Development Specifications Document (DSD)

Process Name: *Event Registration Bot*

Table of Contents

Version Control

Date	Version	Role	Name	Organization Department	Function	Comments
10 Jan 2023	1.0	Author	Muhammad Ahmed Naiem	Social	Send Invitation Email	Point of contact for everything in the project.

I. Document Overview

The Development Specifications Document (DSD) is created for every business process automated using RPA. The DSD needs to be reviewed and updated for every change requested and applied to the automated process. This document provides a technical snapshot and must always reflect the latest design and key features of the automated workflow.

The document naming convention will follow the naming convention and the version of the automated process. This can be “business process name version” or it can be defined, case by case, as part of the larger RPA project design.

This document is completed by the RPA Solution architect and RPA developer who automates the business process. It is reviewed by the business process owner, application owner, and CoE design authority.

This document is meant to assist the RPA COE, IT operations and process owners by providing a snapshot of the automated process details and components. It can also serve developers to have a quick glance at the setup, before diving into the code, to troubleshoot or update changes. The purpose of the document is to record the outcome specific to the automated master project and its subcomponents: projects, workflows, sequences etc.

II. Automated Master Project Details

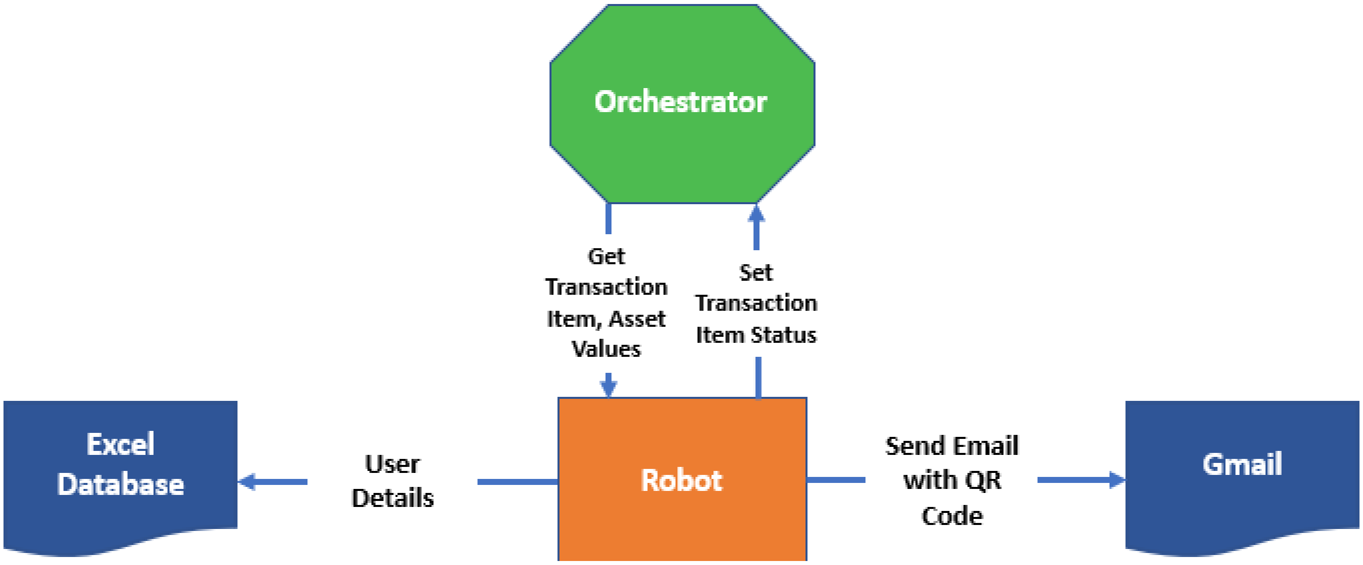
Details filled in by the developer reflect the actual information for the master project released for production.

#	**Item**	**Details** **Fill in with free text. If not applicable, mark the filed as "N/A". No empty fields.**
1	Master Project Name and Version	EventRegistrationBot V 1.0
2	Robot Type (attended/unattended/mix)	mix
3	Is Orchestrator used? (Yes/No)	Yes
4	Scalable? (Yes/No) Can the process be run by multiple robots in parallel?	Yes

III. Runtime Guide

1. Runtime Diagram

Architectural Structure of the Master Project Display the interaction between components (package / robots, Orchestrator queues, and running order).



