

# **UiPath Automation**

## **Process Design Document**

**Project Name : Invoices Generator**

# Process Design Document History

Date	Version	Role	Name	Organization
01.07.2023	1.0	Draft	Animesh	uipath.

# Table of Contents

1. Introduction.....	
1.1 Purpose of the document.....	
1.2 Objectives.....	
1.3 Process key contacts.....	
2. AS IS Process Description.....	
2.1 Process overview.....	
2.2 Detailed Process map.....	
2.3 Detailed Process Steps.....	
2.4 Exceptions handling.....	
2.5 Error mapping and handling.....	
2.6 In-Scope application details.....	
3. Development details.....	
3.1 Prerequisites for development.....	
3.2 Password policies.....	
3.3 Credentials and asset management.....	

# 1. Introduction

## 1.1 Purpose of the document.

The Process Design Document describes the business processes chosen for automation using the UiPath Robotic Process Automation (RPA) technology. This document describes the sequence of steps performed as part of the process, as well as the conditions and requirements prior to its automation. This design document serves as a base documentation for developers to collect the details required for robotic automation of the same business process

## 1.2 Objectives

The process has been selected for RPA as part of the larger project initiative, the Finance and Accounting department. The objective of this process automation is linked to the project business case and is mainly intended to

- ◆ Deliver faster processing
- ◆ Reduce redundant activities
- ◆ Improve overall performance and reliability

## 1.3 Process key contact

The Design Document includes a brief, but comprehensive set of requirements for the process. Its structure is based on the input provided by the Subject Matter Expert (SME) in the process.

Role	Name	Date of action	Notes
Process SME	Animesh	TBD	Point of contact for questions related to business exceptions and passwords
Reviewer / Owner		TBD	POC for process exceptions
Approval for production		TBD	Escalations, Delays

## 2. AS IS Process Description

### 2.1 Process overview

General information about the process selected for RPA implementation, prior to its automation:

AS Is process details	
Process full name	Invoice generator
Function	Cost-cutting, payment validation
Department	Finance and Accounting
process short description (operation, activity, outcome)	Process creates invoices with the help of data stored locally or remotely and update the respective recipient
Role required for performing the process	System1 user
Process schedule	Alternate days/daily
# of item processes / day	7 – 15 Clients
Average handling time per item	Average handling time per item
Peak period (s)	No peak period
# of FTEs supporting this activity	1
Level of exception rate	no expected exceptions
Input data	Client Data
Output data	Invoice at mail or locally

### **2.1.1 In scope for RPA**

The activities and exceptions in this process that are in the scope for RPA, are listed below:

- ❖ Full Scope for RPA - the process is to be 100% automated.

