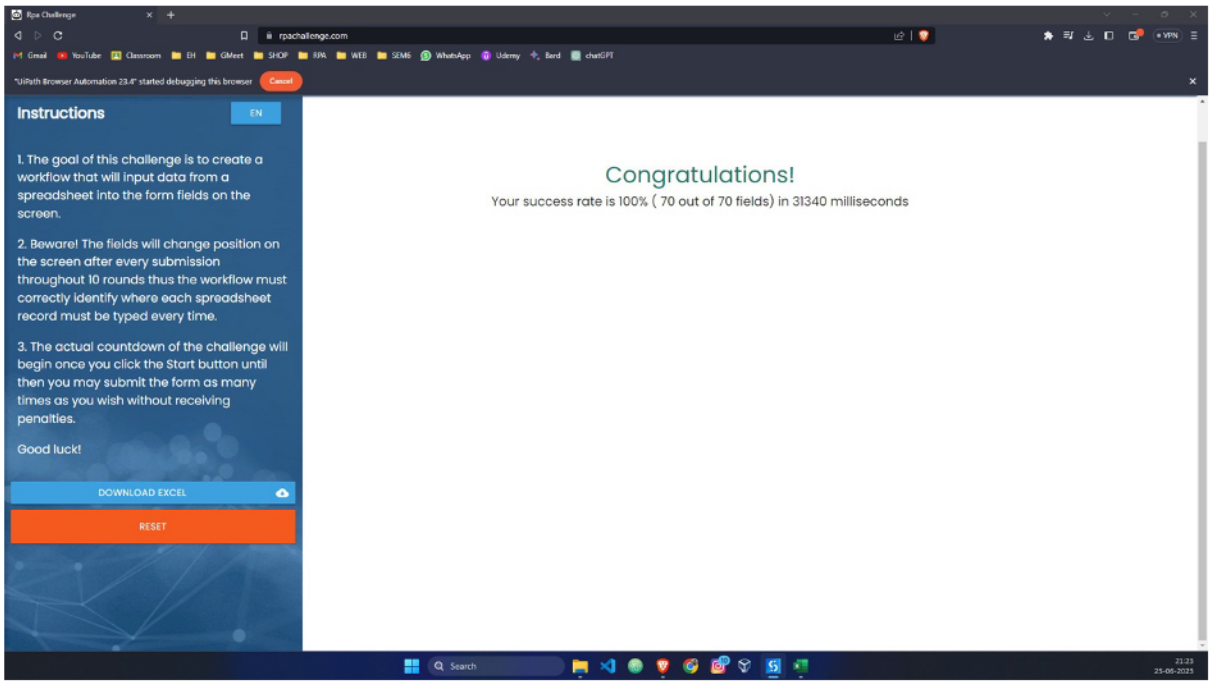
WORKFLOW AND OUTLINE:




Parallel: 2GI20CS127
Anchor Base: 2GI20CS127
Find Element: 2GI20CS127
First Name
Type Into: 2GI20CS127
Address
Phone Number
Email


Outline
Main
Main Sequence
Excel Application Scope: 2...
Body
Do
Read Range: 2GI20CS127
For Each Row in Data Tabl...
Body
Body
Click: 2GI20CS127
Parallel: 2GI20CS127
Anchor Base: 2GI20CS127
Anchor Base: 2GI20CS127
Anchor Base: 2GI20CS127
Anchor Base: 2GI20CS127
Anchor Base: 2GI20CS127
Anchor Base: 2GI20CS127
Anchor Base: 2GI20CS127
Click: 2GI20CS127

OUTPUT:


Rpa Challenge
rpachallenge.com
"UiPath Browser Automation 23.4" started debugging this browser
Cancel
Instructions
EN
1. The goal of this challenge is to create a workflow that will input data from a spreadsheet into the form fields on the screen.
2. Beware! The fields will change position on the screen after every submission throughout 10 rounds thus the workflow must correctly identify where each spreadsheet record must be typed every time.
3. The actual countdown of the challenge will begin once you click the Start button until then you may submit the form as many times as you wish without receiving penalties.
Good luck!
DOWNLOAD EXCEL
RESET
Congratulations!
Your success rate is 100% (70 out of 70 fields) in 31340 milliseconds
Search

CONCLUSION:

In conclusion, the termwork involving the automation of form filling on the RPAChallenge.com website using organized data from an Excel file demonstrates the effectiveness and efficiency of UiPath Studio in automating repetitive data entry tasks. By leveraging activities such as Excel Application Scope, Read Range, For Each Row, Click, Parallel, Anchor Base, and Find Element, the workflow successfully accesses the Excel data, interacts with the website, and populates the form fields dynamically.

The outcomes of this termwork are as follows:

Streamlined Data Entry: The workflow efficiently retrieves data from the Excel file using the Read Range activity and populates the form fields on the RPAChallenge.com website using the Type Into activity. This automation eliminates manual data entry, reducing the risk of errors and saving time.

Dynamic Form Filling: The workflow utilizes the Anchor Base and Find Element activities within the Parallel activity to locate the correct text boxes on the website's form. This dynamic approach allows for flexibility in adapting to changes in the form layout or structure, ensuring accurate data entry regardless of any modifications made to the website.

TERMWORK 5

DATE: 19/04/2023

PROBLEM STATEMENT:

a. Build a workflow using Switch activity that asks users' their eye color and displays their personality in a message box.

b. Build a workflow using While loop that tells the User if the input is a prime number or not.

(5a) THEORY:

The workflow is designed to interact with users and gather information about their eye color. It utilizes the Switch activity in UiPath Studio to handle different eye color choices provided by the user. Based on the input, the workflow displays the corresponding personality trait associated with the selected eye color using message boxes.

Activities Used:

Flowchart: The workflow is structured using a flowchart activity, which provides a visual representation of the logical flow of activities.

Input Dialog Activity: The Input Dialog activity is used to prompt the user to input their eye color choice. The input is expected as an integer value (1, 2, 3, or 4) corresponding to the available options.

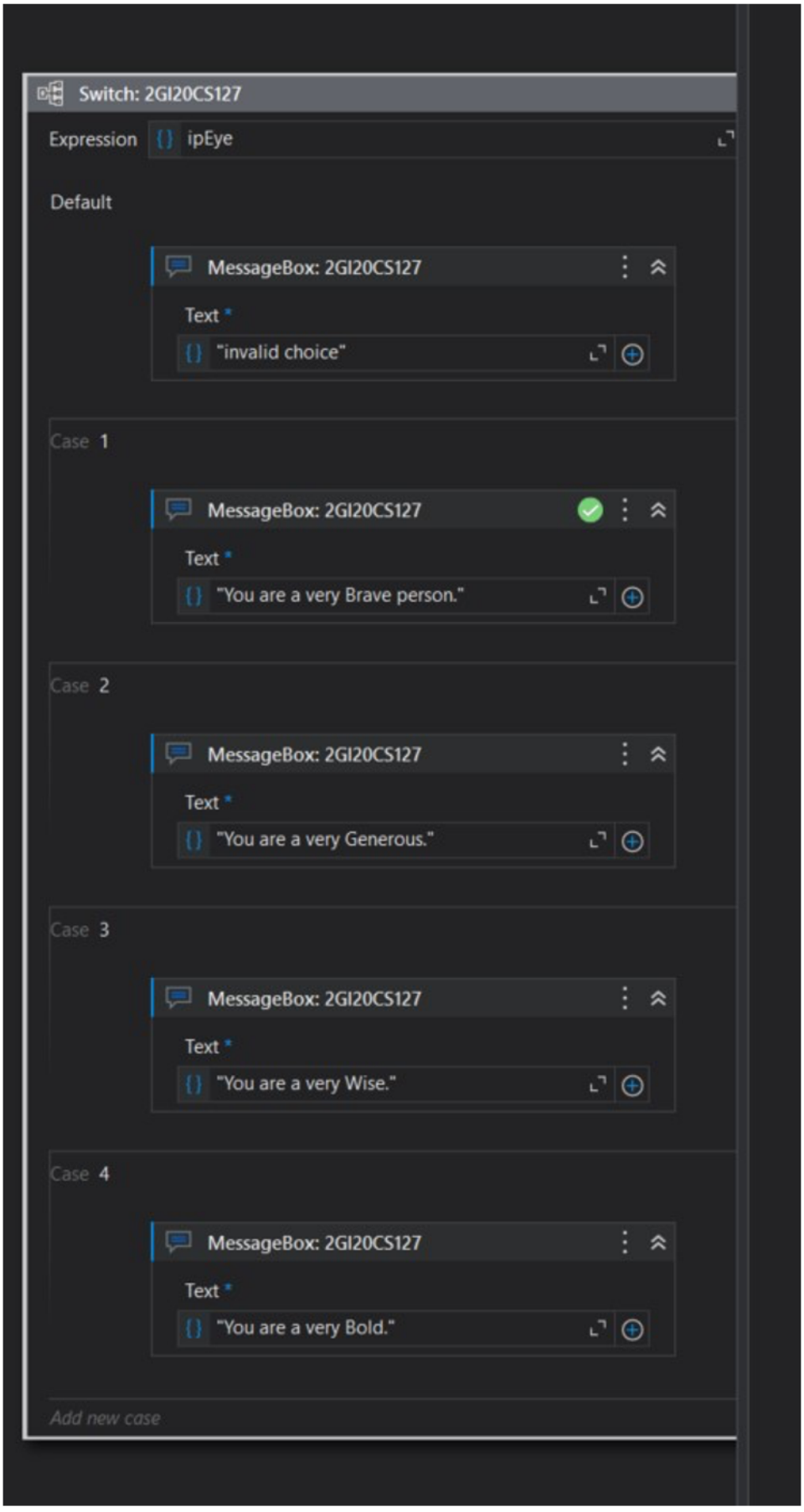
Switch Activity: The Switch activity is used to evaluate the user's input and execute different branches of the workflow based on the selected eye color.

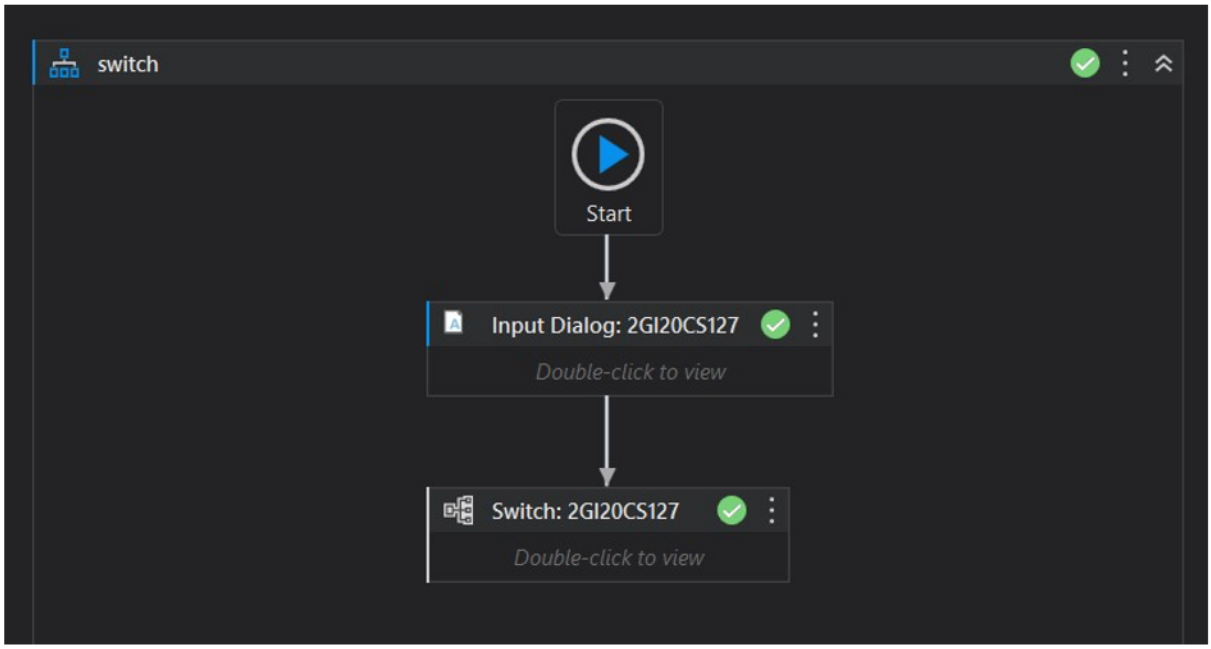
Message Box Activities: The Message Box activity is utilized within each case of the Switch activity to display the personality trait associated with the selected eye color. The message boxes provide a visual output to the user.

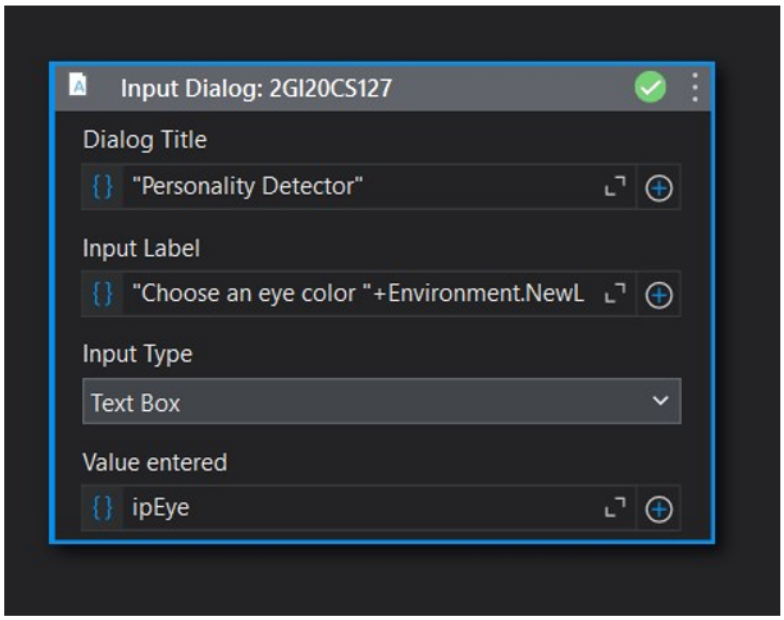
Packages:

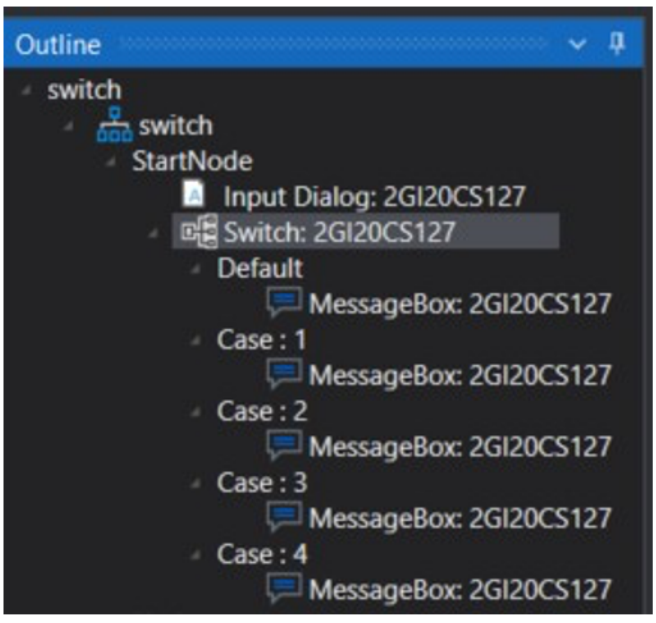
The workflow does not require any additional packages beyond the default activities provided by UiPath Studio. The activities used, such as Flowchart, Input Dialog, Switch, and Message Box, are part of the core activities available in UiPath.

