🌟🌟🌟 Day 85 🌟🌟🌟
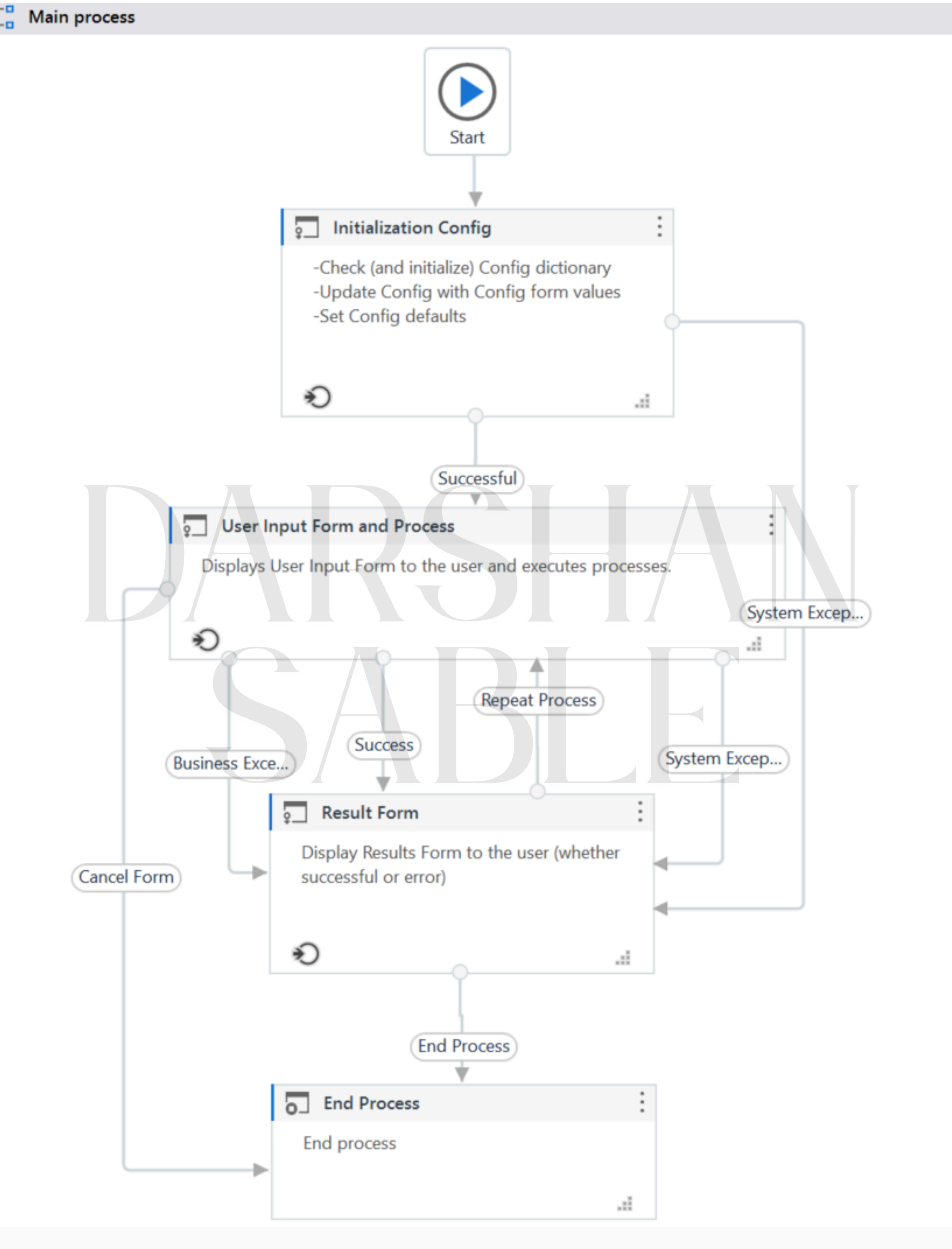
of Our Daily UiPath Q&A Challenge!

Welcome to Day 85 of our exciting UiPath Q&A Challenge! Each day, we'll be answering one key question to help you master UiPath and revolutionize your automation journey. 🚀

⛔ Question for Today

What is Attended Automation Framework?

It has 4 states and 8 transitions

Start

Initialization Config

-Check (and initialize) Config dictionary
-Update Config with Config form values
-Set Config defaults

Successful

User Input Form and Process

Displays User Input Form to the user and executes processes.

System Excep...

Repeat Process

Success

Business Exce...

System Excep...

Cancel Form

Result Form

Display Results Form to the user (whether successful or error)

End Process

End Process

End process

Overview

- The Attended Automation Framework is a comprehensive template tailored to attended scenarios, which aims to provide a seamless and user-friendly experience by offering a set of intuitive forms and corresponding trigger workflows.

How it works

The template consists of six distinct folders, each serving a particular purpose:

1. Framework_Workflows—Stores supporting workflows necessary for the template to execute correctly. This folder does not require modification.
2. Form_Inputs—Enables users to input information and interact with the automation through intuitive forms, and triggers workflows based on their actions (such as clicking Cancel, Continue, Select folder, or View user guide). The values are stored in the ConfigUtility.json file, which is used to pre-populate the forms in subsequent runs.
3. Form_ConfigUtility—Provides the ability to add custom logic and configurations, invoked by the ConfigUtility.xaml file and includes trigger workflows for canceling configuration changes, resetting all configurations to default, or saving new configurations. Allows you to customize the automation settings, including default form values, customized exceptions, additional settings, constants, and orchestrator assets, to tailor the automation to your specific needs. To set the configurations run this workflow at design time.
4. Form_Success—Displays the form when the transaction is successful and includes trigger workflows for options such as closing, repeating the process, or viewing the output.
5. Form_Error—Displays when an error or exception occurs during the automation process and includes trigger workflows for options such as closing the automation, emailing the error to the default email address (saved in the Form_ConfigUtility form file), or repeating the process.
6. Form_StatusWindow—Displays the automation process status throughout the automation, providing real-time updates on its progress.