­­­­­

|  |  |
| --- | --- |
|  |  |



Solution Design

Document

Table of Contents

[I. Purpose 3](#_Toc5787525)

[II. Automated process details 4](#_Toc5787528)

[3 Runtime guide 5](#_Toc5787529)

[3.1 Architectural structure of the Master Project 5](#_Toc5787530)

[3.2 Master Project Runtime Details 5](#_Toc5787531)

[3.3 Project name 1 6](#_Toc5787532)

[3.4 Project(s) workflows 6](#_Toc5787533)

[3.5 Packages 7](#_Toc5787534)

[3.6 Architectural structure of the Master Project 7](#_Toc5787535)

[4 Other Details 8](#_Toc5787536)

[Future Improvements 8](#_Toc5787537)

[Other Remarks 8](#_Toc5787538)

[5 Glossary 9](#_Toc5787539)

# Purpose



The purpose of our project, Getaway Guru, is to assist clients with their vacation needs to give suggestions for the potential hotels to stay at, activities they can do, and local restaurants they would like to dine at. Getaway Guru also lets the client know about the weather in the area they stay at in advance. Every process is fully automated, and the output will be an excel file that will display all information about the hotels, weather, activities, and restaurant info. These processes use REST API’s and website scraping to build the report.

# Automated process details

Details filled in need to reflect the actual information for the Master Project released for production. The following table will be populated:

|  |  |
| --- | --- |
| Item | Description |
| Master Project Name | Getaway Guru |
| Robot Type | Attended Robot |
| Orchestrator used? | Yes |
| Scalable | YES |
| UiPath version used | 23.4.0 |

# Runtime guide

## Architectural structure of the Master Project

Display the interaction between components (package / robots, Orchestrator queues, and running order) in a diagram

## Master Project Runtime Details

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

|  |  |
| --- | --- |
| ITEM NAME | DESCRIPTION  *Fill in each bolded section - empty fields are not allowed. If the section does not apply to your automation then mark as n/a.* |
| Production environment details | ***Running from the orchestrator, the file with info is saved and used for the workflows that will get different information for the travelers’ destination*** |
| Prerequisites to run | ***Email send of travelers’ info*** |
| Input Data | ***One Excel file that contains travelers’ info*** |
| Expected output | ***An email sent of an excel file with info grabbed from the process along with a personalized message*** |
| How to start the automated process | ***Start project in UiPath Studio*** |
| Reporting  (queues reporting, Kibana or another platform) | ***Orchestrator logs and jobs dashboards.*** |
| How is Orchestrator used? | ***Asset passwords and API keys, as well as the queue.*** |
| Password policies  (mention any specific compliance requests) | ***N/A*** |
| Stored credentials  (Never use hardcoded credentials in the workflow!) | ***Credentials for G-mail integration and API keys*** |
| List of queues names  (Naming convention: ProcessName\_QueueName) | ***tripQueue*** |
| Schedule Details | ***At users’ convenience*** |
| Multiple Resolutions Supported?  (in case of image automation / Citrix and VDI) | ***N/A*** |
| Recommended Resolution | ***N/A*** |

## 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) | ***GetawayGurus(Github Repository)*** |
| Environment prerequisites  (OS details, libraries, required apps) | ***Microsoft excel, Windows 10/11, G-mail, UiPath, WeatherApi key*** |
| Repository for project  (where is the developed project stored) | *https://github.com/230206-UiPath-BNYM/GetawayGurus* |
| Configuration method  (assets, excel file, Json file) | ***Assests, Excel Files*** |
| List of reused components | ***N/A*** |
|
| List of new reusable components | ***N/A*** |

Add tables for as many projects as you need and fill them in.

## 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 |
| Main | ***Invokes all other workflow files*** |
| Create\_Customer\_Files | ***Creates file for each customer based on a template*** |
| CurrentWeather | ***Takes location as input, sends current weather request to the weather API, outputs data table with weather information added as well as latitude and longitude.*** |
| ForecastWeather | ***Takes location, start date, and length of stay as input, sends forecast weather request to the weather API, outputs data table with weather information added as well as latitude and longitude.*** |
| FutureWeather | ***Takes location, start date, and length of stay as input, sends future weather request to the weather API, outputs data table with weather information added as well as latitude and longitude.*** |
| Get\_Activities | ***Outputs Activity name and address as data table to be stored in customer file*** |
| Get\_Flight\_Info | ***Data scrapes website to get flight into as data table to be stored in customer file*** |
| Get\_Hotels | ***Data scrapes website to get hotel info as data table to be stored in customer file*** |
| GetWeather | ***Main weather file. Takes location, start date, and end date as input, invokes relevant weather file depending on dates after calculating length of stay. Outputs the completed weather data table as well as latitude and longitude of the location.*** |
| Send\_Final\_Customer\_Email | ***Sends a personalized email to the customer that includes an excel file with their collected trip information*** |
| Suggest\_Restaurants | ***Outputs Restaurant name and address as data table to be stored in customer file*** |

## Packages

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

|  |  |
| --- | --- |
| Package Name | Description |
| *Azure.Core = 1.30.0* | ***Implementation of the Azure Client Pipeline*** |
| *Google.APIs.Auth = 1.60.0* | ***Runtime Client for working with Google services*** |
| *Google.Apis.Gmail.v1 = 1.60.0.2958* | ***Client Library for working with Gmail*** |
| *UiPath.Azure.Activities = 1.3.2* | ***Package to be able to implement Azure studio databases in UiPath*** |
| *UiPath.Database.Activities = 1.7.1* | ***Package used to manipulate databases and get info from databases*** |
| *UiPath.Excel.Activities = 2.16.2* | ***Execute Excel related operation using for the Open Office XML format(XLSX) or Excel.Interope*** |
| *UiPath.Mail.Activities = 1.18.2* | ***Retrieve and send mail using POP3, IMAP, SMTP and exchange protocols*** |
| *UiPath.System.Activities = 22.10.4* | ***Core activities which enable the robots to manipulate data tables and collections, work with files and folders. Also uses workflow operators, dialog forms, debugging and invoking methods*** |
| *UiPath.Testing.Activites = 22.10.3* | ***Testing activities which enable users to verify expression values and control states in test case workflows*** |
| *UiPath.UIAutomation.Activities = 22.10.5* | ***Contains core activities which enable the automation of desktop applications, browsers, and virtual machines*** |
| *UiPath.WebAPI.Activities = 1.13.3* | ***Activities that enable the automation of various software components with less effort by using their APIs, including JSON, REST, SOAP, XML.*** |

## Architectural structure of the Master Project

Display the interaction between components (package / robots, Orchestrator queues, and running order) in a diagram.

Diagram, table

Description automatically generated

# Other Details

### Future Improvements

Fill in any improvements that need to be considered for the future:

*• Personalized Exception handling if the user inputs wrong format of input variables*

*• Future, send out a form to add new client info to database.*

*• Pinpoint exact Latitude and longitude of where they are staying to get more accurate events and restaurants*

### Other Remarks

Please mention here any other points that you consider relevant for the automation process.

*Some Addresses in different country are not obtainable from the foursquare API.*

# 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.