(5a) WORKFLOW AND OUTLINE:

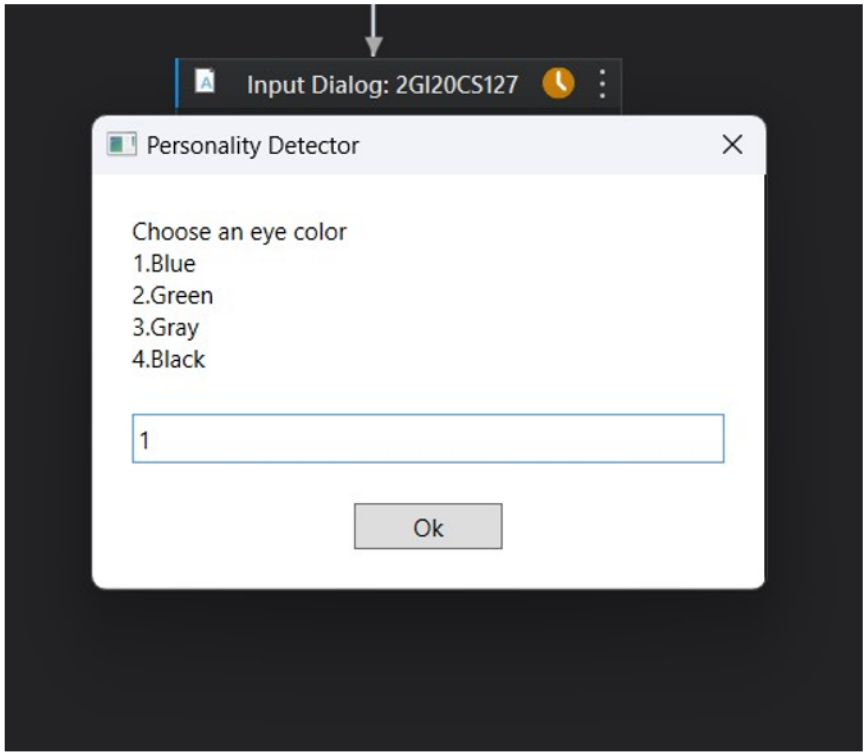


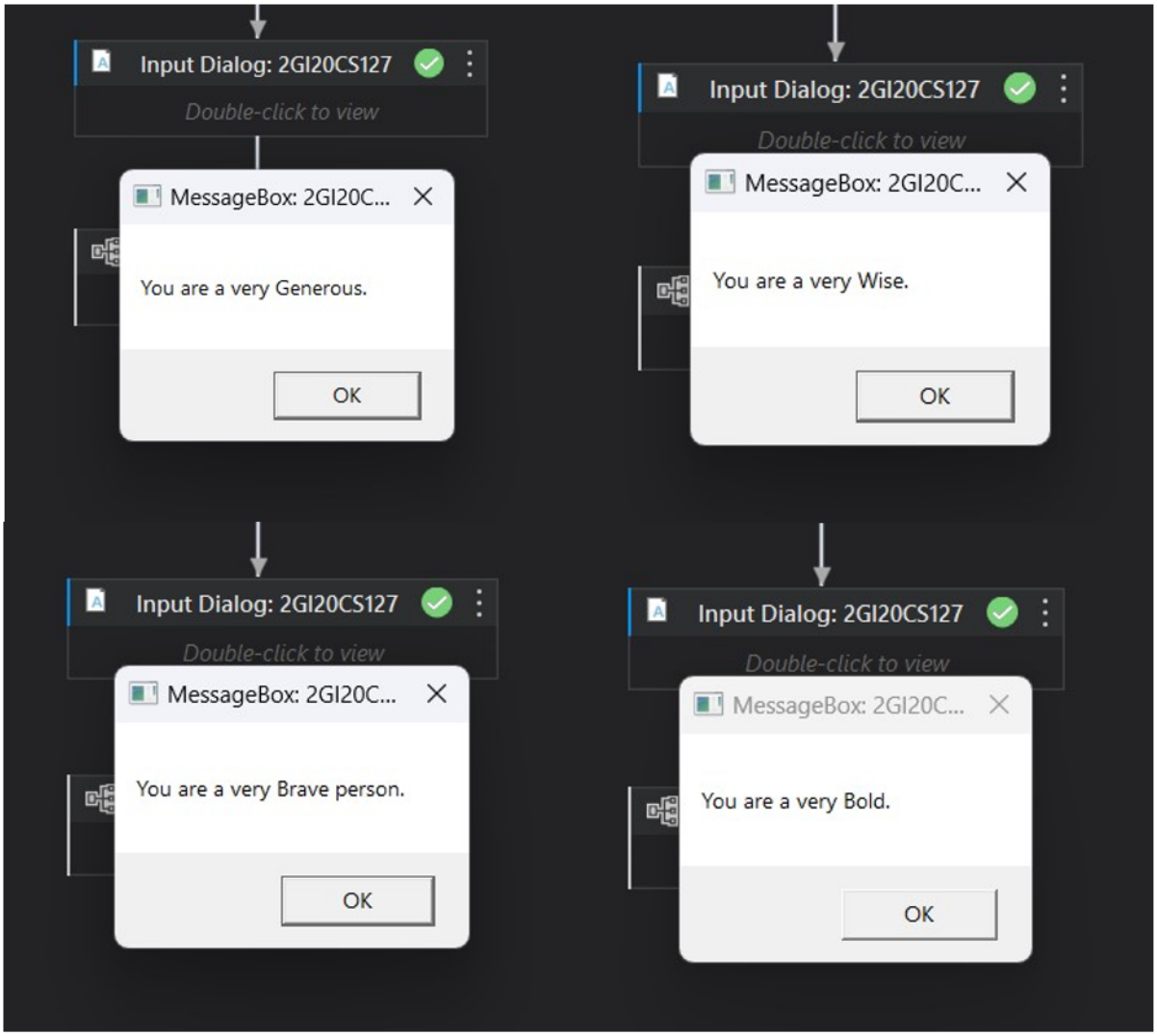

switch
Start
Input Dialog: 2GI20CS127
Double-click to view
Switch: 2GI20CS127
Double-click to view


Input Dialog: 2GI20CS127
Dialog Title
"Personality Detector"
Input Label
"Choose an eye color "+Environment.NewL
Input Type
Text Box
Value entered
ipEye


Outline
switch
switch
StartNode
Input Dialog: 2GI20CS127
Switch: 2GI20CS127
Default
MessageBox: 2GI20CS127
Case : 1
MessageBox: 2GI20CS127
Case : 2
MessageBox: 2GI20CS127
Case : 3
MessageBox: 2GI20CS127
Case : 4
MessageBox: 2GI20CS127

(5a) OUTPUT:


Input Dialog: 2GI20CS127
Personality Detector
Choose an eye color
1.Blue
2.Green
3.Gray
4.Black
1
Ok


Input Dialog: 2GI20CS127
Double-click to view
MessageBox: 2GI20C...
You are a very Generous.
OK
Input Dialog: 2GI20CS127
Double-click to view
MessageBox: 2GI20C...
You are a very Wise.
OK
Input Dialog: 2GI20CS127
Double-click to view
MessageBox: 2GI20C...
You are a very Brave person.
OK
Input Dialog: 2GI20CS127
Double-click to view
MessageBox: 2GI20C...
You are a very Bold.
OK

(5b) THEORY:

The following termwork involves building a workflow using a While loop to determine if a user-inputted number is a prime number or not. The workflow utilizes a flowchart in UiPath Studio and incorporates activities such as Input Dialog, While, Assign, If, and Message Box. It provides a user-friendly way to check the primality of a given number and provides appropriate feedback based on the result.

Activities Used:

Flowchart: The flowchart activity serves as the structure for organizing the workflow, allowing for visual representation and easy navigation between different activities.

Input Dialog Activity: The Input Dialog activity prompts the user to enter a number. The user's input is stored in the "ipVal" variable, which will be evaluated for primality.

While Activity: The While activity implements a loop that continues until a specific condition is met. In this case, the loop continues as long as the value of "i" is less than or equal to "ipVal".

Assign Activity: The Assign activity is used to assign values to variables within the workflow. In this workflow, the Assign activity is used to increment the "flag" variable when the condition "(ipVal Mod i) = 0" is met, indicating a factor of the number.

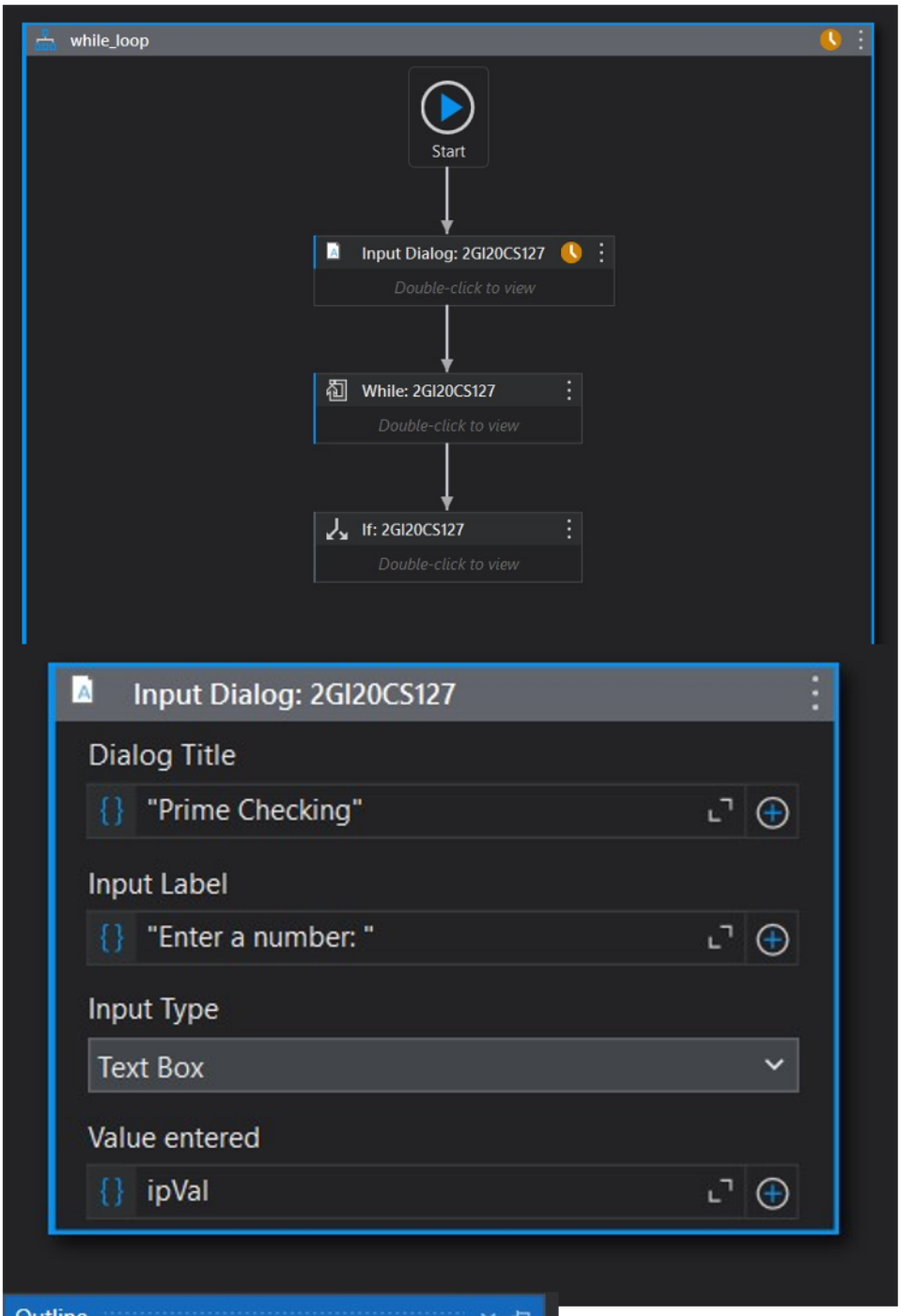
If Activity: The If activity evaluates the condition "flag = 2" to determine if the number is prime or not. If the condition is true, it means that the number has only two factors (1 and itself) and is therefore prime. If the condition is false, it means that the number has more than two factors and is not prime.

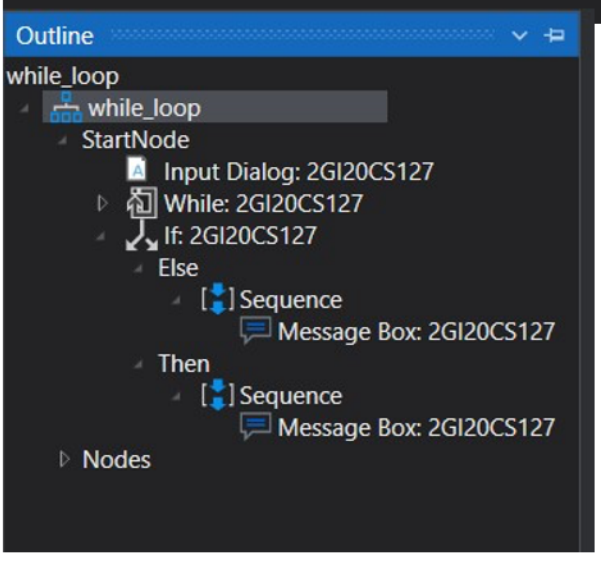
Message Box Activity: The Message Box activity is used to display messages to the user. In this workflow, it is used to show whether the inputted number is a prime number or not.

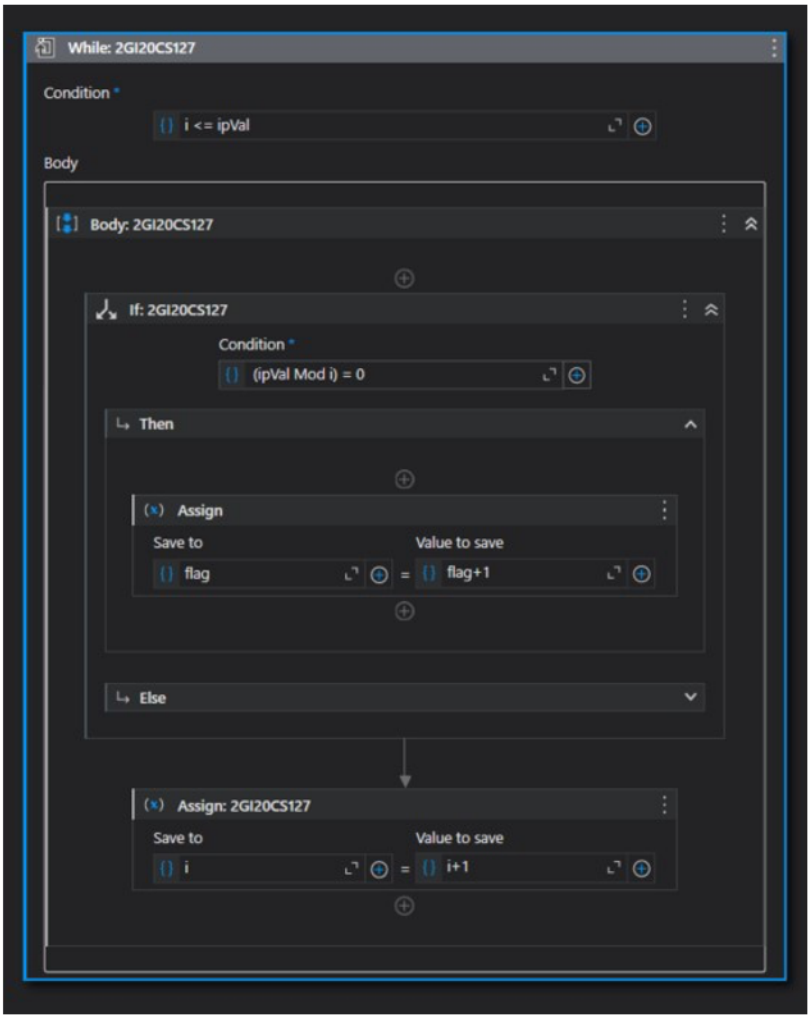
Packages:

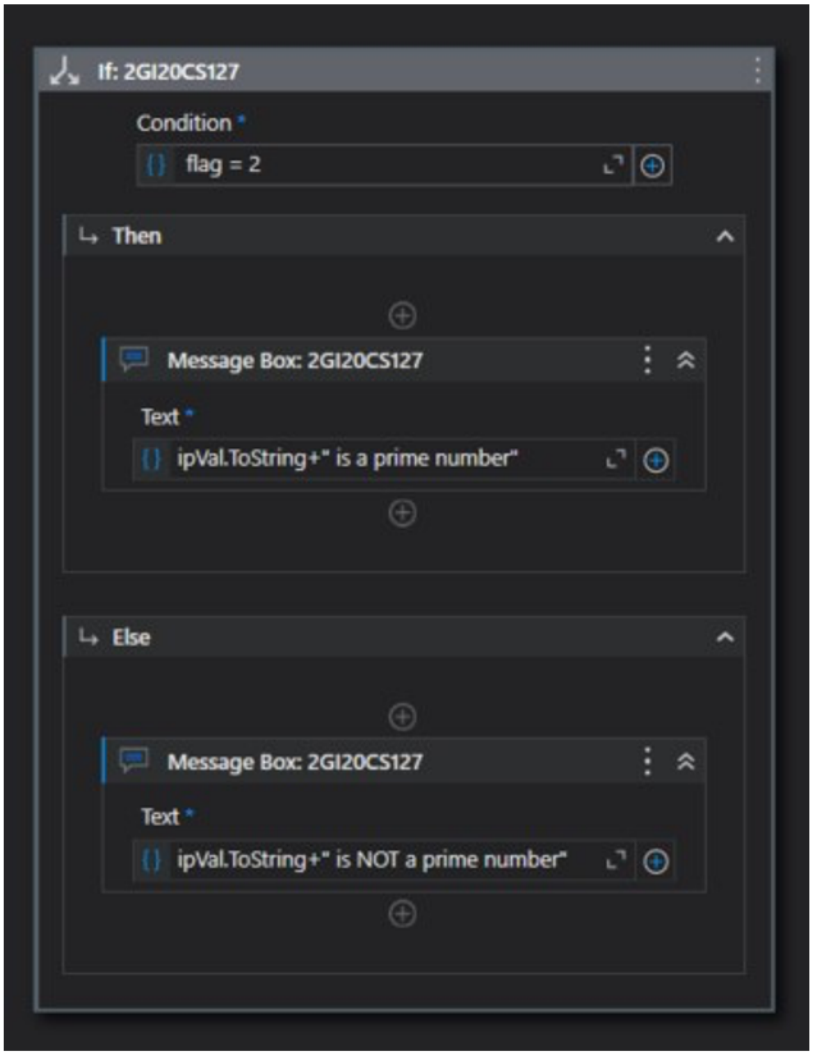
The workflow does not require any additional packages beyond the default activities provided by UiPath Studio. The activities used, such as Flowchart, Input Dialog, While, Assign, If, and Message Box, are part of the core activities available in UiPath.