2. List of Packages

Include **the list of packages and the high level description** for each of them, to explain each one's purpose:

#	Package Name	High-Level Description
1	EventRegistrationBot	• Initializes process settings by reading the config.xlsx file which exists in Data folder. • Gets the user details from Event registration queue transaction item. • Generates a custom ID form the user details. • Generates QR code image with user details along with the custom ID encoded within it. • Send invitation Email to the registered user with QR code image as an attachment. • Saves the user details in UsersDatabase.xlsx file in Data folder which is used as database.

*Add more rows to the table to include all the project names and versions. No fields should be left empty. Use “N/A” for the items that don't apply to your project.

3. Master Project Runtime Details

Details of the automated process:

#	Item	Details (Fill in with free text. If the section does not apply to your automation, mark the field as "N/A". No empty fields.)
1	**Production Environment Details**	Running on windows 10 machine
2	**Prerequisites to run**	• Config.xlsx file in data folder must be filled with the required and correct data. • A queue to hold the user data.
3	**Input Data**	Config.xlsx file Transaction Item holds user data
4	**Expected Output (output data)**	Invitation Email with QR code image are sent to user.
5	**How to start the automated process?**	From orchestrator or Uipath Assistant
6	**Resuming the process from a particular step**	It can only be restarted from the beginning
7	**Reporting** **queues reporting, Kibana or another platform**	Orchestrator logs Jobs dashboard For unexpected errors the Exception_Messages.txt file
8	**Manual Error Handling** **roll back or manually complete failed transactions. Procedures to reset the item. Ex "set status as investigating"**	Manually complete failed transactions
	A. How to resume the process in case of error	N/A
	B. How to manually fix transactions with error	if the QR code image is generated you can: • Scan the QR code image to retrieve user email and manually send the Email to him. • Add the user details to **UsersDetails.xlsx** file in data folder.
9	**Use of Orchestrator**	• Queue which holds process data. • Assets for credentials. • Process lunching

	A. Password Policies **specific compliance requests?**	Gmail Passwords or any smtp mail service, but make sure to change the .
	B. Stored Credentials **Never hard code credentials in the workflow**	Stored in orchestrator asset
	C. List of Asset Names	GmailCredential
	D. List of Queues Name	Event registration queue
	E. Schedule Details	N/A
10	**Recommended Resolution**	N/A

IV. Project Details

In this section describe all the projects that compose the automated process.

For each project, describe the workflow(s) in the logical order that they are called in.

If the workflow is a flowchart, also include the exported image from Studio.

If the automated process is composed of multiple projects, copy paste and fill in the table below for each project with its specific details (there are 2 here already, assuming a dispatcher and performer project)

1. Project Name: *Event Registration Bot*

General information about the process selected for RPA prior to automation.

#	**Item Name**	**Details** **Fill in with free text. If not applicable, mark the field as “N/A". No empty fields.**
1	**Environment used for development** **name, location, configuration details etc**	Studio 2022.12.0-beta.11471 - 12/20/2022 Community License Per-user Installation License Provider: Orchestrator
2	**Environment prerequisites** **OS details, libraries, required apps**	Microsoft Windows 10 Pro 64-bit 6.0.7
3	**Logging level**	Info

4	**Details about automation** **if the apps were automated using UI Automation, Image & Text**	- There is no UI automation involved in this process. - All data table actions use workbook activities to reduce error rate when trying to open excel file.
5	**In case of attended bot, can the user operate the computer while the robot is running?**	Yes
6	**Repository for project** **where the developed project is stored**	N/A
7	**List of reused components**	**InitAllSettings** workflow from REF template are reused in this process under the name of **SettingInitialization** workflow with some modification to suit the process needs.
8	**Custom logs defined in the workflows** **where Throw Activity was used or custom log message was defined**	- Asset Failed To Load Exception – happens when process can't get asset from orchestrator. - initialization Exception – happens when there is a problem during setting initialization. - Stop process requested – happens when a stop signal is sent from orchestrator to stop the process. - No transaction item – happens when the process cannot retrieve transaction item or there is no new items. - User custom ID – shows generated users' custom ID. - AuthenticationException – happens when there is an exception with email account credentials.
9	**Frequent errors found in the development phase**	initialization Exception happens sometimes due to nonvalid data in config file.
10	**Workarounds used in the automation phase**	in **SaveUserDetails.xaml** workflow I found no way to append new row to **UsersDatabase.xlsx** file using the workbook activities(i.e., without opening the excel), so I built a new data table then used **add data row** activity to assign the user details row to the new table, then I appended the whole new table range (which contains only the user details row) to **UsersDatabase.xlsx** using **Append Range Workbook** activity.
11	**Configuration method** **assets, excel file, Json file**	**Config.xlsx** file in **Data** folder

