Solution Design Document

UiPath

TABLE OF CONTENTS

I. PURPOSE

The purpose of this Master Project is to deliver a robust Robotic Process Automation (RPA) solution designed to extract, validate, and report citation metrics for researchers from **Google Scholar**. The solution is built upon the **Robotic Enterprise Framework (REFramework)** to ensure high standards of exception handling, logging, and process stability.

The solution consolidates the **Dispatcher** (Data Scraping) and **Performer** (Data Processing) patterns into a single project using a logic-based split within the processing state. This approach ensures that the robot can autonomously generate its own workload (retrieving the list of papers) and subsequently process that workload (validating individual citation scores) within the same execution cycle.

Architectural Focus

The architecture has been designed with a specific focus on the following pillars:

- **Robustness:**
  - The solution leverages the **REFramework**'s native Retry Scope and Global Exception Handler to manage application exceptions (e.g., browser timeouts) and business exceptions (e.g., missing data fields).
  - By isolating transactions, a failure in processing one paper (e.g., a specific URL failing to load) does not stop the processing of the remaining queue items.
- **Scalability:**
  - The **Queue-based architecture** allows for horizontal scaling. While currently designed for a single robot, the decoupling of the "Workload Generation" (Producer) and "Workload Processing" (Consumer) allows multiple robots to consume items from the REFGoogleScholar queue simultaneously if volume increases significantly (e.g., processing a profile with 5,000+ citations).
- **Efficiency:**
  - **Bulk Loading:** The use of Table Extraction combined with Bulk Add Queue Items ensures that the initial retrieval of the publication list is performed in O(1) time relative to the page load, rather than iterating through list items individually on the profile page.
  - **Targeted Navigation:** The Consumer logic navigates directly to specific URLs provided by the queue, eliminating the need to search or traverse the profile page repeatedly.

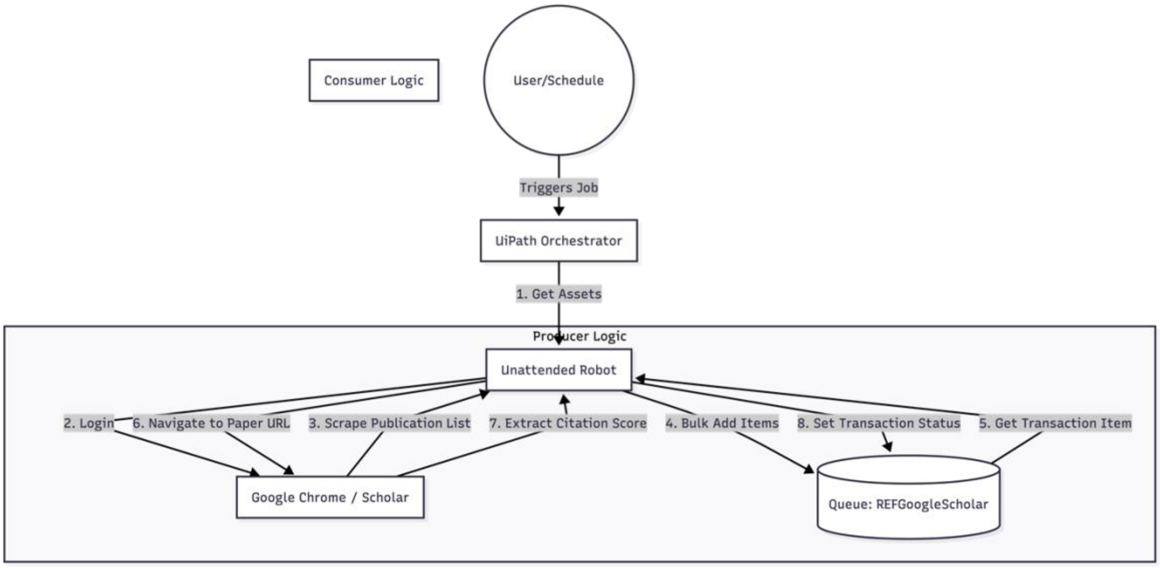
- Replicability:
  - The solution is environment-agnostic. All variable configurations (URLs, Credential Names, Queue Names) are stored in Config.xlsx and Orchestrator Assets, allowing the bot to be deployed from Development to UAT or Production environments without code changes.
- Reusability of Components:
  - The Login Module (InitAllApplications) is modular and can be reused for any future automation requiring Google authentication.
  - The Data Scraping Logic is designed to be adaptable to any Google Scholar profile structure.

II. AUTOMATED PROCESS DETAILS

Item	Description
Master Project Name	REFGoogleScholar
Robot Type	BOR (Back Office Robot / Unattended)
Orchestrator used?	Yes
Scalable	Yes
UiPath version used	26.0.182.0

2 RUNTIME GUIDE

2.1 Architectural structure of the Master Project



2.2 Master Project Runtime Details

Outlines the details of the automated process by filling in the table below.