(5b) WORKFLOW AND OUTLINE:

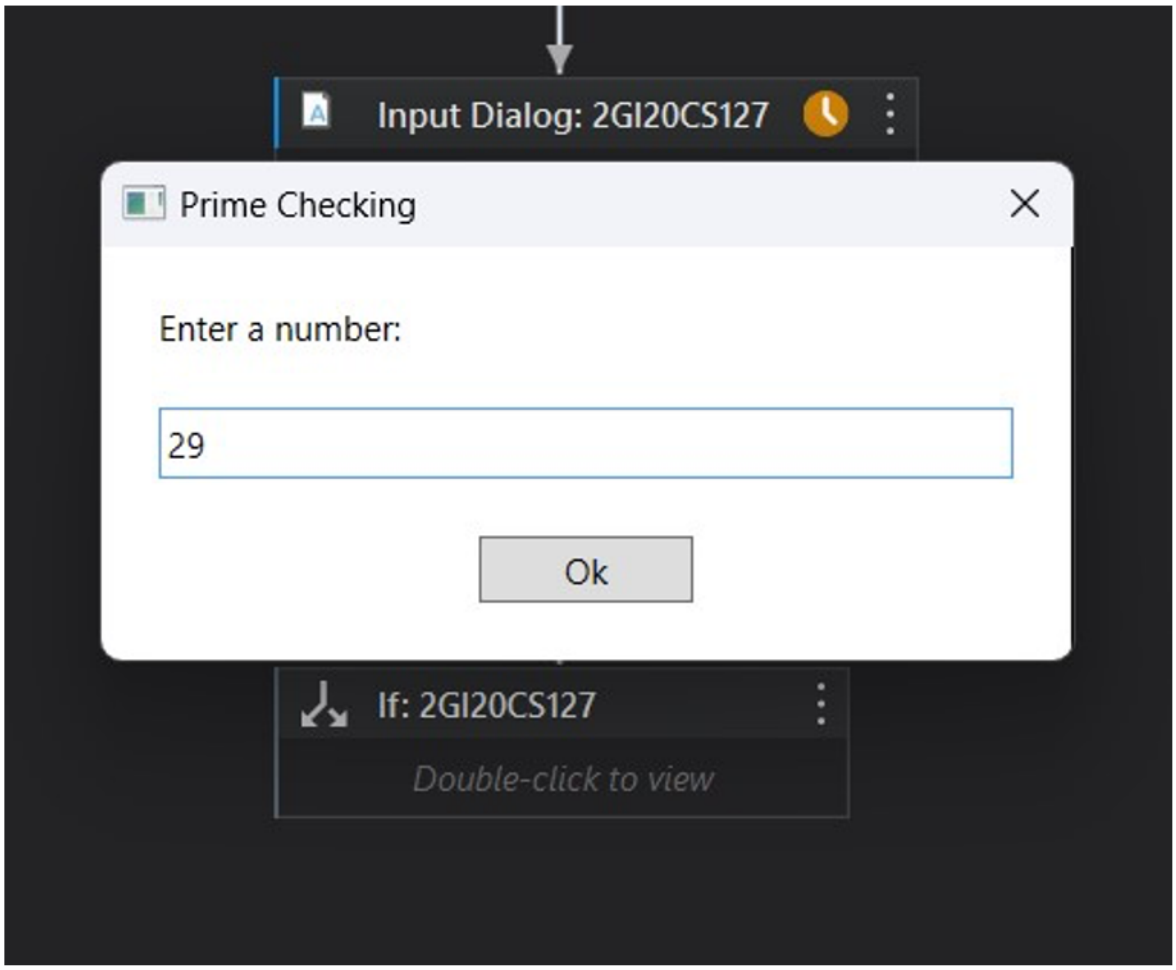


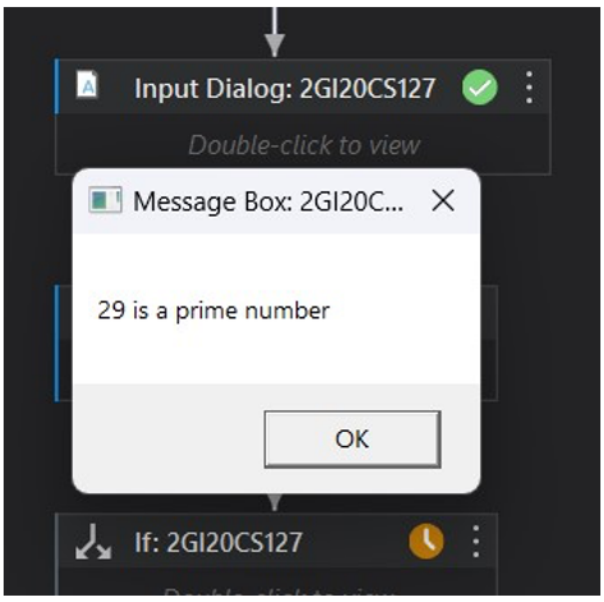



While: 2GI20CS127
Condition *
i <= ipVal
Body
Body: 2GI20CS127
If: 2GI20CS127
Condition *
(ipVal Mod i) = 0
Then
Assign
Save to
Value to save
flag
=
flag+1
Else
Assign: 2GI20CS127
Save to
Value to save
i
=
i+1


If: 2GI20CS127
Condition *
flag = 2
Then
Message Box: 2GI20CS127
Text *
ipVal.ToString+" is a prime number"
Else
Message Box: 2GI20CS127
Text *
ipVal.ToString+" is NOT a prime number"

(5b) OUTPUT:


Input Dialog: 2GI20CS127
Prime Checking
Enter a number:
29
Ok
If: 2GI20CS127
Double-click to view


Input Dialog: 2GI20CS127
Double-click to view
Message Box: 2GI20C...
29 is a prime number
OK
If: 2GI20CS127


Input Dialog: 2GI20C...
Message Box: 2GI20...
9 is NOT a prime number
OK
If: 2GI20CS127
Double-click to view

CONCLUSION:

a) In conclusion, The use of the Switch activity to gather users' eye color and display their personality traits through message boxes showcases the versatility of UiPath Studio in creating interactive workflows. By incorporating user input and branching logic, the workflow provides a personalized experience for users based on their eye color selection.

b) In conclusion, The use of a While loop to determine if a user-inputted number is a prime number or not demonstrates the power of UiPath Studio in automating mathematical computations. By utilizing activities such as Input Dialog, While, Assign, If, and Message Box, the workflow provides a user-friendly and efficient way to check the primality of a given number.

The outcomes of this termwork are as follows:

User Input Handling: The Switch termwork allows users to provide input regarding their eye color choice. The workflow effectively captures and processes this input, ensuring proper handling of user preferences.
Decision-Making: The Switch activity evaluates the user's input and directs the workflow to different branches based on the selected eye color. This outcome showcases the ability to make decisions and execute specific actions based on user choices.
Personalized Feedback: The use of Message Box activities within each case of the Switch activity enables the workflow to provide personalized feedback to users.
Error Handling: The default case in the Switch activity ensures that if the user inputs an invalid choice

User Input Validation: The While termwork allows users to input a number to determine if it is a prime number. The workflow validates the user's input, ensuring that only valid numbers are processed.
Primality Testing: The While loop evaluates the primality of the user's inputted number by checking for factors other than 1 and the number itself. The workflow accurately determines whether the number is prime or not based on this evaluation.
Efficient Looping: The While loop continues until a specific condition is met, allowing for efficient iteration and evaluation of larger numbers. This outcome showcases the ability to handle repetitive tasks and calculations effectively.
User-Friendly Results: The workflow provides clear and concise feedback to the user regarding the primality of the inputted number. The use of Message Box activities ensures that users receive understandable and informative results.

TERMWORK 6

DATE: 26/04/2023

PROBLEM STATEMENT:

Build a workflow using Format, Join, IndexOf, Split, and Substring methods that extracts key information from a text and prints in a different format.

THEORY:

The following termwork involves building a workflow that utilizes various string manipulation methods to extract key information from a given text and display it in a different format. The workflow employs activities such as Assign and Message Box and incorporates methods like Format, Join, IndexOf, Split, and Substring. This automation aims to showcase the power of UiPath Studio in processing and transforming text data efficiently.

Activities Used:

Assign Activity: The Assign activity is used to assign values to variables within the workflow. In this termwork, the Assign activity is utilized to store and manipulate the given text, extract specific information, and format it for display.

Message Box Activity: The Message Box activity is employed to display the formatted information extracted from the text. It provides a user-friendly interface to present the results to the user.

Methods Used:

Format Method: The Format method is used to format the extracted information and create a formatted string for display. It allows for the insertion of variables and values into a specified pattern or template.

Join Method: The Join method is used to combine multiple string elements into a single string. It is utilized to concatenate specific words or phrases extracted from the text.

IndexOf Method: The IndexOf method is employed to determine the position of a specific character or substring within a string. It is utilized to identify the starting point for extracting relevant information from the text.

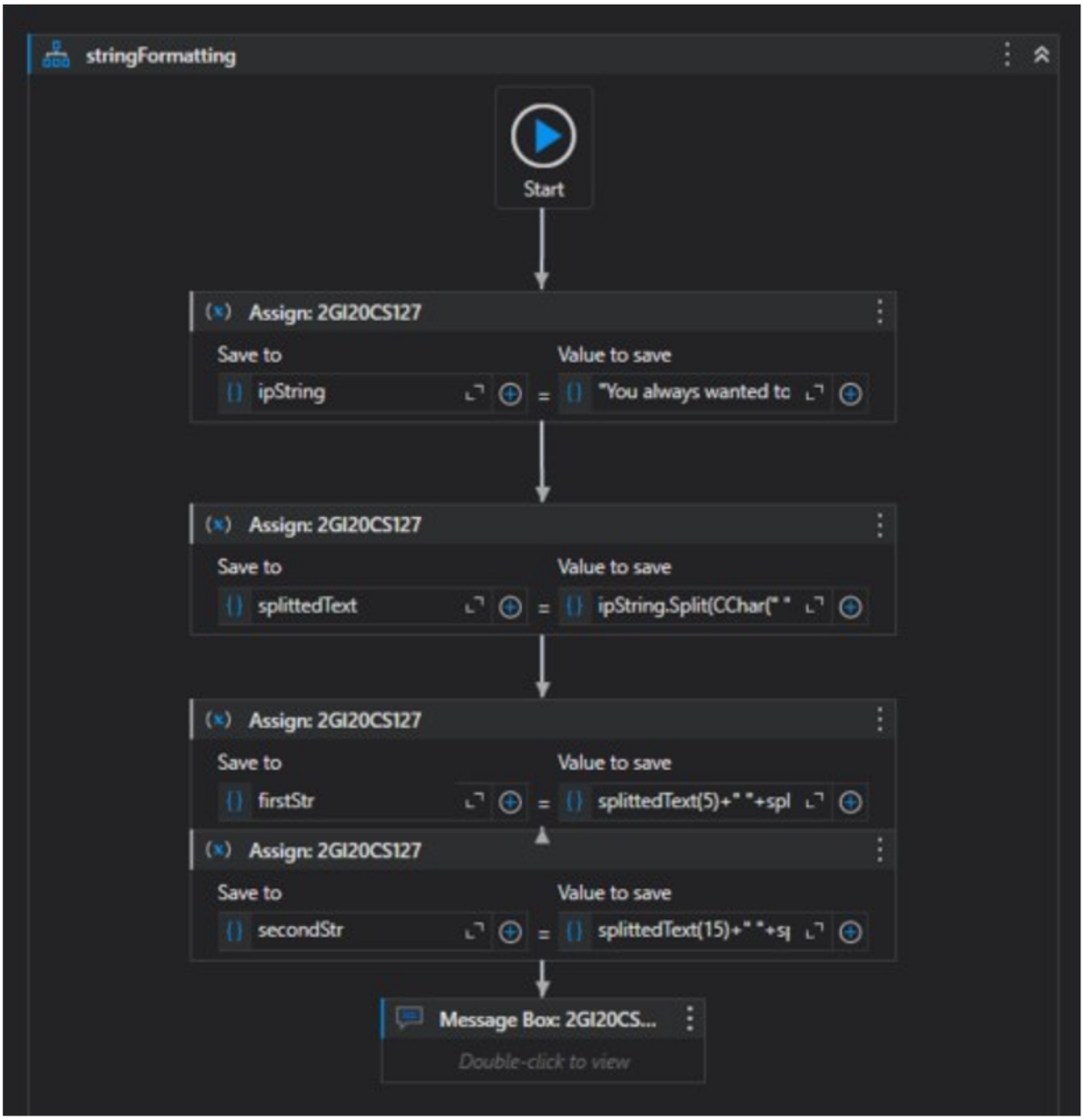
Split Method: The Split method is used to split a string into an array of substrings based on a specified delimiter. It enables the extraction of individual words or phrases from the given text.

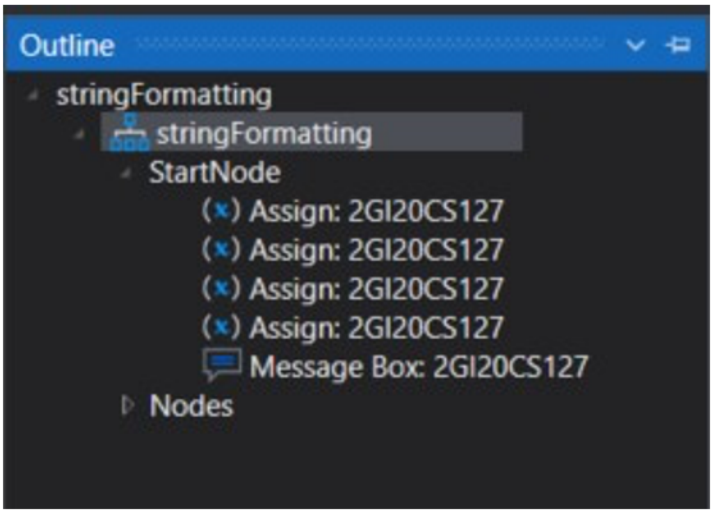
Substring Method: The Substring method is employed to extract a specific portion of a string based on the starting and ending indices. It allows for the retrieval of desired segments of the text.

