# DEVELOPMENT SPECIFICATIONS DOCUMENT

## User Review NLP Analysis

# Contents


Document Information 3

Document History 3

Version Control 3

Document Overview 4

Automated Master Project details 5

Runtime Guide 6

Flow chart diagram 6

List of packages 7

Master Project Runtime details 7

Project details 8

Project Name: Virtual User Review NLP Analysis 8

Workflow(s) specific to Virtual User Review NLP Analysis 9

Other Details 12

Future improvements 12

Debugging tips 12

Other Remarks 12

Glossary 13

# Document Information

## Document History

| Date | Version | Role | Name | Organization Department | Function | Comments |
|---|---|---|---|---|---|---|
| 19 Jan 2020 | 1.0 | DSD | *DSD – TechGig – Virtual - User Review NLP Analysis* | *NA* | *NA* | NA |

## Version Control

| Date | Version | Role | Name | Organization Department | Function | Comments |
|---|---|---|---|---|---|---|
| 19 Jan 2020 | 1.0 | Author | Krutika | NA | NA | NA |
| 19 Jan 2020 | 1.0 | Peer Reviewer | Pranav | NA | NA | NA |
| 19 Jan 2020 | 1.0 | Reviewer | Pranav | NA | NA | NA |
| 19 Jan 2020 | 1.0 | Approver | Prathamesh | NA | NA | NA |

# Document Overview

This Development Specifications Document (DSD) is created for business process that is automated using the RPA technology. The process will explains how ecommerce users have to look for reviews to a product on every site to make a decision. With this project, users just need to provide a product name and the UiPath robot will analysis reviews on multiple ecommerce websites using Natural Language Processing (NLP) and generate an end user report. This report consists of overall ratings each website has for a particular product, analysis of positive and negative reviews. Also the report can analyses the reason behind the review to be positive or negative.

# Automated Master Project details

| # | Item | Details |
|---|---|---|
| 1 | Master project name and version | TechGig_Virtual_UserReviewNLPAnalysis – version 1.0 |
| 2 | Robot type | Unattended Robot |
| 3 | Is Orchestrator used? | No |
| 4 | Scalable? | Yes |

