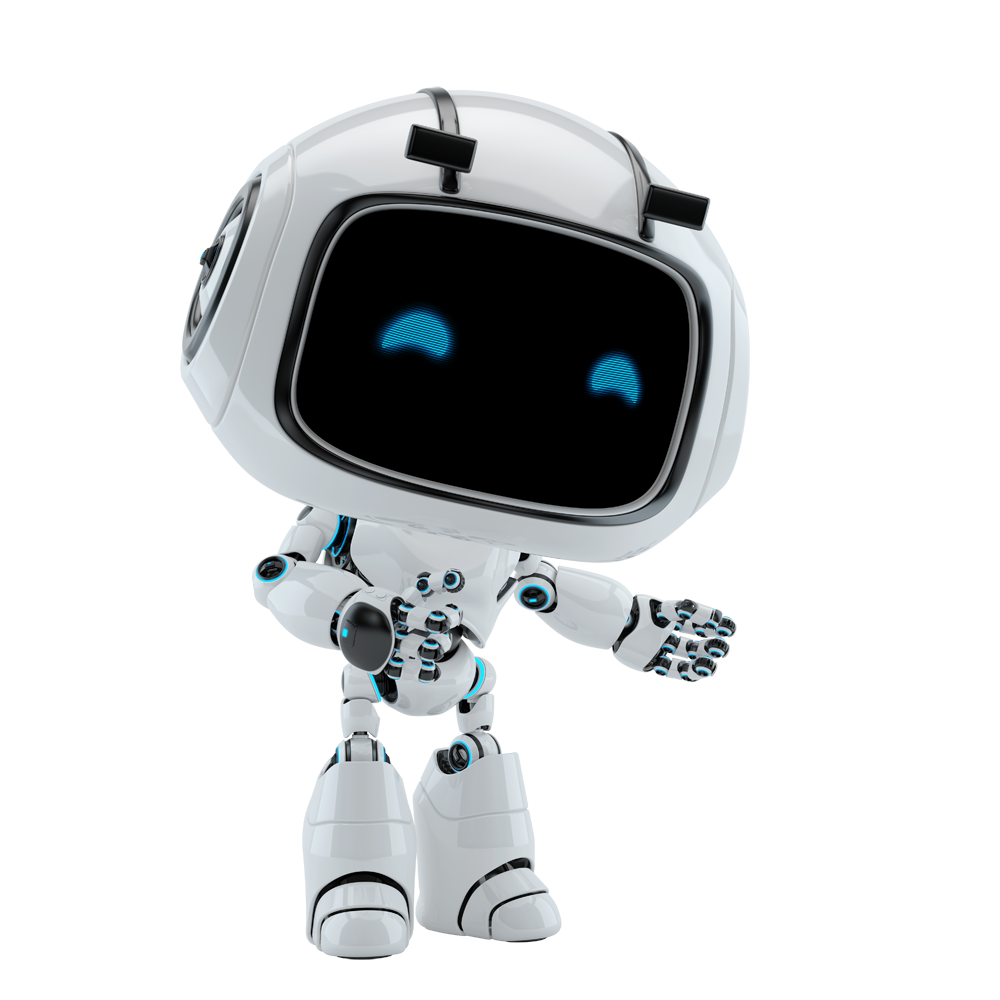


DEVELOPMENT SPECIFICATION DOCUMENT



[Date]

Grandida LLC

[Company address]

# PURPOSE

The purpose of the document is to record the outcome specific to the automated master project**,** Company Shelf**,** and its sub components: projects, workflows, sequences etc.This document alsoexplains the technical aspects of the robot designed and developed using UiPath in detail.