Packages:

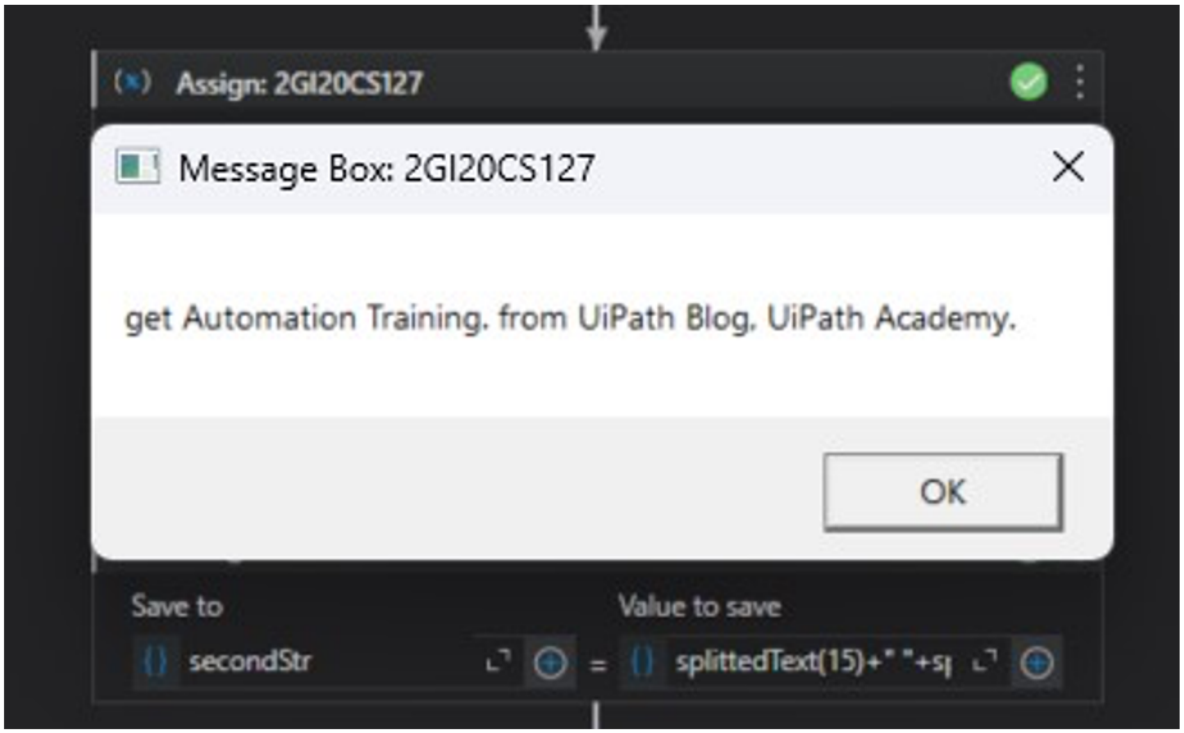
The workflow does not require any additional packages beyond the default activities provided by UiPath Studio.

WORKFLOW AND OUTLINE:





OUTPUT:


Assign: 2GI20CS127
Message Box: 2GI20CS127
get Automation Training. from UiPath Blog. UiPath Academy.
OK
Save to
secondStr
=
Value to save
splittedText(15)+" "+s

CONCLUSION:

In conclusion, the termwork involving the utilization of string manipulation methods in UiPath Studio showcases the ability to extract and format key information from a given text. By employing activities such as Assign and Message Box, and incorporating methods like Format, Join, IndexOf, Split, and Substring, the workflow demonstrates efficient text processing capabilities.

The outcomes and benefits of this termwork are as follows:

Information Extraction: The workflow successfully extracts specific pieces of information from the given text, such as the desired phrases or words. This outcome highlights the ability to locate and isolate relevant data within a larger text.

Format Transformation: By utilizing the Format method, the workflow transforms the extracted information into a different format or structure. This capability enables the creation of a formatted string that can be displayed or further processed.

Efficient String Manipulation: The use of Join, IndexOf, Split, and Substring methods allows for effective manipulation of the text data. These methods provide the ability to concatenate, search, split, and extract specific portions of the text, enabling precise and controlled processing.

User-Friendly Output: The Message Box activity presents the formatted information in a user-friendly manner. It provides a clear and easily understandable display of the extracted and formatted data.

Automation of Text Processing: The workflow showcases the power of UiPath Studio in automating text processing tasks. By utilizing string manipulation methods, it eliminates the need for manual extraction and formatting, saving time and effort.

TERMWORK 7

DATE: 10/05/2023

PROBLEM STATEMENT:

Build a workflow using data table activities to join two library databases using matching student ID and display output in a message box.

THEORY:

The following termwork involves building a workflow that utilizes data table activities in UiPath Studio to join two library databases based on matching student IDs. The workflow reads data from two Excel sheets, performs an inner join operation using the Join Data Table activity, and displays the joined output in a message box. This automation aims to showcase the capabilities of UiPath Studio in handling and manipulating data tables to perform database-related tasks efficiently.

Activities Used:

Excel Application Scope: The Excel Application Scope activity is used to specify the Excel file ("data.xlsx") and perform operations on it, such as reading data from specific sheets and writing data to other sheets.

Read Range: The Read Range activity is utilized to read the data from the specified sheets ("Sheet1" and "Sheet2") of the Excel file. It retrieves the data and stores it in data tables (table1 and table2) for further processing.

Join Data Table: The Join Data Table activity is employed to join the two data tables (table1 and table2) based on the matching student IDs. By specifying the join conditions and type, it combines the data from both tables into a single output data table ("outputData").

Write Range: The Write Range activity is used to write the joined output data table ("outputData") to a specified sheet ("Sheet4") in the Excel file. This activity ensures that the joined data is saved for future reference or analysis.

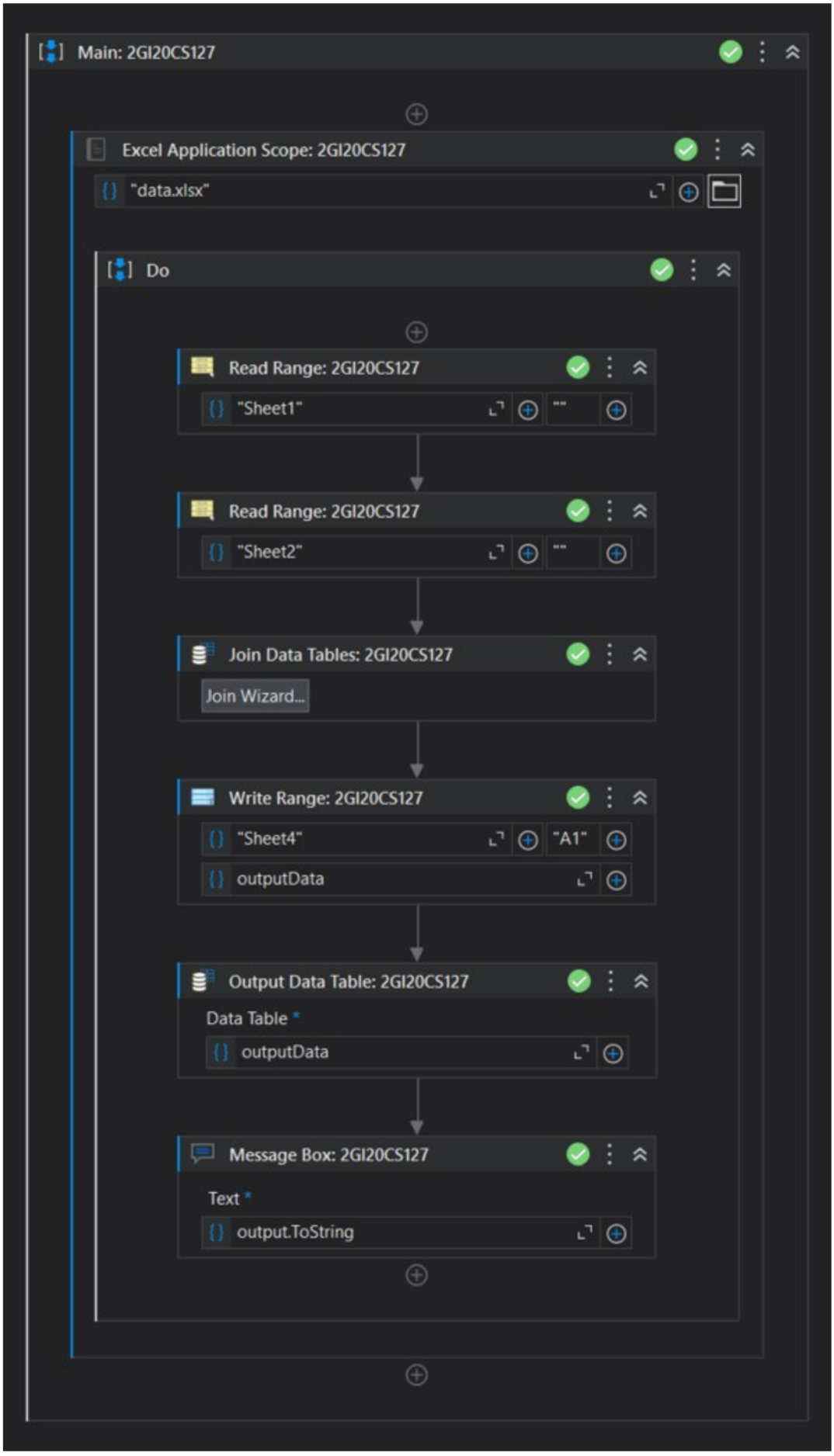
Output Data Table: The Output Data Table activity is utilized to store the joined output data table ("outputData") in a variable ("output") for further usage or display.

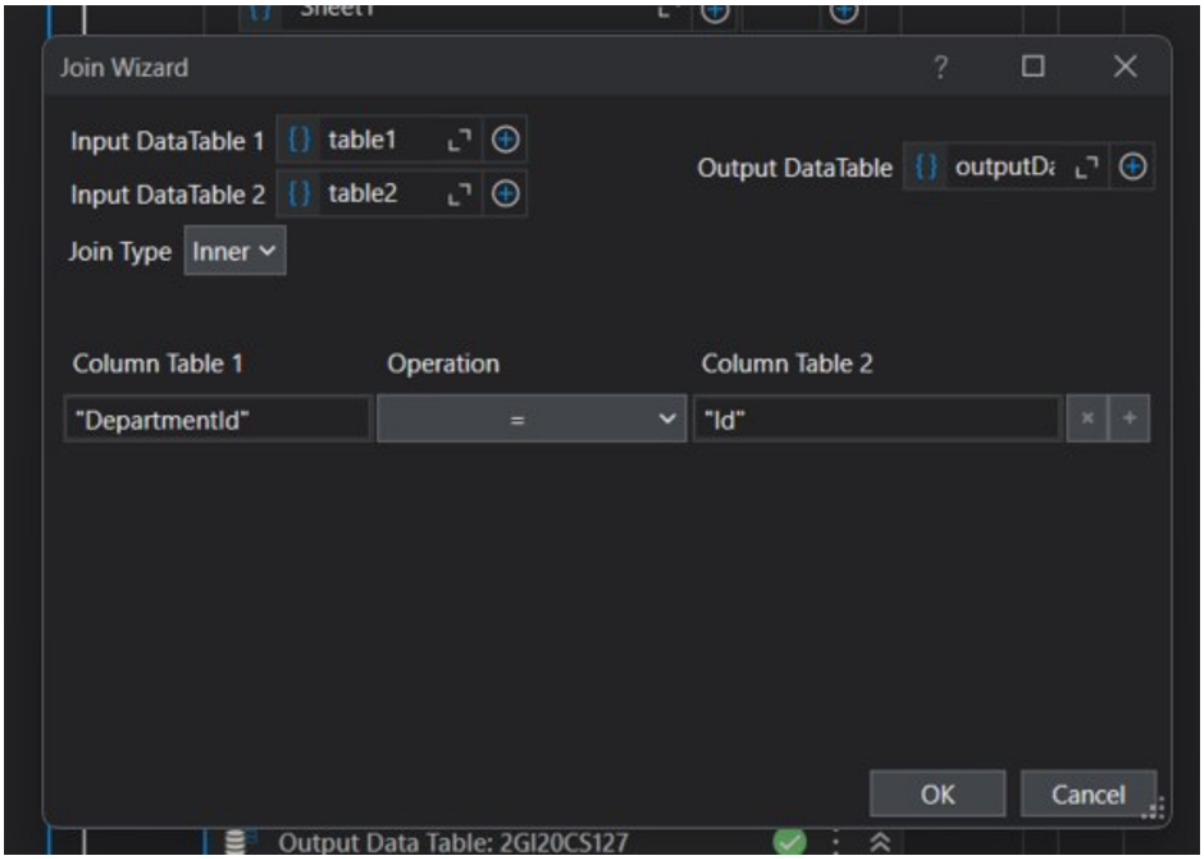
Message Box: The Message Box activity is employed to display the joined output data table ("output") in a pop-up message box. It provides a user-friendly interface to present the results of the data joining operation.

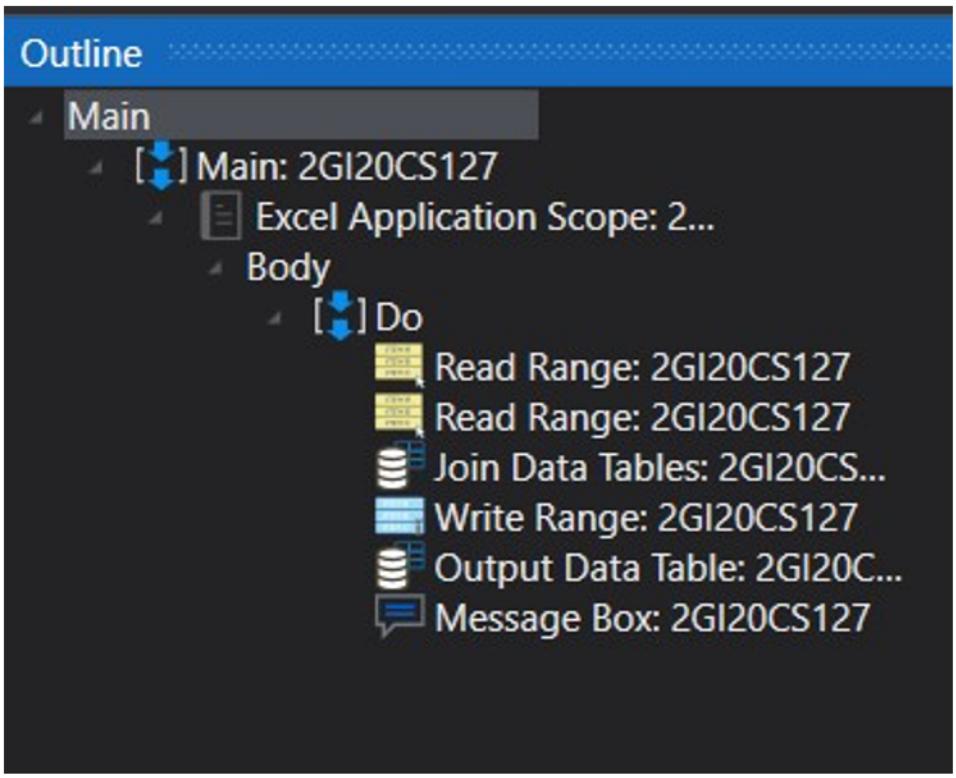
Packages:

The workflow utilizes the default activities provided by UiPath Studio and does not require any additional packages for data table operations.