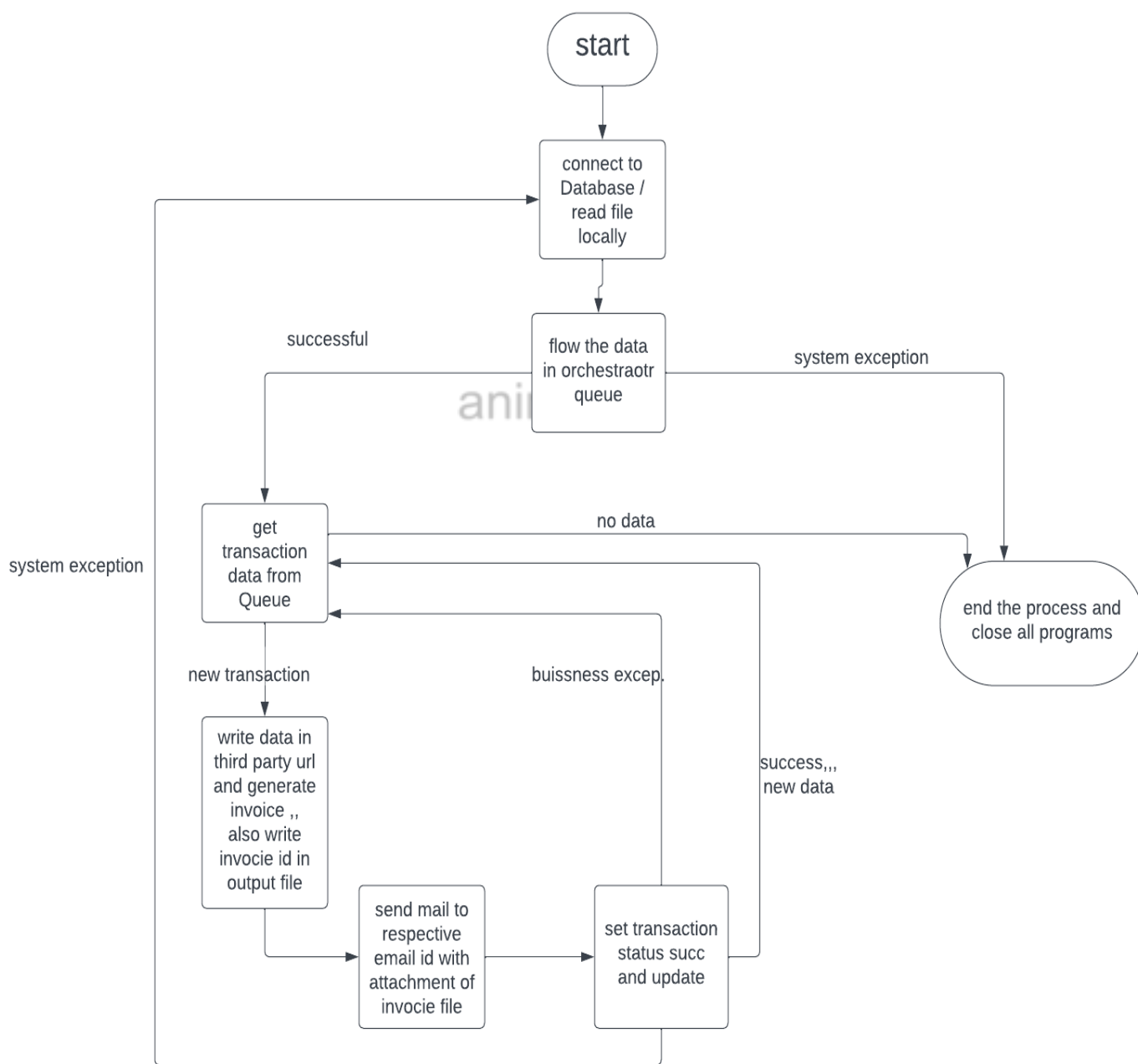
### **2.1.2 Out of scope for RPA**

There are no activities out of scope for RPA

animesh

## 2.2 Detailed Process map

This chapter presents the chosen process in detail, which enables the developer to build the automated process:



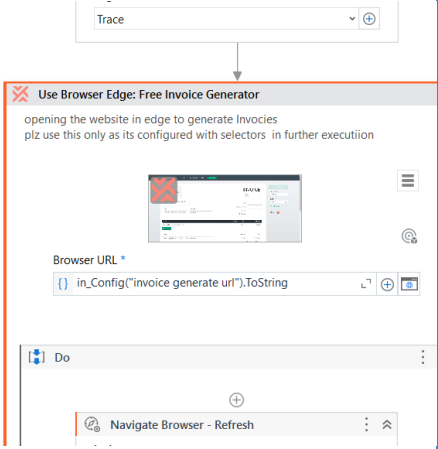
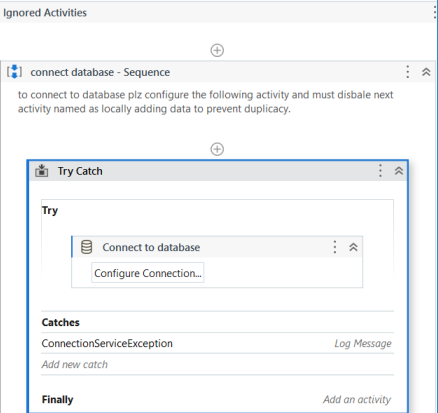
Step	Short Description
1.1	Open the Microsoft Edge browser with the URL of the invoice generator
1.2	Retrieve email attachment as data or get it from the local device
1.3	Update the data in Orchestrator queue
1.4	Optionally, use database connection to upload the data to the queue
1.5	Retrieve each item as a transaction item from the queue
1.6	In the process
1.7	Perform the invoice download procedure