# Runtime Guide

## Flow chart diagram

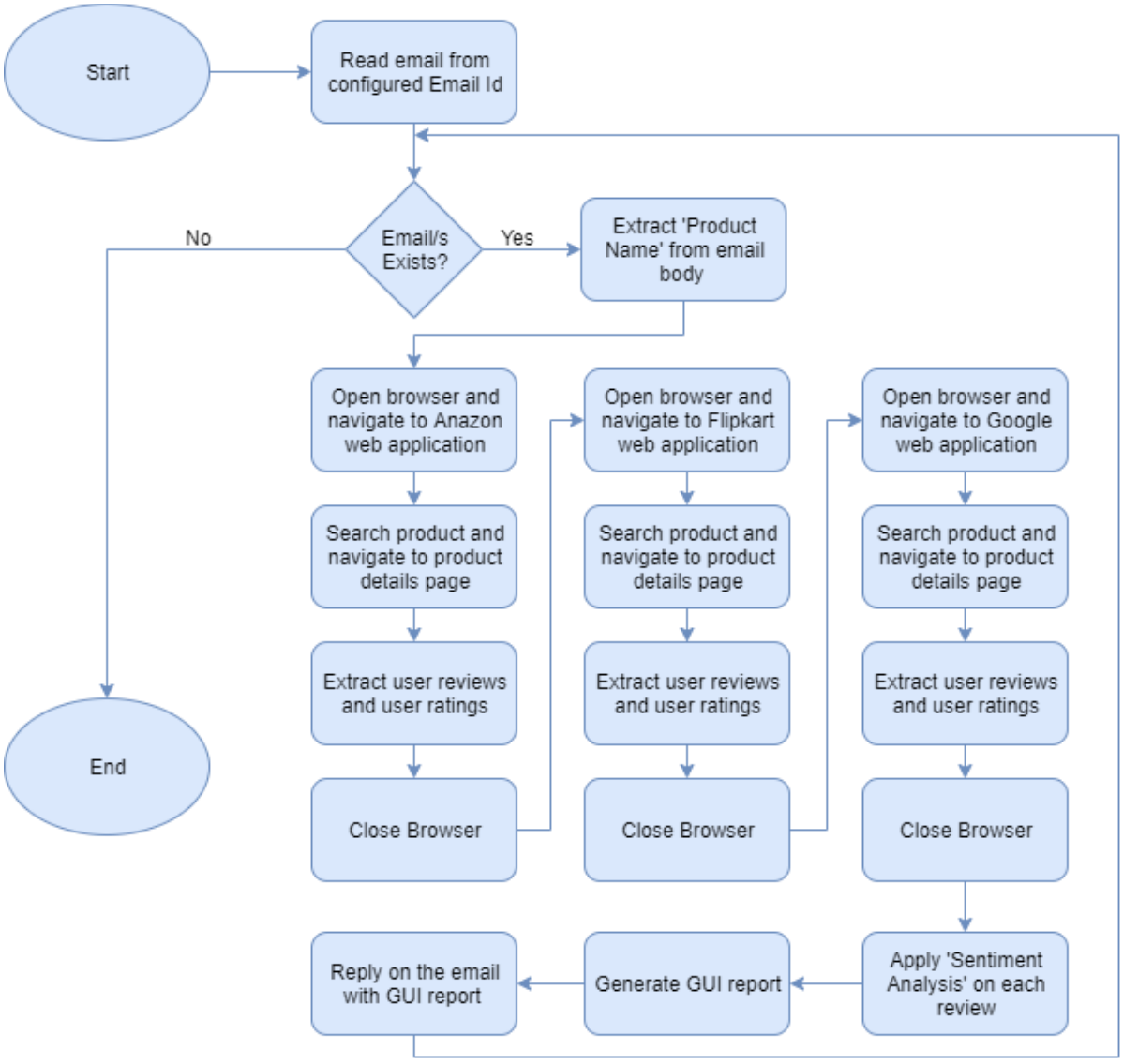


Fig. Flow Chart

## List of packages

| # | Package name | High Level description |
|---|---|---|
| 1 | Amazon.nupkg | This package extracts review from Amazon website for the requested product by user |
| 2 | Flipkart.nupkg | This package extracts review from Flipkart website for the requested product by user |
| 3 | Google.nupkg | This package extracts all google reviews for the requested product by user |
| 4 | Outlook.nupkg | This package reads the user request for product review |
| 5 | Sentiment API.nupkg | This package Analyses the reviews of the products extracted from the website |

## Master Project Runtime details

| # | Item | Details |
|---|---|---|
| 1 | Production environment details | N/A |
| 2 | Prerequisites to run | IE ,Chrome , Outlook desktop application |
| 3 | Input Data | Reviews extracted from the websites |
| 4 | Expected Output (output data) | End user report consisting of ratings each website has for a specific product. |
| 5 | How to start the automated process? | Execute the Main.xaml file in the project folder |
| 6 | Use of Orchestrator | N/A |
| 7 | Multiple resolutions supported | N/A |

# Project details

## Project Name: Virtual User Review NLP Analysis

| # | Item name | Details |
|---|---|---|
| 1 | Environment used for development | Processor: Intel ® Core™ i3-6006U CPU @ 2.00GHz 1.99GHz<br>Memory : 4.00GB<br>Hard Disk : 500GB |
| 2 | Environment prerequisites | OS – Windows 10 64 –bit OS x64 based processor. |
| 3 | Logging level | Trace Log, Error log, Orchestrator log |
| 4 | Details about automation | UI Automation. |
| 5 | In case of FOR, can the user operate the computer while the robot is running? | N/A |
| 6 | Repository for project | TBD |
| 7 | List of reused components | • Flipkart Components<br>• Amazon Components<br>• Google Components<br>• Outlook Components<br>• Excel Components |
| 8 | Custom logs defined in the workflows | N/A |
| 9 | Frequent errors found in the development phase | • Faced errors in unit testing of NLP API where after NLP processed the text different status codes were received which was not expected.<br>• Flipkart Popup that was displayed once the browser was invoked was breaking the code during unit testing. |
| 10 | Workarounds used in the automation phase | JavaScript used |
| 11 | Configuration method | Excel |
| 12 | Configuration details<br>(path for input files, configuration Orchestrator assets used) | TBD |