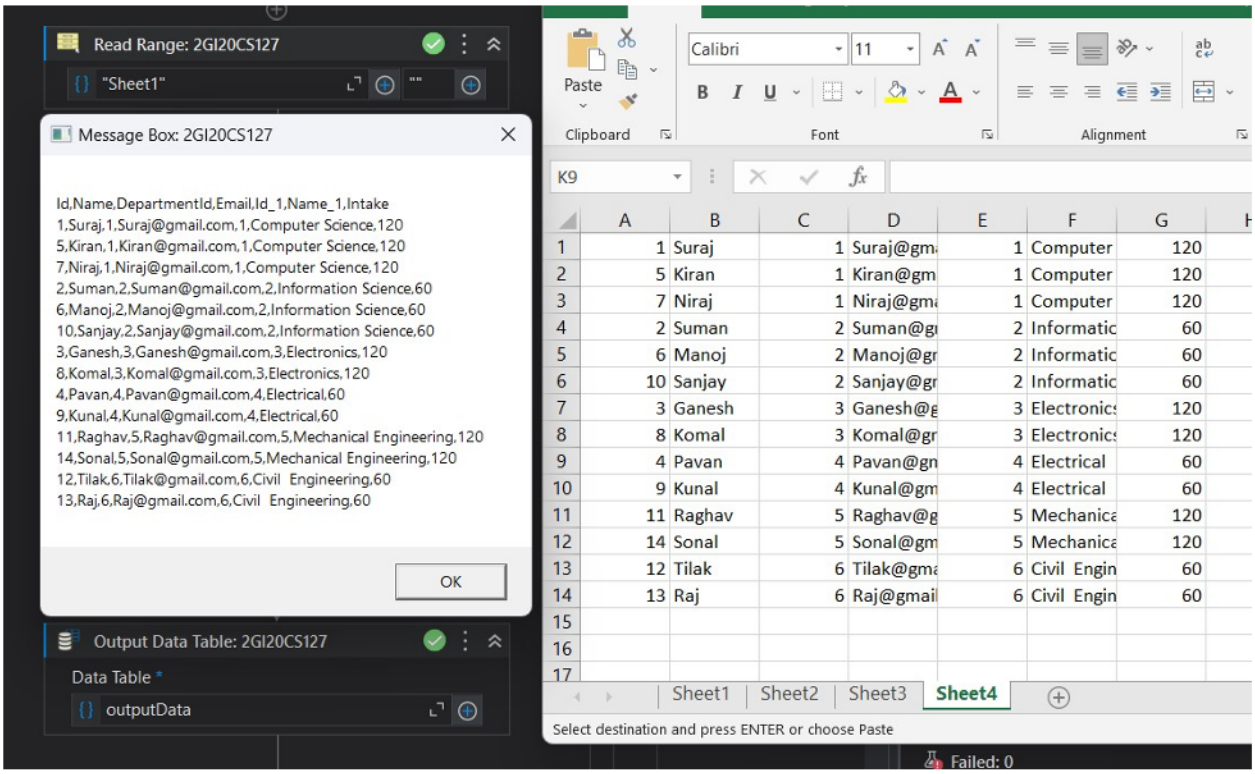
WORKFLOW AND OUTLINE:




Join Wizard
Input DataTable 1
table1
Input DataTable 2
table2
Output DataTable
outputDa
Join Type
Inner
Column Table 1
Operation
Column Table 2
"DepartmentId"
=
"Id"
OK
Cancel
Output Data Table: 2GI20CS127


Outline
Main
Main: 2GI20CS127
Excel Application Scope: 2...
Body
Do
Read Range: 2GI20CS127
Read Range: 2GI20CS127
Join Data Tables: 2GI20CS...
Write Range: 2GI20CS127
Output Data Table: 2GI20C...
Message Box: 2GI20CS127

OUTPUT:



	A	B	C	D	E	F	G
1	1	Suraj	1	Suraj@gm	1	Computer	120
2	5	Kiran	1	Kiran@gm	1	Computer	120
3	7	Niraj	1	Niraj@gm	1	Computer	120
4	2	Suman	2	Suman@g	2	Informatic	60
5	6	Manoj	2	Manoj@gr	2	Informatic	60
6	10	Sanjay	2	Sanjay@gr	2	Informatic	60
7	3	Ganesh	3	Ganesh@g	3	Electronics	120
8	8	Komal	3	Komal@gr	3	Electronics	120
9	4	Pavan	4	Pavan@gn	4	Electrical	60
10	9	Kunal	4	Kunal@gm	4	Electrical	60
11	11	Raghav	5	Raghav@g	5	Mechanica	120
12	14	Sonal	5	Sonal@gm	5	Mechanica	120
13	12	Tilak	6	Tilak@gma	6	Civil Engin	60
14	13	Raj	6	Raj@gmail	6	Civil Engin	60
15							
16							
17							

CONCLUSION:

In conclusion, the termwork involving the utilization of data table activities in UiPath Studio successfully demonstrates the capability to join two library databases based on matching student IDs. By utilizing activities such as Read Range, Join Data Table, Write Range, Output Data Table, and Message Box, the workflow efficiently processes and presents the joined data in a user-friendly manner.

The outcomes and benefits of this termwork are as follows:

Data Integration: The workflow successfully combines data from two separate Excel sheets representing different library databases. By performing an inner join operation based on matching student IDs, the workflow integrates relevant information from both databases into a single output data table.

Improved Data Analysis: The joined output data table provides a consolidated view of the data from both databases. This outcome enhances the ability to analyze and extract insights from the combined data, enabling more comprehensive and informed decision-making.

Efficient Automation: By utilizing data table activities, the workflow automates the process of joining databases, eliminating the need for manual data manipulation. This automation saves time and effort, allowing for faster and more efficient data processing.

Data Persistence: The joined output data table is written back to a specified sheet in the Excel file, ensuring the persistence and availability of the joined data for future reference or analysis. This outcome facilitates data storage and retrieval, enhancing data management practices.

User-Friendly Output: The Message Box activity displays the joined output data table in a pop-up message box, providing a clear and easily understandable format for presenting the results to the user. This outcome enhances the user experience and facilitates quick data validation.

TERMWORK 8

DATE: 17/05/2023

PROBLEM STATEMENT:

a. Build a workflow using Screen Scraper Wizard that scrapes text using Full-Text scraping method and stores it in a Notepad file.

b. Build a workflow using Data Scraping wizard that scrapes blog post titles from UiPath Blog from multiple pages.

(8a) THEORY:

This termwork involves building a workflow in UiPath Studio that utilizes the Screen Scraper Wizard to scrape text from a web page using the Full-Text scraping method. The scraped text is then stored in a Notepad file. This automation showcases the capabilities of UiPath Studio in extracting and saving specific text content from web pages.

Activities Used:

Open Browser: The Open Browser activity is used to navigate to the desired web page (www.google.com) for text scraping. It provides the starting point for interacting with the web content.

Type Into: The Type Into activity is used to enter the search term "UiPath" into the search box of the web page. It simulates the typing action to perform a search.

Screen Scraping Wizard: The Screen Scraping Wizard is a powerful tool in UiPath Studio for extracting data from various applications and web pages. In this workflow, it is used with the Attach Browser activity to target the specific browser window. The Get Full Text option is selected to scrape the desired text content from the web page.

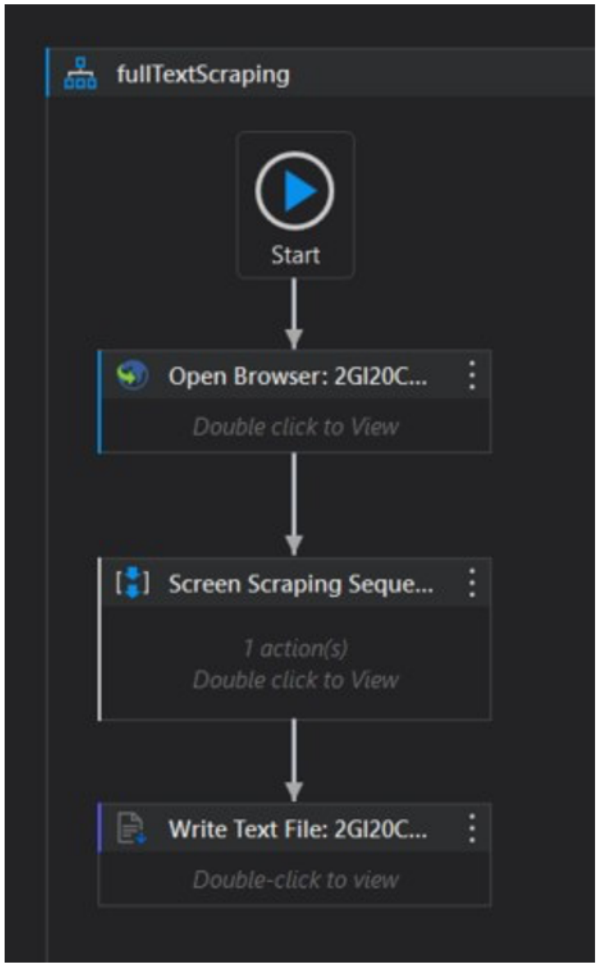
Write Text File: The Write Text File activity is used to write the scraped text content into a Notepad file. It specifies the output file path and content to be written.

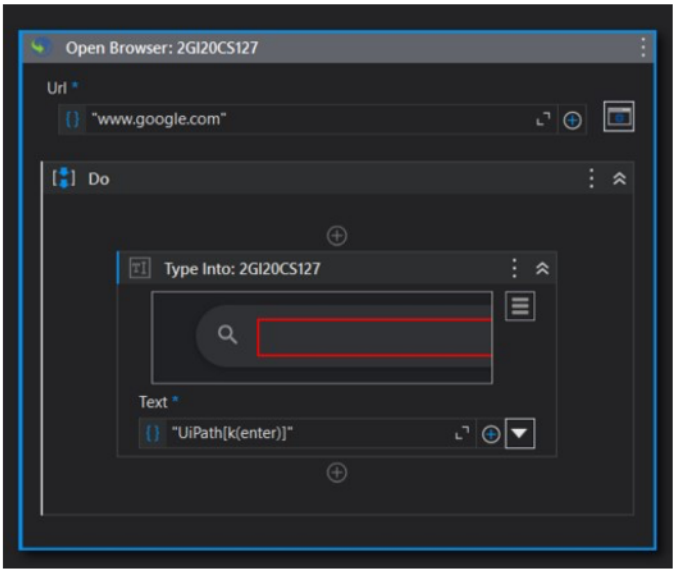
Packages:

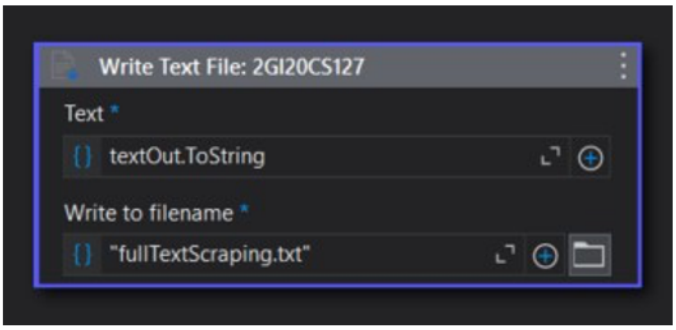
The workflow utilizes the following package:

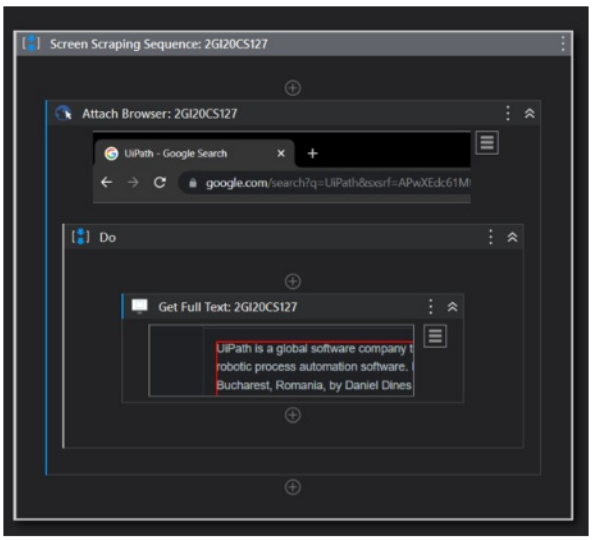
UiPath.UIAutomation.Activities: This package provides activities related to UI automation and screen scraping. The Screen Scraping Wizard is part of this package, allowing the extraction of text content from web pages.

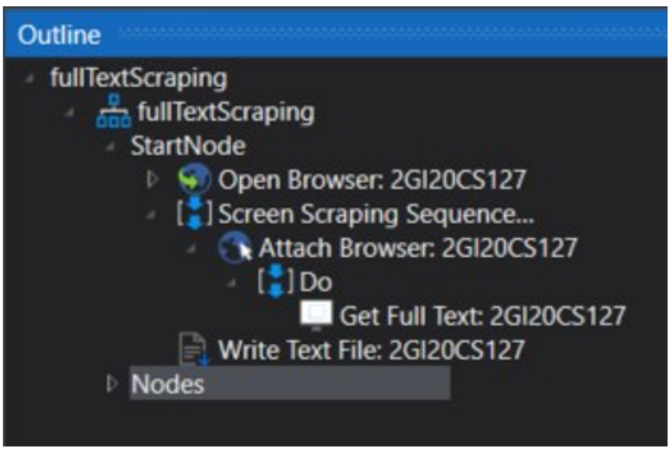
(8a) WORKFLOW AND OUTLINE:



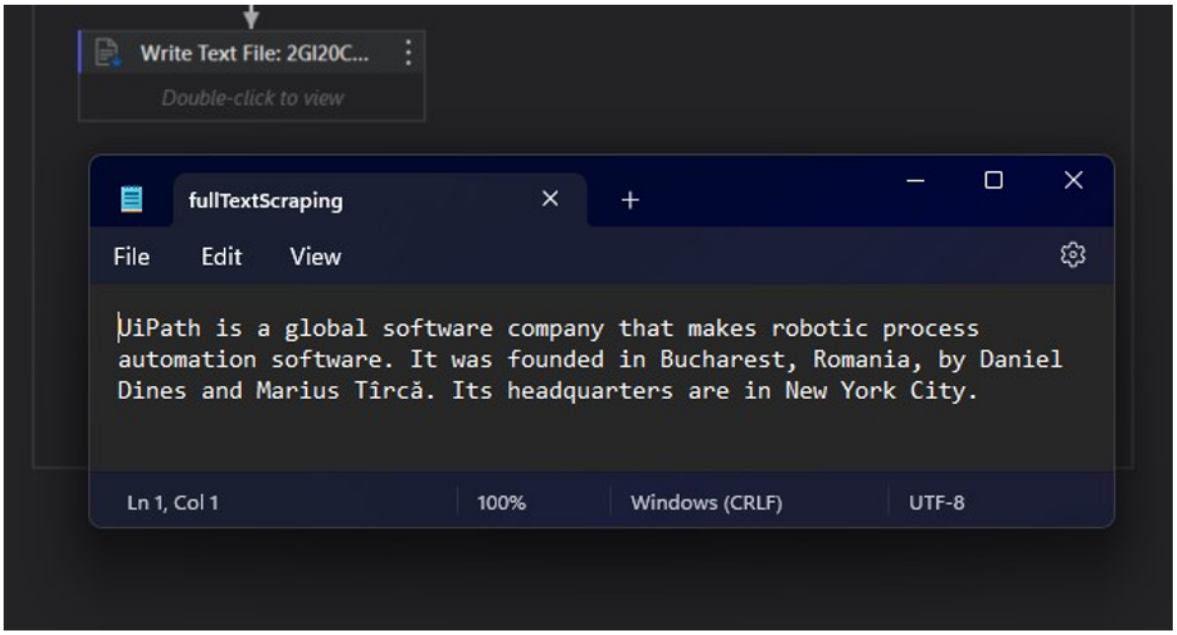








(8a) OUTPUT:


Write Text File: 2GI20C...
Double-click to view
fullTextScraping
File
Edit
View
UiPath is a global software company that makes robotic process automation software. It was founded in Bucharest, Romania, by Daniel Dines and Marius Tîrcă. Its headquarters are in New York City.
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

(8b) THEORY:

This termwork involves building a workflow in UiPath Studio that utilizes the Data Scraping Wizard to scrape blog post titles from the UiPath Blog across multiple pages. The automation will navigate to the UiPath Blog website, extract the structured data using the Data Scraping Wizard, and save the scraped information into an Excel file. This workflow showcases the power of UiPath Studio in automating the extraction of specific data elements from websites.

Activities Used:

Open Browser: The Open Browser activity is used to launch the web browser and navigate to the UiPath Blog website (www.uipath.com/blog). It serves as the starting point for interacting with the web content.

Message Box: The Message Box activity is used to display a notification message to indicate the start of the scraping process. It provides a visual confirmation to the user that the scraping will begin.

Data Scraping Wizard: The Data Scraping Wizard is a powerful tool in UiPath Studio that enables the extraction of structured data from web pages. In this workflow, it is used with the Attach Browser activity to target the specific browser window. The wizard assists in selecting the relevant data elements (blog post titles) and configuring the scraping parameters.

Excel Application Scope: The Excel Application Scope activity is used to open an Excel file ("blogData.xlsx"). It establishes the connection to the Excel application and allows subsequent activities to interact with the spreadsheet.

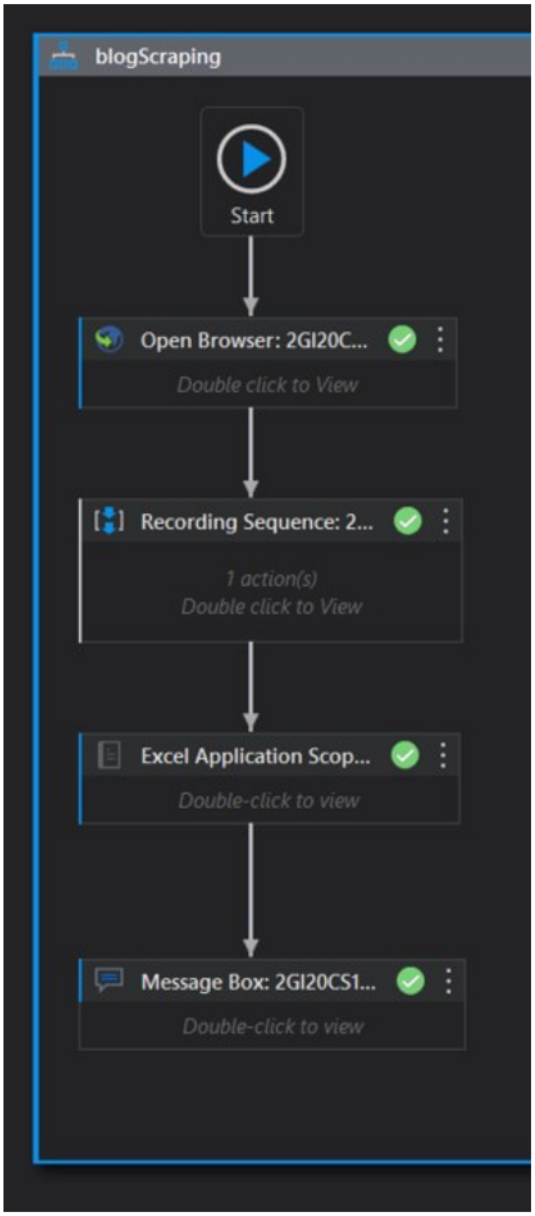
Write Range: The Write Range activity is used within the Excel Application Scope to write the scraped data into the Excel file. It specifies the target worksheet and the data table variable ("ExtractDataTable") that contains the scraped blog post titles.