ITEM NAME	DESCRIPTION
Production environment details	Running on a Standard Virtual Machine (Unattended Robot). The environment must have internet access to reach scholar.google.com.
Prerequisites to run	Google Chrome installed and updated. Microsoft Excel installed (required for reading Config.xlsx). UiPath Robot service running. Valid Google Account credentials available.
Input Data	Config.xlsx: Located in Data\ folder. Orchestrator Queue: REFGoogleScholar (populated automatically by the process or manually with a trigger item).
Expected output	Queue Items: Marked as Successful with the extracted citation score logged. Orchestrator Logs: Entries formatted as "Processing Paper: [Title]"
How to start the automated process	The process is triggered via UiPath Orchestrator (Jobs/Triggers page) or via API call.
Reporting	Orchestrator Logs and Queue Transaction Dashboards (monitoring Success vs. Business Rule Exceptions).

(queues reporting, Kibana or another platform)	
How is Orchestrator used?	Asset Management: Stores UserEmail and UserPassword securely. Queue Management: Manages the workload via the REFGoogleScholar queue. Triggering: Schedules the execution.
Password policies (mention any specific compliance requests)	Standard Google Account Policy. Multi-Factor Authentication (MFA) should be disabled for the service account, or App Passwords should be utilized if enforced.
Stored credentials (Never use hardcoded credentials in the workflow!)	Stored in Orchestrator Assets (Asset names: UserEmail, UserPassword). Never hardcoded.
List of queues names (Naming convention: ProcessName_QueueName)	REFGoogleScholar
Schedule Details	Recommended: Weekly (e.g., Monday 8:00 AM) to track citation changes over time, or Ad-hoc upon request.
Multiple Resolutions Supported? (in case of image automation / Citrix and VDI)	Yes (The automation utilizes Selectors and Maximize Window activities, making it resolution-independent).
Recommended Resolution	1920 x 1080 (Full HD) is recommended for optimal selector visibility.

2.3 Project name

ITEM NAME	**DESCRIPTION** *Fill in each section - empty fields are not allowed. If the section does not apply to your automation then mark as n/a.*
Environment used for development (name, location, configuration details etc)	Local Development Machine / VDI (Windows 10/11 Enterprise, Internet Access enabled for scholar.google.com)
Environment prerequisites (OS details, libraries, required apps)	**OS:** Windows 10/11 **Software:** UiPath Studio (v26.0.182.0), Google Chrome, Microsoft Excel. **Libraries:** UiPath.System.Activities, UiPath.UIAutomation.Activities, UiPath.Excel.Activities
Repository for project (where is the developed project stored)	**Local File System / Git Repository** (Project Source Control)
Configuration method (assets, excel file, Json file)	Hybrid Configuration: 1. **Config.xlsx** (Settings & Constants) 2. **Orchestrator Assets** (Credentials: UserEmail, UserPassword)
List of reused components	**Robotic Enterprise Framework (REFramework)** (Standard UiPath Template used as the project skeleton)

List of new reusable components	**N/A** (The scraping logic is specific to this Master Project and has not been extracted as a separate Library package yet)

2.4 Project(s) workflows

Workflows specific to: Specify Project Name from section above

For the workflow files defined below please specify the input and output parameters.

Workflow Name	Description	Input / Output Parameters
Main	The entry point of the automation. It orchestrates the entire process using the State Machine pattern (Init, Get Transaction, Process, End).	**In:** in_OrchestratorQueueName, in_OrchestratorQueueFolder **Out:** N/A
InitAllSettings	Loads configuration data from Config.xlsx and Orchestrator Assets into a Dictionary.	**In:** in_ConfigFile (String), in_ConfigSheets (String[]) **Out:** out_Config (Dictionary<String, Object>)
InitAllApplications	Opens Google Chrome, navigates to the Google Scholar URL, and performs the login sequence.	**In:** in_Config (Dictionary) **Out:** N/A
GetTransactionData	Retrieves the next item from the Orchestrator Queue (REFGoogleScholar) to be processed.	**In:** in_TransactionNumber, in_Config **Out:** out_TransactionItem, out_TransactionID
Process	The "Brain" of the operation. Contains two logical paths: 1. **Producer:** Scrapes paper list if input is a trigger. 2. **Consumer:** Navigates to a specific paper URL and scrapes the citation score.	**In:** in_TransactionItem (QueueItem), in_Config (Dictionary) **Out:** N/A
SetTransactionStatus	Updates the transaction status in Orchestrator (Success, Business Exception, or System Exception) and handles logging.	**In:** in_Config, in_TransactionItem, in_SystemException, in_BusinessException

		Out: N/A
CloseAllApplications	Logs out of Google Scholar and closes the Chrome browser gracefully.	**In:** N/A **Out:** N/A
KillAllProcesses	Forcefully terminates Google Chrome (chrome.exe) to ensure a clean state before initialization.	**In:** N/A **Out:** N/A
TakeScreenshot	Captures a screenshot of the desktop, typically used when a System Exception occurs.	**In:** in_Folder (String), io_FilePath (String) **Out:** io_FilePath (String)

2.5 Packages

Include the list of packages and high-level description for each of them, to explain their purpose

Package Name	Description
REFGoogleScholar.1.0.0.nupkg	**Google Scholar Citation Scraper Master Package** This package contains the complete logic for the Dispatcher and Performer processes: • **Initializes** the environment and retrieves credentials from Orchestrator. • **Scrapes** the target researcher's Google Scholar profile to identify all publications. • **Populates** the REFGoogleScholar queue with the list of papers. • **Processes** each queue item by navigating to the specific publication URL and extracting the precise citation score. • **Reports** the status and data back to UiPath Orchestrator logs.