12	**Configuration details** **path for input files, configuration Orchestrator assets used**	Use config file to: • Add queues and assets names and orchestrator folder where they exist. • Add some parameters related to: o SMTP mail server and port. o Where to save QR code image. o QR code image dimensions. o Email body Invitation message. o Retry number in case of failing in getting transaction item or setting transaction status.
13	**Workflow File Export List** **Use the project mapping tool**	N/A

1.1 Workflow(s) specific to the Project

Define below all the workflow files (.xaml files) used in the project, with the Input and Output data.

#	Workflow File Name	Description	Arguments	Comments
1	SettingInitialization	This workflow will initialize, populates and outputs a configuration dictionary, Config, to be used throughout the project.	out_Config – Dictionary<String, object> in_ConfigFile – String in_ConfigSheets – String[]	N/A
2	GetUserDetails	This workflow will fetch the user details from the orchestrator queue.	in_Config – Dictionary<String, object> out_UserDetails – QueueItem	N/A
3	GenerateUserCustomID	This workflow will Generate a custom ID for the user to be stored along with his data and act as unique number for him.	in_UserDetails – QueueItem out_UserCustomID - String	N/A

4	GenerateBarcode	This workflow will Generate a barcode containing user details and his unique custom ID.	in_Config – Dictionary<String, object> in_UserDetails – QueueItem in_UserCustomID - String	N/A
5	SendInvitationEmail	This workflow will Send invitation Email along with the QR code Image to the Email from user details.	in_Config – Dictionary<String, object> in_UserDetails – QueueItem	N/A
6	SaveUserDetails	This workflow will Save the user data to the UsersDatabase.xlsx file.	in_Config – Dictionary<String, object> in_UserDetails – QueueItem in_UserCustomID – String in_UsersDetailsSheet – String in_UsersDatabaseFile - String	N/A

V. Compliance Considerations and Reporting Requirements

- N/A

VI. Other Details

1. Future Improvements

- trigger the robot to start if new item was added to the queue.
- save the user data in an online database system.
- Provide a backup email to report for errors in case of error occurred during setting initialization which the original email won't be retrieved.
- Rebuild the process using the RE Framework.

2. Debugging Tips

- At the end of every execution you will find a log message with warn level indicating that there is no more transaction items, that's because when the process starts it begins to process all work items one by one until no work items is left, so in last iteration it will generate this warn level log message.

3. Other Remarks

- **Important**: When editing the config.xlsx file make sure to only change the value field, not the name filed or many errors will arise. In case of asset, only change the values of **Asset** and **OrchestratorAssetFolder** fields.

VII. Post UAT Specifications

- Average duration per transaction (varies depending on the Test environment): 2 – 5 Seconds.
- Recommended number of robots for the specified volumes: only 1 robot will process big amounts.
- Specified schedule: N/A

VIII. Glossary

- **Master project** - the overall output of the development, containing one or multiple projects that together cover the scope of the robotic process automation.
- **Project** - a UiPath Studio project containing one or multiple workflow files. A project can be converted to a package and run independently, covering a particular scope within the master project. The project is used when defining the development and support phase of the automation.
- **Package** - the output of compiling a project. A package can be deployed on the robot machine and be executed by the robot service. Only one package can be executed at a given time by a robot. The package is used when defining the running phase of the automation
- **Workflow** - a component of the package, the workflow encapsulates a part of the project logic. The workflow can be of type: sequence, flowchart or state machine. a workflow is saved as an .xaml file inside the project folder. A workflow file can be invoked from another workflow and by default there is an initial workflow file that will run when executing the package.
- **Activity** - an action that the robot executes.
- **Sequence** - a workflow where activities are executed one after another, in a sequential order
- **Flowchart** - a workflow where activities are connected by arrows and the logic of the workflow can be easily followed in a visual manner. The flowchart can also be exported as an image from UiPath studio
- **State machine** - a more advanced way of organizing a workflow, similar to a flowchart.
- **BOR** - Back office robot
- **FOR** – Front office robot
- **Orchestrator** – Enterprise architecture server platform supporting: release management, centralized logging, reporting, auditing and monitoring tools, remote control, centralized scheduling, queue/robot workload management, assets management.