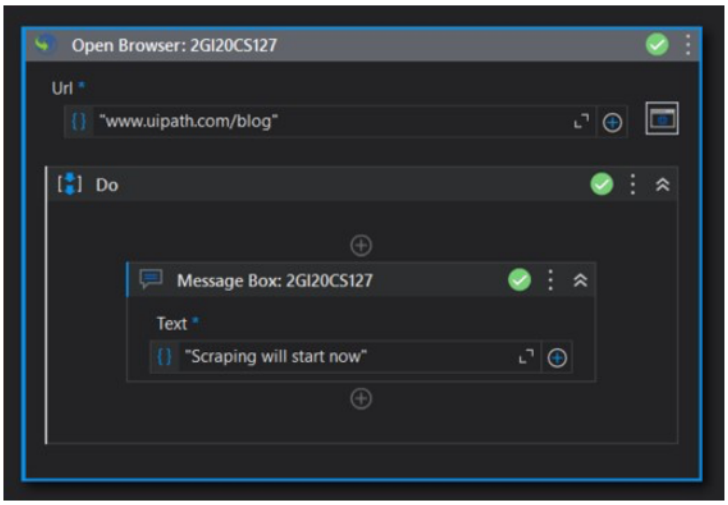
Packages:

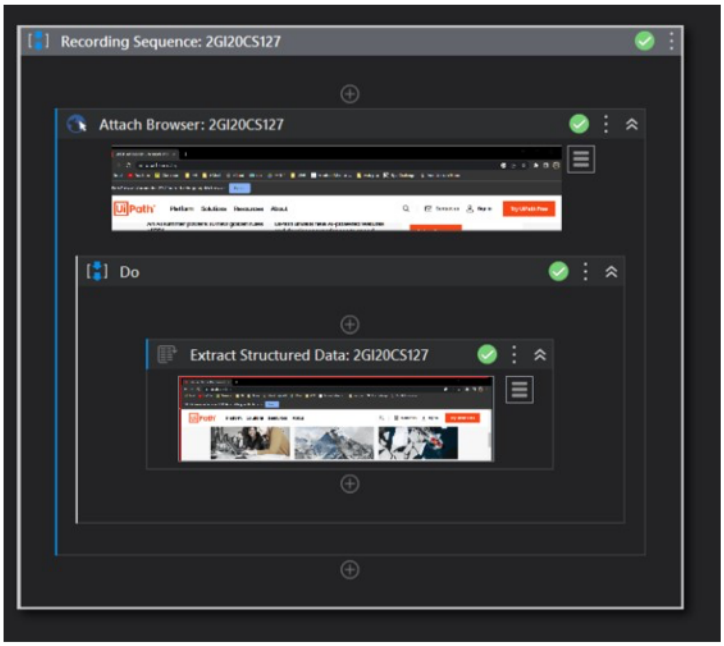
The workflow utilizes the following package:

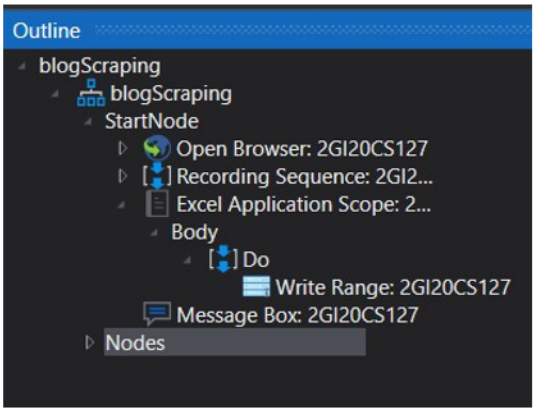
UiPath.Excel.Activities: This package provides activities related to Excel automation, allowing the interaction with Excel files, worksheets, and data. The Excel Application Scope and Write Range activities are part of this package.

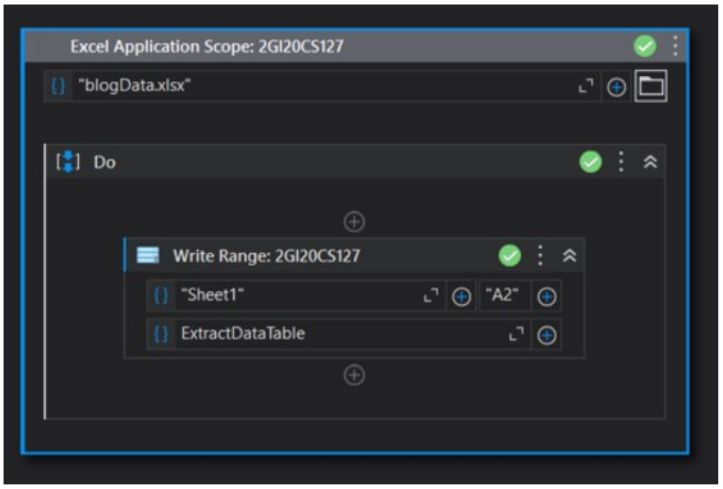
(8b) WORKFLOW AND OUTLINE:



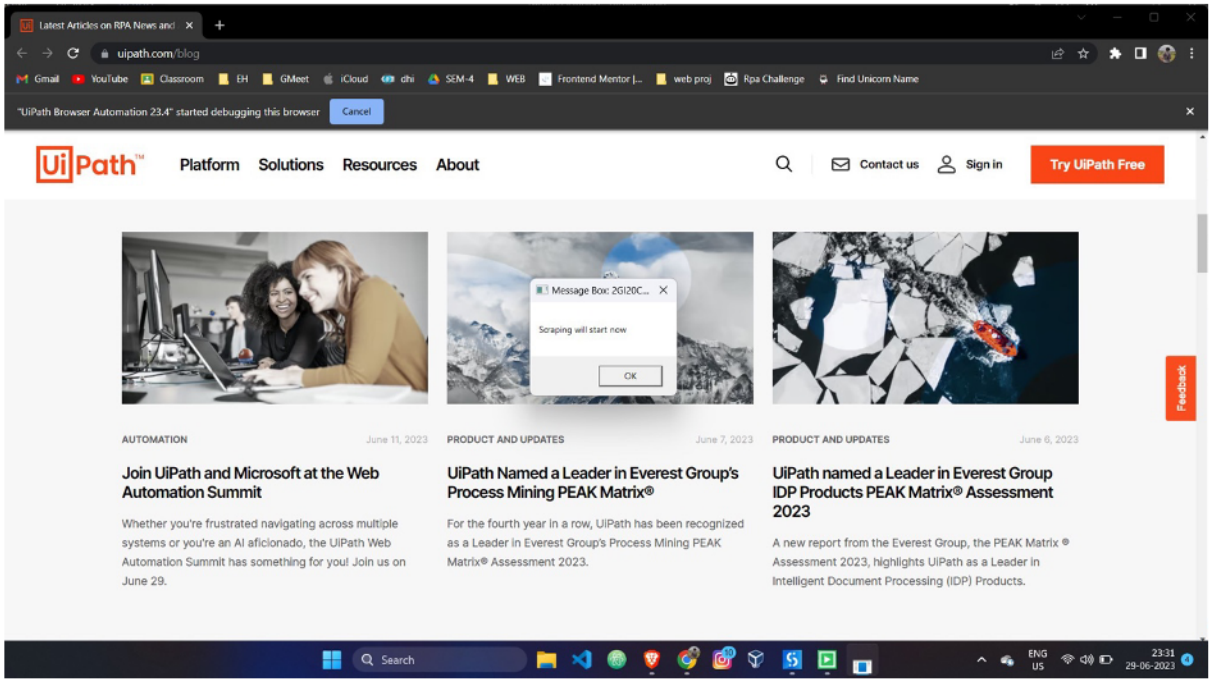


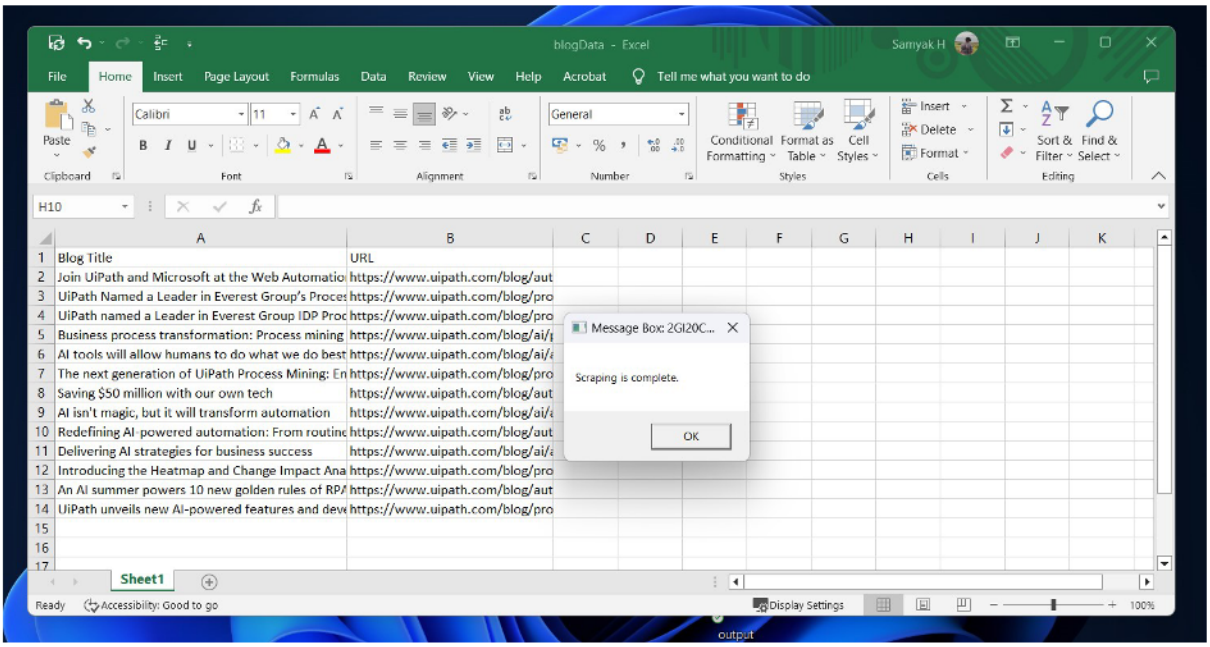






(8b) OUTPUT:


Latest Articles on RPA News and
uipath.com/blog
"UiPath Browser Automation 23.4" started debugging this browser
Cancel
UiPath
Platform
Solutions
Resources
About
Contact us
Sign in
Try UiPath Free
Message Box: 2GI20C...
Scraping will start now
OK
Feedback
AUTOMATION
June 11, 2023
Join UiPath and Microsoft at the Web Automation Summit
Whether you're frustrated navigating across multiple systems or you're an AI aficionado, the UiPath Web Automation Summit has something for you! Join us on June 29.
PRODUCT AND UPDATES
June 7, 2023
UiPath Named a Leader in Everest Group's Process Mining PEAK Matrix®
For the fourth year in a row, UiPath has been recognized as a Leader in Everest Group's Process Mining PEAK Matrix® Assessment 2023.
PRODUCT AND UPDATES
June 6, 2023
UiPath named a Leader in Everest Group IDP Products PEAK Matrix® Assessment 2023
A new report from the Everest Group, the PEAK Matrix ® Assessment 2023, highlights UiPath as a Leader in Intelligent Document Processing (IDP) Products.
Search


blogData - Excel
Samyak H
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Help
Acrobat
Tell me what you want to do
Paste
Calibri
11
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
Sort & Filter
Find & Select
Clipboard
Font
Alignment
Number
Styles
Cells
Editing
H10
Blog Title
URL
Join UiPath and Microsoft at the Web Automatio https://www.uipath.com/blog/aut
UiPath Named a Leader in Everest Group's Proces https://www.uipath.com/blog/pro
UiPath named a Leader in Everest Group IDP Proc https://www.uipath.com/blog/pro
Business process transformation: Process mining https://www.uipath.com/blog/ai/
AI tools will allow humans to do what we do best https://www.uipath.com/blog/ai/
The next generation of UiPath Process Mining: En https://www.uipath.com/blog/pro
Saving $50 million with our own tech https://www.uipath.com/blog/aut
AI isn't magic, but it will transform automation https://www.uipath.com/blog/ai/
Redefining AI-powered automation: From routine https://www.uipath.com/blog/aut
Delivering AI strategies for business success https://www.uipath.com/blog/ai/
Introducing the Heatmap and Change Impact Ana https://www.uipath.com/blog/pro
An AI summer powers 10 new golden rules of RPA https://www.uipath.com/blog/aut
UiPath unveils new AI-powered features and dev https://www.uipath.com/blog/pro
Message Box: 2GI20C...
Scraping is complete.
OK
Sheet1
Ready
Accessibility: Good to go
Display Settings
100%
output

CONCLUSION:

a) In conclusion, the termwork involving the use of the Screen Scraper Wizard to extract text using the Full-Text scraping method and storing it in a Notepad file demonstrates the power and versatility of UiPath Studio in web scraping tasks. By combining activities like Open Browser, Type Into, Screen Scraping Wizard, and Write Text File, the workflow successfully extracts the desired text content from a web page and saves it in a separate file.

b) In conclusion, the termwork involving the use of the Data Scraping Wizard to extract blog post titles from the UiPath Blog demonstrates the effectiveness of UiPath Studio in web scraping tasks. By combining activities such as Open Browser, Message Box, Data Scraping Wizard, Excel Application Scope, and Write Range, the workflow successfully navigates to the UiPath Blog website, extracts the desired data, and stores it in an Excel file.

The outcomes of this termwork are as follows:

Successful Text Extraction: The workflow effectively scraped text using the Full-Text scraping method. The Screen Scraper Wizard allowed for the identification and extraction of relevant text content from a specific source, in this case, a web page.

Accurate and Comprehensive Data Extraction: The Full-Text scraping method ensured that all text content, including paragraphs, headings, and other textual elements, was captured accurately. The extracted text was comprehensive and representative of the original source.

Streamlined Data Storage: The workflow stored the extracted text in a Notepad file, providing a simple and easily accessible format for further use or analysis. The data storage process was automated, eliminating the need for manual copying and pasting, thereby saving time and reducing errors.

Efficient Data Extraction: The Data Scraping Wizard successfully extracted blog post titles from the UiPath Blog. By configuring the wizard to identify and scrape specific elements, such as the post titles, the workflow automated the extraction process.

Seamless Pagination Handling: The workflow was designed to scrape blog post titles from multiple pages of the UiPath Blog. The Data Scraping Wizard efficiently handled pagination, automatically navigating through the pages and capturing the desired data.

Structured Data Organization: The scraped blog post titles were stored in a structured format, such as an Excel file or a datatable. This allowed for easy data management, sorting, filtering, and further analysis.

Scalable Data Extraction: The workflow can be easily adapted to scrape blog post titles from additional pages or even different websites. The flexibility of the Data Scraping Wizard enables users to scale the automation to meet their specific requirements.

Time and Resource Savings: By automating the data scraping process, the workflow significantly reduces the time and effort required for manual extraction. This frees up valuable resources and allows users to focus on more critical tasks.

TERMWORK 9

DATE: 31/05/2023

PROBLEM STATEMENT:

Build a workflow using Read PDF Text activity and extract only Email IDs and Phone Number from a PDF file and store in an MS Word file.

THEORY:

The following termwork involves building a workflow in UiPath Studio that utilizes the Read PDF Text activity to extract email IDs and phone numbers from a PDF file. The extracted data is then stored in an MS Word file. This automation showcases the capabilities of UiPath Studio in extracting specific information from PDF documents and manipulating MS Word files.

Activities Used:

Read PDF Text: The Read PDF Text activity is used to read the text content from the specified PDF file ("challenge.pdf"). It extracts the text data and stores it in the variable "extractedData" for further processing.