3 OTHER DETAILS

Future Improvements

- **Implement Proxy Rotation:** Integrate a proxy management service to rotate IP addresses between transactions, preventing the host machine from being soft-banned by Google Scholar during high-volume processing.
- **CAPTCHA Integration:** Add an API integration (e.g., 2Captcha or Anti-Captcha) to automatically solve "I am not a robot" challenges if they appear, allowing the process to recover without human intervention.
- **Decouple Architecture:** Split the current "Process.xaml" logic into two distinct UiPath Projects (a dedicated **Dispatcher** to scrape the list and a dedicated **Performer** to process items). This would allow multiple robots to process the queue simultaneously.
- **Automated Email Reporting:** Add an SMTP Send Mail activity at the End Process state to automatically email the generated Excel report to the research team.
- **Date Filtering:** Add a configuration parameter to only process papers published within the last X years, optimizing the runtime for active researchers.

Other Remarks

- **Rate Limit Warning:** Google Scholar has strict anti-bot mechanisms. The workflow includes intentional delays (2–5 seconds) between page navigations to mimic human behavior. **Do not remove these delays**, as doing so will likely result in an immediate IP ban.
- **Scheduling Recommendation:** Citation metrics do not fluctuate hourly. It is recommended to schedule this process **Weekly** (e.g., every Monday at 6:00 AM) rather than daily, to reduce server load and the risk of blocking.
- **Resolution Requirement:** The robot assumes a screen resolution of **1920x1080**. Ensure the Unattended Robot machine is configured to this resolution to ensure the "Next Page" and "Citation" selectors are visible.
- **Account dependency:** The process relies on a specific Google Account to view the profile. Ensure the credentials in Orchestrator do not have 2-Factor Authentication (2FA) enabled, or use an App-Specific Password if required.

4 GLOSSARY

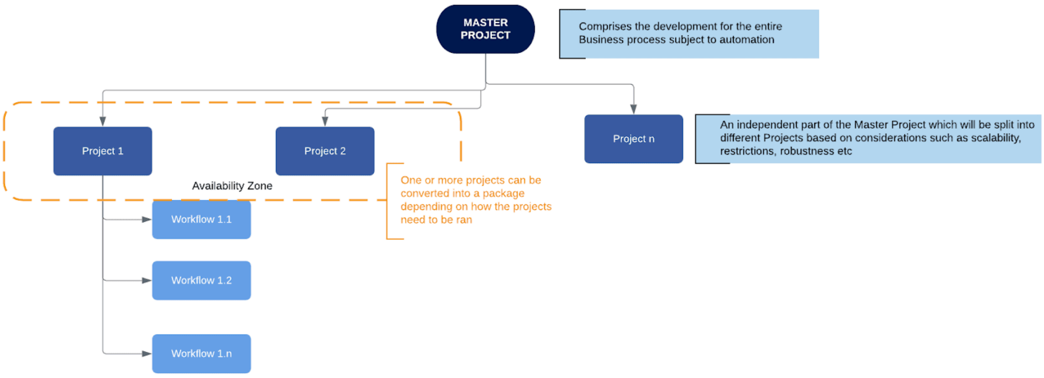
The main terms used in the Solution Architecture Document are defined below:

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.


MASTER PROJECT
Comprises the development for the entire Business process subject to automation
Project 1
Project 2
Availability Zone
Project n
An independent part of the Master Project which will be split into different Projects based on considerations such as scalability, restrictions, robustness etc
One or more projects can be converted into a package depending on how the projects need to be ran
Workflow 1.1
Workflow 1.2
Workflow 1.n

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.

REFramework (Robotic Enterprise Framework) – A standard, high-quality template provided by UiPath based on State Machines. It provides a robust structure for exception handling, logging, and transaction processing, serving as the backbone of this Master Project.

Dispatcher (Producer) – The logical component of the process responsible for extracting data from a source system (in this case, scraping the list of papers from Google Scholar) and uploading it to an Orchestrator Queue.

Performer (Consumer) – The logical component of the process responsible for retrieving items from the Orchestrator Queue and processing them one by one (in this case, navigating to the paper URL and validating the score).

Transaction Item – The smallest unit of data that is processed by the robot. In this project, a Transaction Item represents a single academic paper (containing the Title and URL) retrieved from the Orchestrator Queue.

Queue – A container in UiPath Orchestrator that holds Transaction Items. It acts as a buffer between the Dispatcher and Performer, allowing for workload management and scalability.

Asset – A shared variable stored in UiPath Orchestrator (e.g., "UserEmail", "UserPassword") used to securely manage configuration data and credentials without hardcoding them in the workflow.