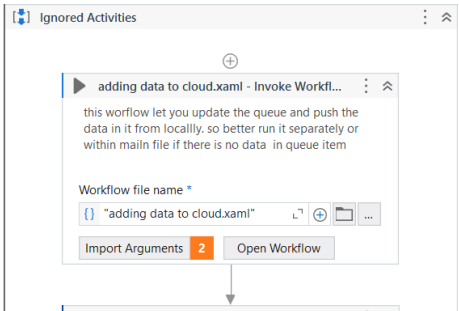
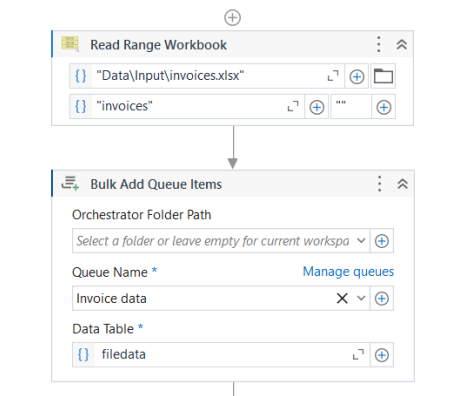
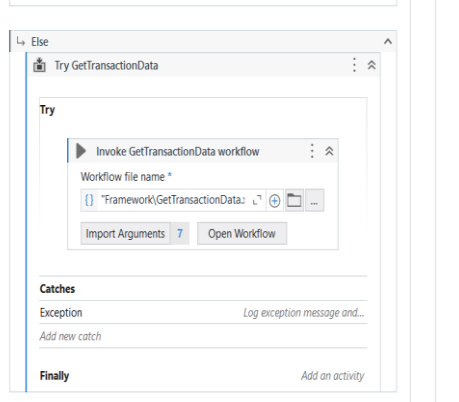


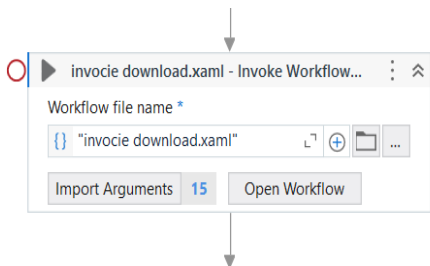
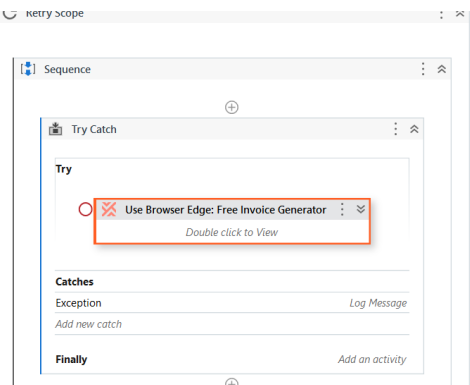
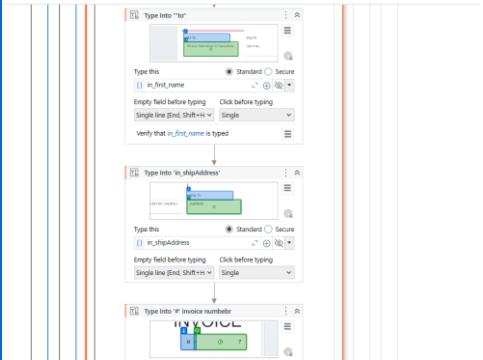
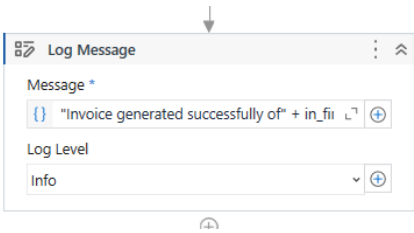
Step	Short Description
1.7.a	Use the browser to access the invoice generator website
1.7.b	Perform required operations on each detail of the item on the website
1.7.c	Configure the selectors to handle exceptions
1.7.d	Download the generated invoice
1.8	Write the invoice ID into the respective column of the output worksheet
1.9	Send an email to the respective email ID with the invoice attachment
2.	Continue with the next item in the queue