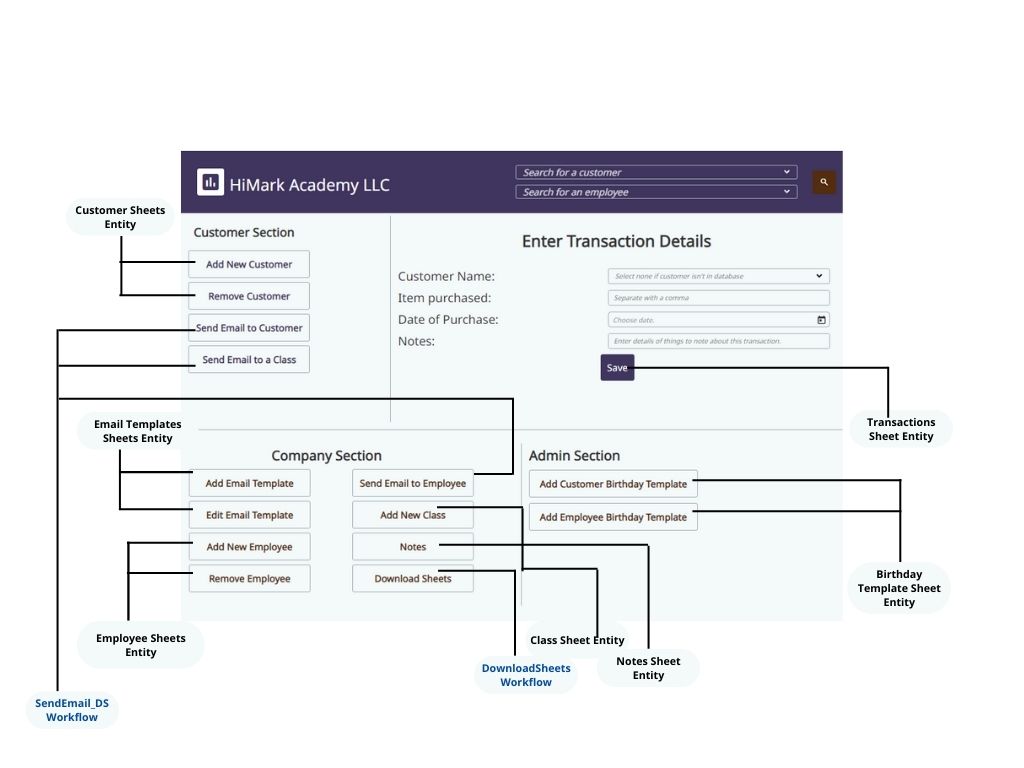
The content of this document is detailed and is designed to enable troubleshooting of the automated solution in UAT as well as post release into production. This will give an overview of the design of the bot and can be used by developers or other stakeholders to understand the prerequisites and requirements to execute the bot successfully.

# AUTOMATED PROCESS DETAILS

|  |  |
| --- | --- |
| Item | Description |
| Master Project Name | Company Shelf |
| Robot Type | FOR |
| Orchestrator used? | Yes |
| Scalable | Yes |
| UiPath version used | 2022.4.3 |

# RUNTIME GUIDE

## Architectural Structure of the Master Project



## Master Project Runtime Details

|  |  |
| --- | --- |
| ITEM NAME | DESCRIPTION |
| Production environment details | To be used on Desktop PC as an application. |
| Prerequisites to run | Having Excel on Machine  Storing Gmail credentials in Windows Credential Manager as “Gmail\_Login” |
| Input Data | User’s input from the desktop application interface. |
| Expected output | Excel files stored in a directory in the user’s Desktop |
| How to start the automated process | Open the desktop application |
| How to restart the process from a certain step? | Click the back arrow on the desktop application. |
| Reporting |  |
| Manual error handling? | Close the desktop application and restart. |
| How to resume the process in case of error? | Close the desktop application and restart. |
| How to manually fix transactions with error? | N/A |
| How is Orchestrator used? | To store files in storage bucket  To store information in data service entities |
| Password policies | N/A |
| Stored credentials | Gmail credentials in Windows Credential Manager as “Gmail\_Login” |
| List of Entities | Class Sheets, Customers Sheet, Customers Sheet Copy, Email Templates Sheet, Employees Sheet, Notes Sheet, Birthday Template Sheet, Transactions Sheets. |
| List of Storage buckets | SendEmail­\_File |
| Schedule Details | N/A |
| Multiple Resolutions Supported? | N/A |
| Recommended Resolution | N/A |