Message Box: The Message Box activity is employed to display the extracted text data ("extractedData") in a pop-up message box. This activity provides a way to verify the correctness of the extracted data during the workflow execution.

Word Application Scope: The Word Application Scope activity is utilized to create a Word document and perform operations on it. It specifies the MS Word file ("extractedData") to work with.

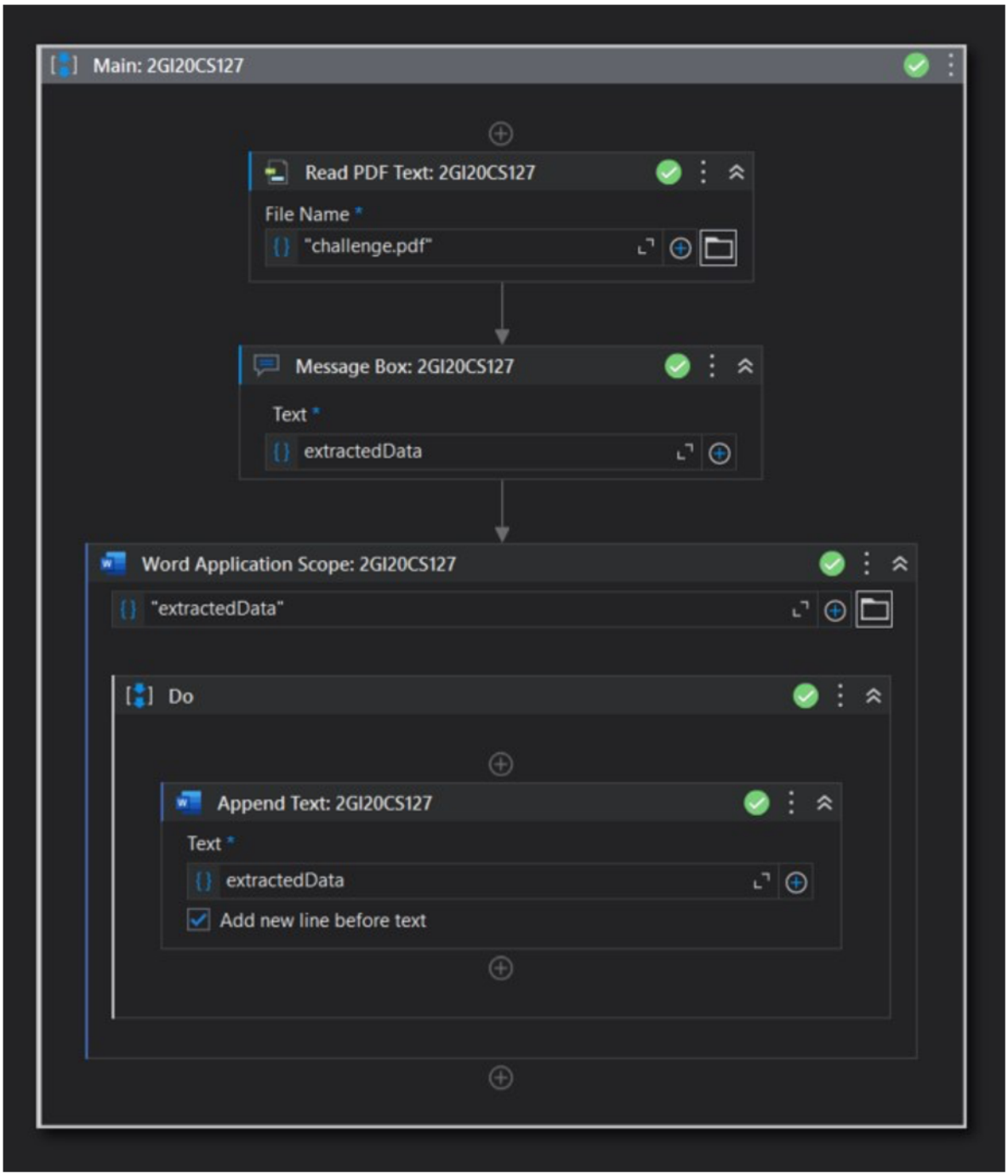
Append Text: The Append Text activity is used to add the extracted data ("extractedData") to the MS Word document. This activity ensures that the extracted email IDs and phone numbers are appended to the existing content of the Word file.

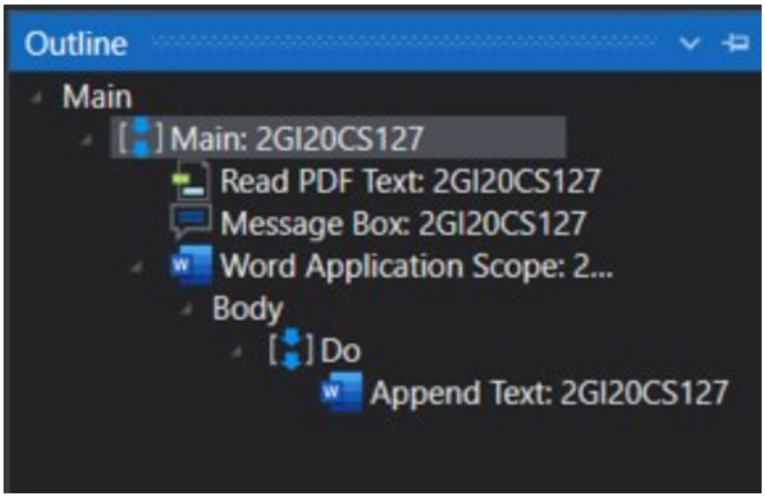
Packages:

UiPath.PDF.Activities: This package provides the Read PDF Text activity, which is essential for extracting text data from the PDF file.

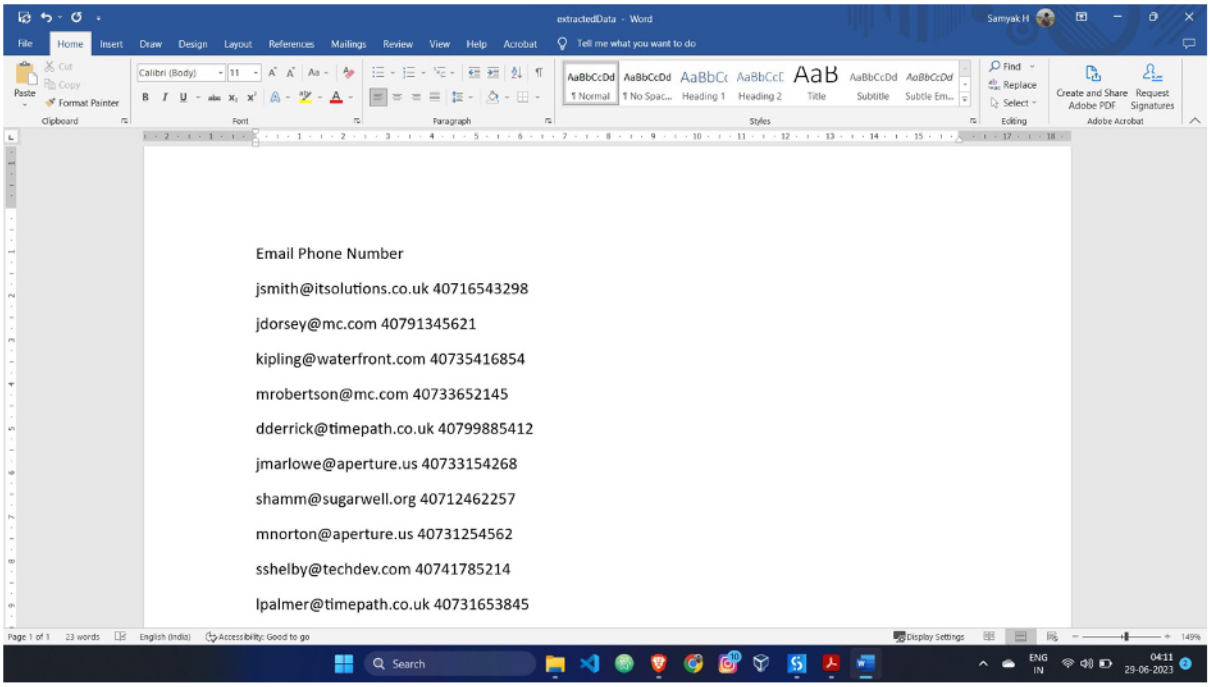
UiPath.Word.Activities: This package includes activities for working with MS Word files, such as Word Application Scope and Append Text. It allows the automation to interact with and manipulate Word documents.

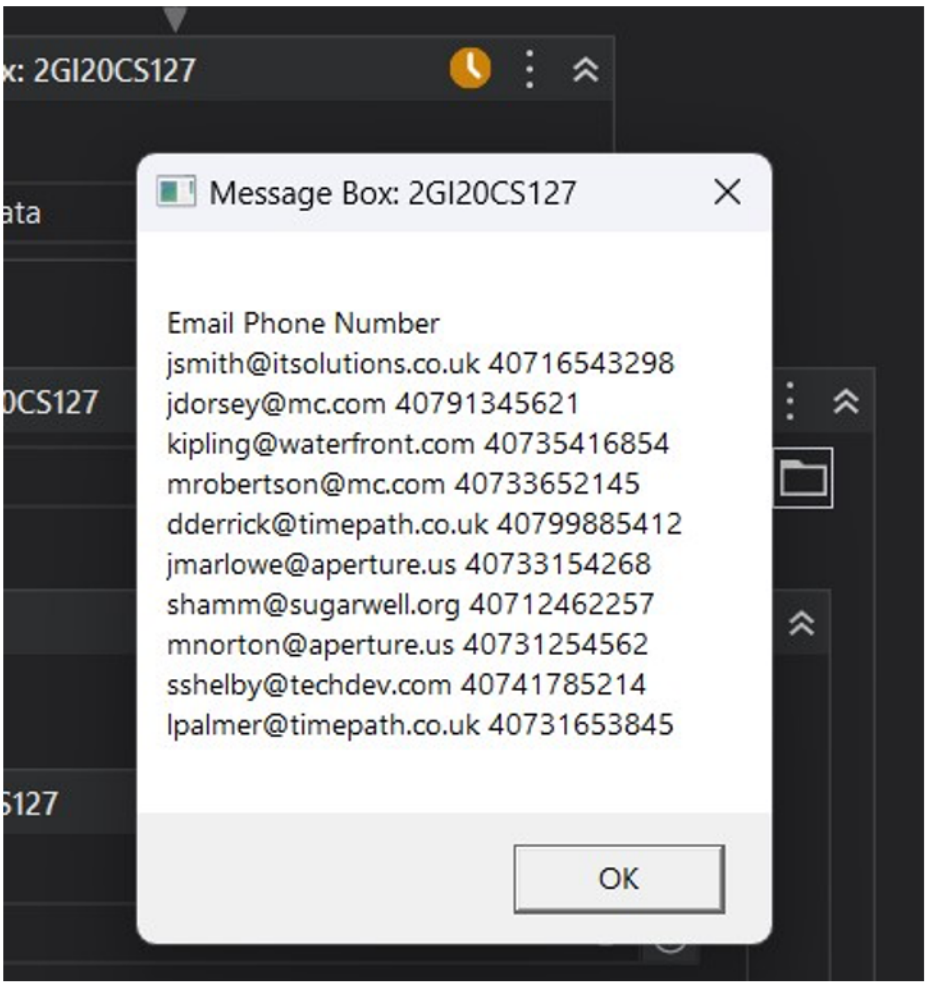
WORKFLOW AND OUTLINE:





OUTPUT:


extractedData - Word
Email Phone Number
jsmith@itsolutions.co.uk 40716543298
jdorsey@mc.com 40791345621
kipling@waterfront.com 40735416854
mrobertson@mc.com 40733652145
dderrick@timepath.co.uk 40799885412
jmarlowe@aperture.us 40733154268
shamm@sugarwell.org 40712462257
mnorton@aperture.us 40731254562
sshelby@techdev.com 40741785214
lpalmer@timepath.co.uk 40731653845


Message Box: 2GI20CS127
Email Phone Number
jsmith@itsolutions.co.uk 40716543298
jdorsey@mc.com 40791345621
kipling@waterfront.com 40735416854
mrobertson@mc.com 40733652145
dderrick@timepath.co.uk 40799885412
jmarlowe@aperture.us 40733154268
shamm@sugarwell.org 40712462257
mnorton@aperture.us 40731254562
sshelby@techdev.com 40741785214
lpalmer@timepath.co.uk 40731653845
OK

CONCLUSION:

In conclusion, the termwork involving the extraction of email IDs and phone numbers from a PDF file and storing them in an MS Word document demonstrates the capability of UiPath Studio in handling document processing tasks. By utilizing the Read PDF Text activity and Word activities, the workflow successfully extracts the desired information and saves it in a different file format.

The outcomes and benefits of this termwork are as follows:

Automated Data Extraction: The workflow eliminates the need for manual extraction of email IDs and phone numbers from the PDF file. By utilizing the Read PDF Text activity, the automation accurately extracts the desired information, reducing human error and saving time.

Data Accuracy: The automation ensures the accuracy of the extracted email IDs and phone numbers by directly retrieving the data from the PDF file. This outcome minimizes the risk of data entry mistakes that may occur during manual extraction.

Improved Data Management: By storing the extracted information in an MS Word document, the workflow enhances data management practices. The MS Word format provides a familiar and accessible format for storing and sharing the extracted data, facilitating easy retrieval and analysis.

Increased Efficiency: The automation streamlines the process of extracting email IDs and phone numbers from PDFs. It eliminates the need for manual searching and copying, allowing users to focus on more value-added tasks while the automation handles the data extraction process.

Reusability: The workflow can be easily modified and reused to extract similar information from different PDF files. By adjusting the PDF file input and adapting the extraction logic, the automation can be applied to various scenarios, increasing its versatility and utility.

TERMWORK 10

DATE: 14/06/2023

PROBLEM STATEMENT:

Build a workflow that extracts attachments from the emails containing the word "Resume" in its subject.

THEORY:

This termwork involves building a workflow that extracts attachments from emails with the subject containing the word "Resume". The workflow utilizes activities such as input dialog, get password, get IMAP mail message, for each, if condition, message box, and save attachments. The goal is to retrieve specific emails and save their attachments to a designated folder.

Activities Used:

Input Dialog: Prompts the user to enter their email username.

Get Password: Retrieves the password associated with the email.

Get IMAP Mail Message: Connects to the email server using IMAP protocol, retrieves emails from the "inbox" folder, and retrieves the top 10 emails.

For Each: Iterates over each email in the collection.

If Condition: Checks if the subject of the email contains the keyword "Resume".

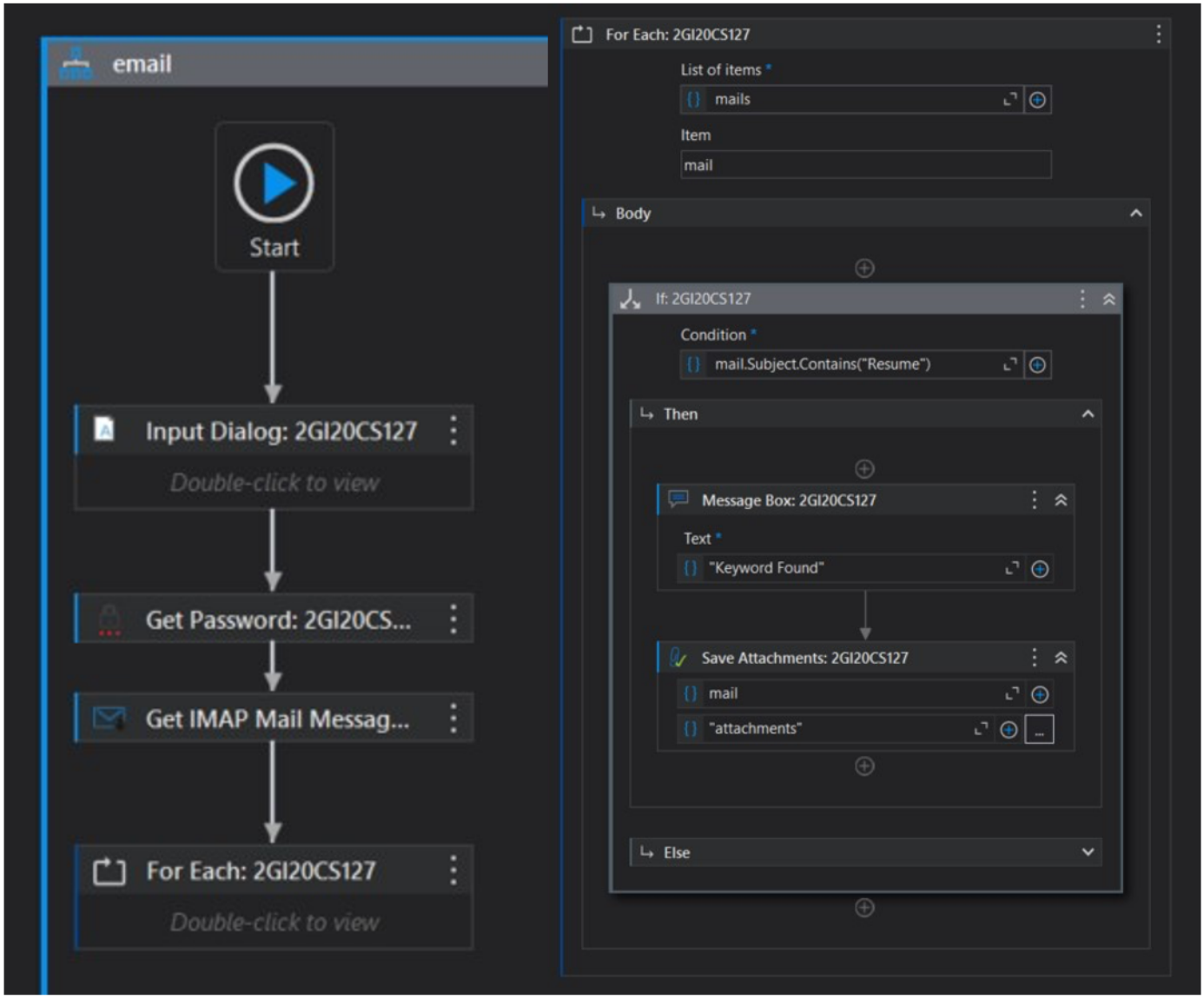
Message Box: Displays a message stating that the keyword was found in the subject of the email.