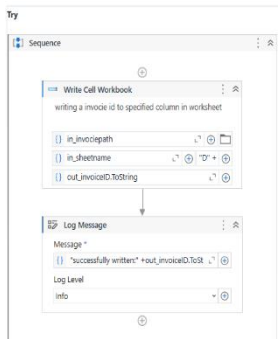
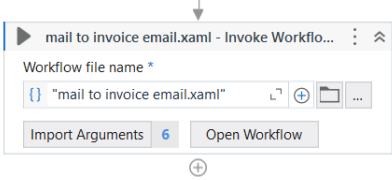
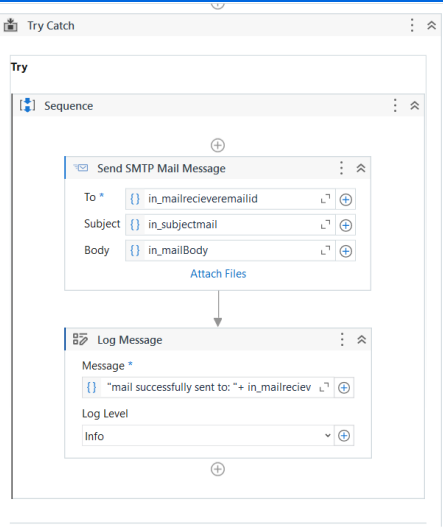
### 3. Detailed Process Steps

The complete set of steps performed in the process are given below with handled all kind of exceptions.

#	Step action description	Screenshot	Expected result	Remarks
1.1	opening the website in edge to generate Invoices		Browser opened.	Possible exception: - Handle exception if browser failed to open
1.2	Optionally can use database server to retrieve the data		connection successful	Possible exception: - Handle exception if Incorrect server root and password

1.3	Invoke workflow			
1.4	<p>To read the data from local file</p> <p>➔ Update the data to orchestrator Queue</p>		<p>File read success</p> <p>And</p> <p>Queue item updated !</p>	<p>Possible exceptions:</p> <p>File doesn't exist- Throw exception</p> <p>Queue doesn't exist Throw exception</p>
1.4.a	Retrieve the item from queue		In transaction item updated success and move to now process state	<p>Possible handled exception:</p> <p>by framework itself</p>

1.5	Invoke workflow in Process state Now			
1.5.a	Using use browser to type into the data in website and configure all selectors with accuracy more than 80-90%		Should be able to complete with no errors and download invoice in specific path assigned	Possible exception handled:  Browser failed to launch
1.5.b	All sub activities to be executed on website to fill the data		Continue till the end of this and invoice download	Possible handled exception:  if sub activity selector failed to configure
1.5.c	Log the message:  Invoice generated successfully of {name} ,ID is { invoice id}		Log success	NA

1.5.d	writing an invoice id to respective recipient column in worksheet		Cell in sheet should be filled with invoice ID	Possible handled Exception:  If file doesn't exist handled with throw and move to next item
1.6	Invoke workflow to send smtp mail		NA	NA
1.6.a	Using activity send smtp mail message to send mail to respective recipient mail id with the attachment as invoice downloaded.  Ex. Invoice #23021.pdf		Mail should be sent successfully to receiver email id with the attachment file  And log message Email successfully send.....	Possible exception handled :  Email id not valid /no-exist And throw business exception And proceed another item in a queue.
1.7	Proceed with next Item in Queue.			

## 2.4 Exceptions handling

The types of exceptions identifiable in the automation process can be classified according to the table below.

Area	Known	Unknown
<b>Business</b>	Previously encountered situation. A possible scenario is defined, and clear actions and workarounds are provided for each case.	A situation never encountered before. It can be caused by external factors.

#	Exception Name	Step where exception is encountered	Parameters	Action to be taken
1.	Browser failed to open	1.1	If any unexpected error in system	Throw System exception Clean all the states and try again
2.	Database connection failed	1.2	Incorrect credentials	Incorrect credentials Stop process

3.	Filenotfound Throw exception  Queue doesn't exist Throw exception	1.4	File doesn't exist and not queue too	stop the process ,critical error
4.	Browsernotopened exception	1.5.a	System problems	Clean the state and retry
5.	filenotfoundexception	1.5.d	Incorrect file path	Ignore this and log message , Continue on next transaction item
6	Invalid email exception	1.6	Invalid email id	Send report to abc@gmail And update the status scenario in excel file .

For any other unanticipated or unknown exceptions, the robot send an email notification at exceptions@abc.com with the original email and error message screenshot attached.

## **3. Development details**

### **3.1 Prerequisites for development**

Development or testing environment are to be provided for development purposes.

~ The provided development and testing environments are exact replicas of the production environment

. ~ Dedicated system and application access are given to developers with the adequate permissions.

### **3.2 Password policies**

Users manage their own passwords. There are no special policies in place.

animesh

### **3.3 Credentials and asset management**

Login details (user IDs and passwords) should be stored under Windows Credential Manager or UiPath Orchestrator Assets



animesh