## Project Details

|  |  |
| --- | --- |
| ITEM NAME | DESCRIPTION |
| Environment used for development  (name, location, configuration details etc.) | UiPath Computer |
| Environment prerequisites | Windows 10, Unattended robot license, Microsoft Excel, Gmail account |
| Repository for project | https://github.com/Tessy8/Company\_Shelf |
| Configuration method | N/A |
| Configuration details | N/A |
| List of reused components | ***Component name:*** *N/A* |
| ***Description / comments:*** *N/A* |
| Login Level | N/A |
| Details about automation | UiPath Apps was used to design the interface for a friendly experience. Gmail credentials needed to send emails. |
| In case of FOR, can the user operate the computer while the robot is running? | Yes |
| Custom error logs defined in the workflows | N/A |
| Frequent errors found in the development phase | No internet connection  API call quota reached for entities. |
| Workarounds used in the automation phase | Capability to group customers for sending bulk emails. |

## Project Workflows

**SendEmail\_DS:** This process sends email(s).

* Get Gmail Credential from Windows Credential Manager
* Ensure maximum records are queried.
* The maximum number of records that can be queried is 1000.  
  This loops 10 times in order to query 10000 records.
* If there's no attachment to be sent alongside the email, find which one of customers or employees to send an email to and send.
* If there's an attachment to be sent alongside the email, find which one of customers or employees to send an email to, attach the file and send.
* Delete files after usage.

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| in\_EmailBody | InArgument(x:String) |  |
| in\_EmailSubject | InArgument(x:String) |  |
| in\_EmployeeName | InArgument(x:String) |  |
| in\_Class | InArgument(x:String) |  |
| in\_CustomerName | InArgument(x:String) |  |
| in\_Cc | InArgument(x:String) |  |
| in\_Bcc | InArgument(x:String) |  |
| in\_File | InArgument(x:String) |  |

**PROJECT DEPENDENCIES**

• UiPath.Credentials.Activities: [2.0.0]  
• UiPath.DataService.Activities: [21.10.1]  
• UiPath.Excel.Activities: [2.12.3]  
• UiPath.Mail.Activities: [1.15.2]  
• UiPath.System.Activities: [22.4.1]  
• UiPath.UIAutomation.Activities: [22.4.5]

**DownloadSheets:** This workflow downloads the files from data service entitities to a folder.

* Initialize counter variable to keep track when querying records (Query record as a max of 1000, Counter helps to terminate the loop after 10000 records have been queried.)
* Check if "Customer Shelf" folder exists else creates it.
* Build datatables to be used in the workflow.
* Queries data service entities to get their details.
* Add items queried to datatables
* Write to excel sheets.

**PROJECT DEPENDENCIES**  
• UiPath.DataService. Activities: [21.10.1]  
• UiPath.Excel.Activities: [2.12.3]   
• UiPath.Mail.Activities: [1.15.2]  
• UiPath.System.Activities: [22.4.1]  
• UiPath.UIAutomation.Activities: [22.4.5]

**SendBirthdayEmailMsg:** This workflow is scheduled to run daily. It sends emails to customers and employees on their birthdays.

* Initialize counter variable to keep track when querying records (Query record as a max of 1000, Counter helps to terminate the loop after 10000 records have been queried.)
* Queries data service entities to get their details.
* If an email template exists in the birthday template sheet entity, the process loops through the list to check for birthdays that are the current day’s date and month.
* If found, an email is sent to the customer or employee if an email is found.

**PROJECT DEPENDENCIES**  
• UiPath.Credentials.Activities: [2.0.0]  
• UiPath.DataService.Activities: [21.10.1]  
• UiPath.Excel.Activities: [2.12.3]  
• UiPath.Mail.Activities: [1.15.2]  
• UiPath.System.Activities: [22.4.1]  
• UiPath.UIAutomation.Activities: [22.4.5]

# GLOSSARY

The main terms used in the Solution Architecture Document are defined below:

**Master project** - the overall output of the development, containing one or multiple projects that together cover the scope of the robotic process automation. There is a 1 to 1 connection between the Master Project and the Process to be automated (As presented in the PDD).

**Project** - an 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. Or multiple projects can be converted into one package depending on the aims and restrictions of the automation. The project is used when defining the development and support phase of the automation.

**Package** - the output of compiling one or multiple projects. 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.