## Workflow(s) specific to Virtual User Review NLP Analysis

| # | Workflow file name | Description | Input Argument | Output Argument |
|---|---|---|---|---|
| 1 | Main.xaml | | | |
| 2 | InitAllSettings | This workflow outputs a settings Dictionary with key/value pairs to be used in the project. Settings are read from local config file then fetched from Orchestrator assets. Assets will overwrite the config file settings | in_ConfigFile.<br>in_ConfigSheets. | out_Config |
| 3 | InitAllApplications | Open and initialize application as needed. | in_Config | out_LstMailMessages |
| 4 | URNP - Get Email Details.xaml | Get email details from outlook | in_IntTimeoutS.<br>in_IntTimeoutL.<br>in_IntTimeoutM.<br>in_IntDefaultDelay.<br>in_StrEmailAccount.<br>in_StrFolderName.<br>in_StrSubjectFilter.<br>in_IntNoOfEmailsToRead. | out_LstMailMessages |
| 5 | GetTransactionData.xaml | Get data from spreadsheets, databases, email, web API or UiPath server queues. If no new data, set out_TransactionItem to Nothing. | in_TransactionNumber.<br>in_Config.<br>io_TransactionData. | io_TransactionData.<br>out_TransactionField.<br>out_TransactionField2.<br>out_TransactionID.<br>out_TransactionItem. |
| 6 | Process.xaml | It process the records in the list | in_TransactionItem.<br>in_Config. | |
| 7 | URNP - Application Data Initialization.xaml | Initialize application data<br>- Add data coloum in review table<br>- Extract product name from email body | io_DtNLPProcessDataTable.<br>in_strEmailBody. | io_DtNLPProcessDataTable.<br>out_strProductName. |
| 8 | URNP - Get Product Reviewa From Amazon.xaml | Get Product Reviews From Amazon. | in_StrAmazonURL.<br>in_IntTimeoutS.<br>in_IntTimeoutL.<br>in_IntTimeoutM. | io_DtNLPReviews.<br>out_DictAmazonProductReviews |

| | | | | |
|---|---|---|---|---|
| | | | in_IntDefaultDelay.<br>in_StrProductName.<br>in_StrBrowserType.<br>in_IntNoOfReviewsToProcess.<br>io_DtNLPReviews | |
| 9 | URNP - Get Product Reviews From Flipkart.xaml | Get Product Reviews From Flipkart. | in_StrFlipkartURL.<br>in_IntTimeoutS.<br>in_IntTimeoutL.<br>in_IntTimeoutM.<br>in_IntDefaultDelay.<br>in_StrProductName.<br>in_StrBrowserType.<br>in_IntNoOfReviewsToProcess.<br>io_DtNLPReviews | io_DtNLPReviews.<br>out_DictFlipkartProductReviews |
| 10 | URNP - Get Product Reviewa From Google.xaml | Get Product Reviews From Google. | in_StrGoogleURL.<br>in_IntTimeoutS.<br>in_IntTimeoutL.<br>in_IntTimeoutM.<br>in_IntDefaultDelay.<br>in_StrProductName.<br>in_StrBrowserType.<br>in_IntNoOfReviewsToProcess.<br>io_DtNLPReviews. | io_DtNLPReviews.<br>out_DictGoogleProductReviews. |
| 11 | URNP - Extract Sentiment Analysis.xaml | Extract sentiment analysis data from NLP server | io_DtNLPProcessDataTable.<br>in_StrAPIName.<br>in_StrKey.<br>in_StrServiceUrl.<br>in_StrUsername.<br>in_StrPassword.<br>in_StrServiceVersion.<br>in_StrLanguage. | io_DtNLPProcessDataTable |
| 12 | URNP - Create Report And Reply.xaml | - Create HTML report using NLP data<br>- Reply on email | in_DtNLPProcessDataTable.<br>in_StrProductName. | |

| | | | | |
|---|---|---|---|---|
| | | | in_MmsgEmailMessage.<br>in_DictAmazonProductReview.<br>in_DictFlipkartProductReview.<br>in_DictGoogleProductReview.<br>in_StrNLPBadThreshold.<br>in_StrNLPAvaragеThreshold.<br>in_StrHTMLFilePath. | |

# Other Details

## Future improvements

1. Reading the user request from outlook using Natural language processing.
2. Accepting multiple product for reviews and presenting graphical report for multiple products.
3. This process could be extended to cover more functional areas like hotel etc.

## Debugging tips

N/A

## Other Remarks

This entire process contains reusable components like Flipkart libraries, amazon libraries, outlook libraries that could be used by other project. For example if the requirement of any project is to just read the emails based on specific subject or only extract the product review on Flipkart.

# 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 a .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.