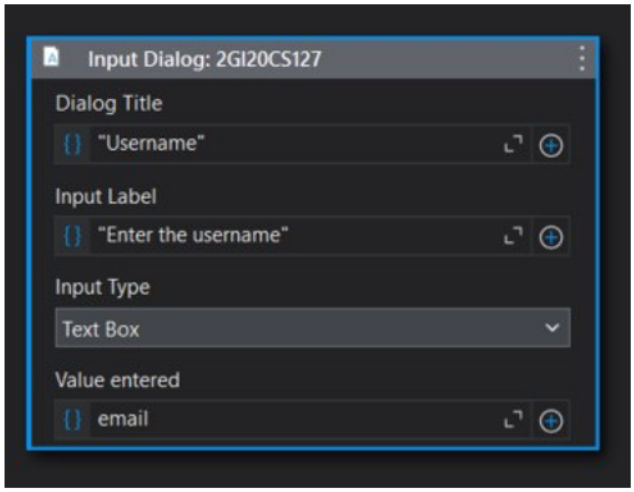
Save Attachments: Saves the attachments from the email to a designated folder.

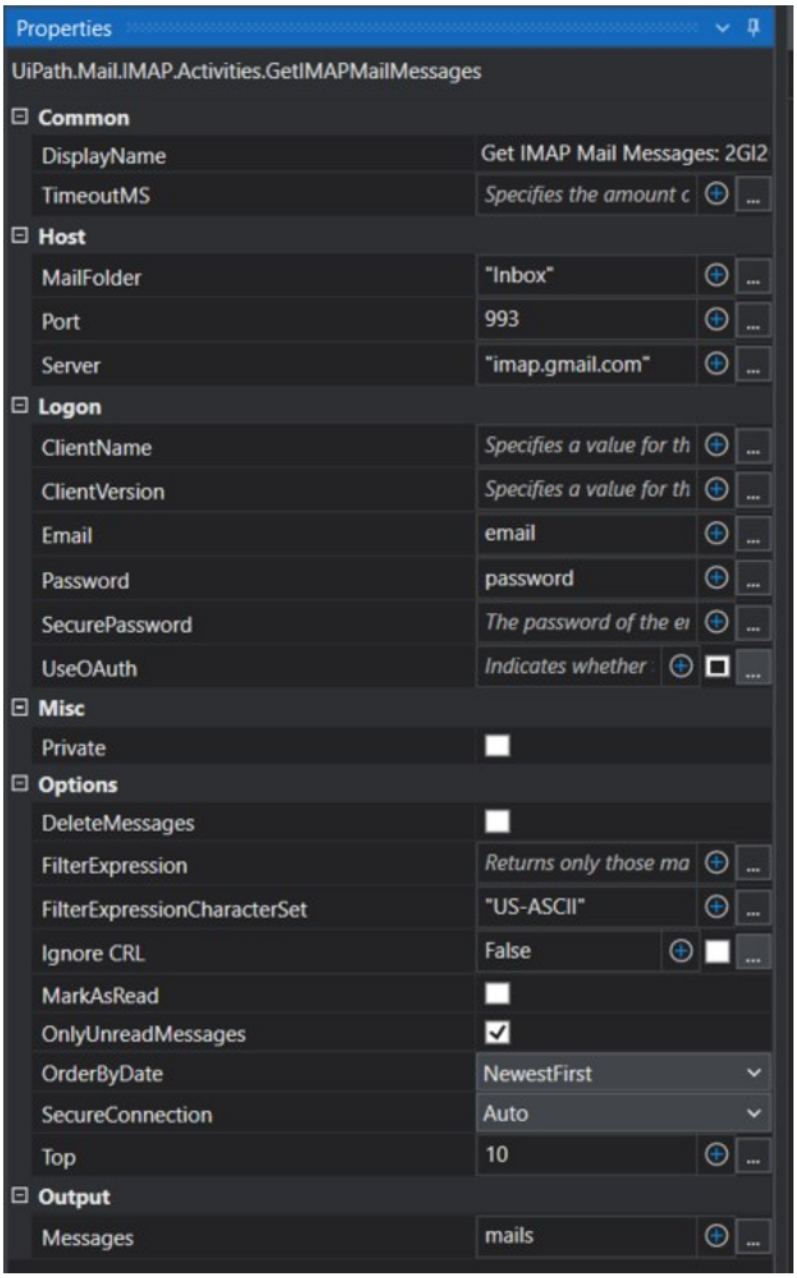
Packages Used:

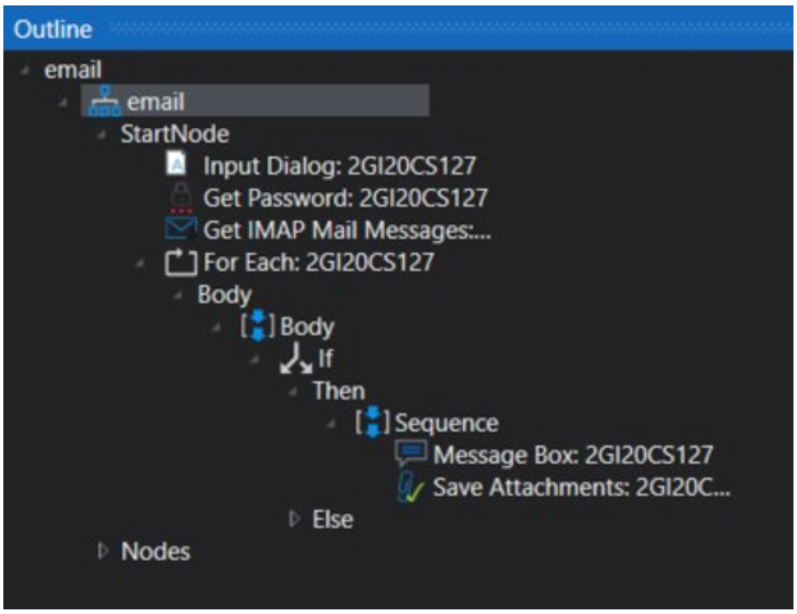
UiPath.Mail.Activities: Required for the "Get IMAP Mail Message" and "Save Attachments" activities.

WORKFLOW AND OUTLINE:

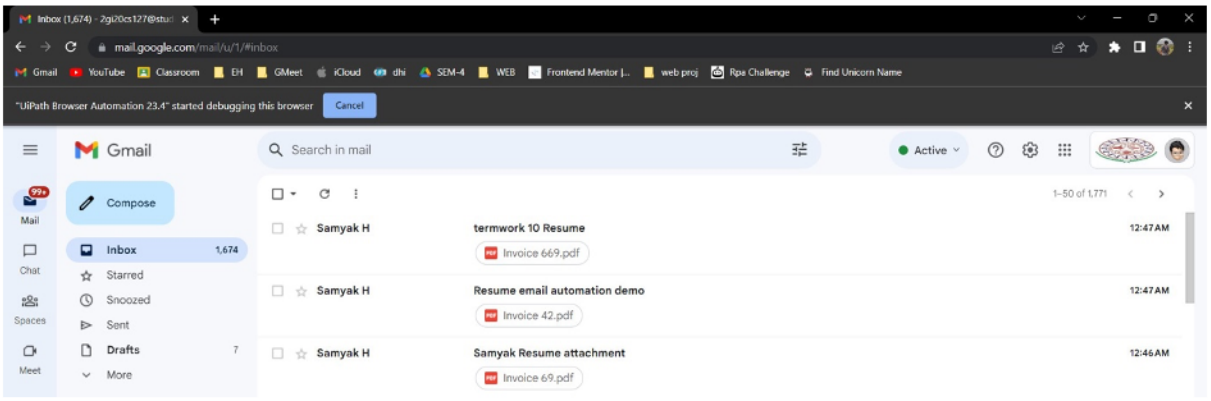


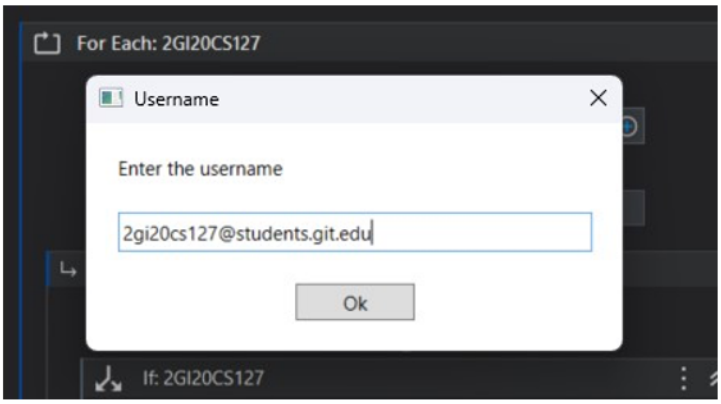


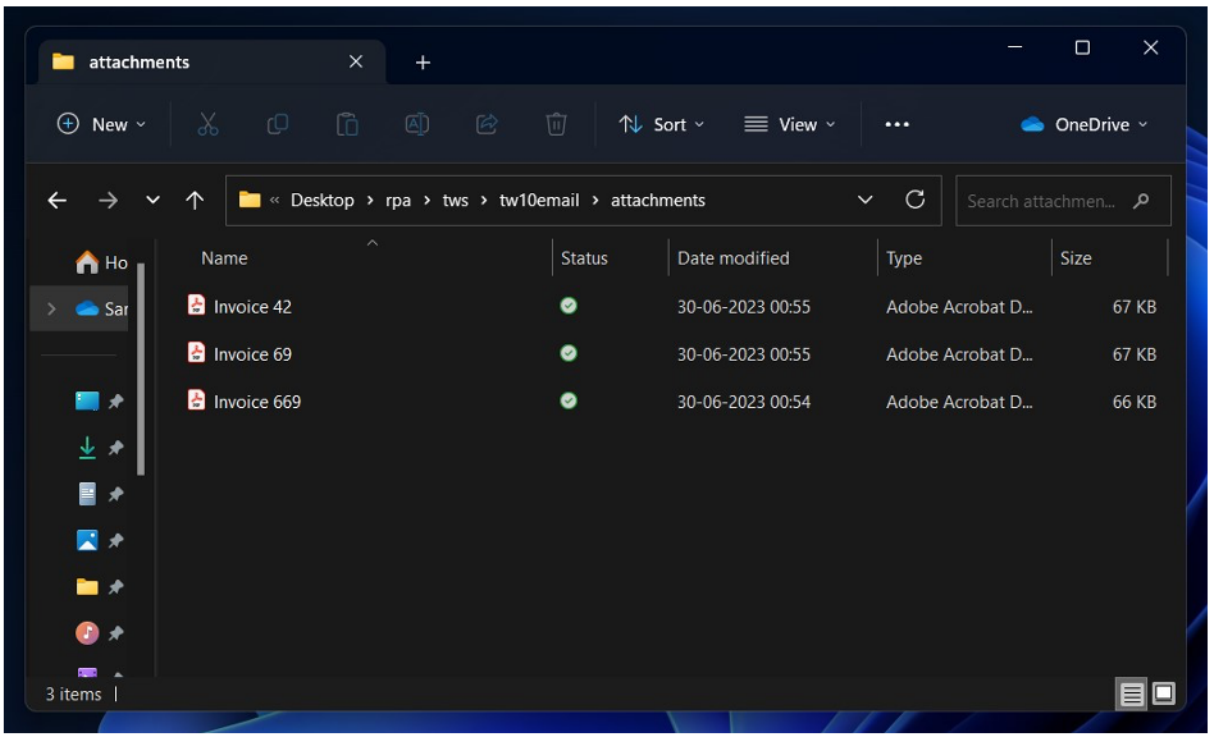

Properties
UiPath.Mail.IMAP.Activities.GetIMAPMailMessages
Common
DisplayName
Get IMAP Mail Messages: 2GI2
TimeoutMS
Specifies the amount c
Host
MailFolder
"Inbox"
Port
993
Server
"imap.gmail.com"
Logon
ClientName
Specifies a value for th
ClientVersion
Specifies a value for th
Email
email
Password
password
SecurePassword
The password of the e
UseOAuth
Indicates whether
Misc
Private
Options
DeleteMessages
FilterExpression
Returns only those ma
FilterExpressionCharacterSet
"US-ASCII"
Ignore CRL
False
MarkAsRead
OnlyUnreadMessages
OrderByDate
NewestFirst
SecureConnection
Auto
Top
10
Output
Messages
mails


Outline
email
email
StartNode
Input Dialog: 2GI20CS127
Get Password: 2GI20CS127
Get IMAP Mail Messages:...
For Each: 2GI20CS127
Body
Body
If
Then
Sequence
Message Box: 2GI20CS127
Save Attachments: 2GI20C...
Else
Nodes

OUTPUT:


Gmail
Search in mail
Compose
Inbox 1,674
Starred
Snoozed
Sent
Drafts
More
Samyak H
termwork 10 Resume
Invoice 669.pdf
12:47 AM
Samyak H
Resume email automation demo
Invoice 42.pdf
12:47 AM
Samyak H
Samyak Resume attachment
Invoice 69.pdf
12:46 AM


For Each: 2GI20CS127
Username
Enter the username
2gi20cs127@students.git.edu
Ok
If: 2GI20CS127


attachments
New
Sort
View
OneDrive
Desktop › rpa › tws › tw10email › attachments
Search attachmen...
Name
Status
Date modified
Type
Size
Invoice 42
30-06-2023 00:55
Adobe Acrobat D...
67 KB
Invoice 69
30-06-2023 00:55
Adobe Acrobat D...
67 KB
Invoice 669
30-06-2023 00:54
Adobe Acrobat D...
66 KB
3 items

CONCLUSION:

In conclusion, the termwork successfully demonstrated the automation of extracting email attachments based on specific criteria using UiPath. By implementing activities such as input dialogs, email retrieval, conditional statements, message boxes, and the save attachments function, the workflow effectively identified emails with the keyword "Resume" in their subject lines and saved the corresponding attachments to a designated folder. The practical application of this workflow showcased the ability to streamline and simplify the process of handling email attachments, ultimately saving time and effort. This termwork highlighted the potential of UiPath in automating email-related tasks and provided valuable insights into leveraging the power of robotic process automation for efficient and accurate attachment extraction.

The outcome of the termwork can be summarized as follows:

Automation of Attachment Extraction: The workflow successfully automates the process of extracting attachments from emails, eliminating the need for manual intervention.

Targeted Email Selection: The workflow filters emails based on the subject containing the word "Resume", ensuring that only relevant emails are processed.

Efficient Attachment Saving: The attachments from the selected emails are saved to a designated folder, allowing for organized storage and easy access.

Time-saving: By automating the attachment extraction process, the workflow saves time and effort that would otherwise be spent manually reviewing and saving attachments.

Error Reduction: Automation reduces the chances of human errors that may occur during manual extraction, ensuring accurate and reliable results.

4. REFERENCES:

- https://www.youtube.com/playlist?list=PL9ONIcWWS0hQ455pZKT8dPL56gwDKQkSj
- https://academy.uipath.com/
- https://docs.uipath.com/