**Solution Design Document**

**Hotel Booking System**



**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Purpose | Author | Reviewer | Release Date |
| v.1 | Initial Document Overview | Shivam Gupta/Madhav Lomash | Rutika Jadhav | 16-01-2024 |
| v.2 | Workflow design changes | Shivam Gupta/Madhav Lomash | Rutika Jadhav | 16-01-2024 |
| v.3 | Workflow design changes | Shivam Gupta/Madhav Lomash | Rutika Jadhav | 17-01-2024 |
| v.4 | Formatting changes | Shivam Gupta/Madhav Lomash | Rutika Jadhav | 17-01-2024 |

**Table of Contents**

[**1.** **Introduction/Background** 4](#_Toc153891613)

[**2.** **Functional Requirements Overview** 4](#_Toc153891614)

[**2.1** **Current Process** 4](#_Toc153891615)

[**2.2** **Future Process** 4](#_Toc153891616)

[**3.** **Proposed Solution** 5](#_Toc153891617)

[**3.1** **BOTs Design** 5](#_Toc153891618)

[**3.1.1 BOT Scope** 5](#_Toc153891619)

[**3.1.2 BOT Goal and Objective** 5](#_Toc153891620)

[**3.1.3 Design Details** 6](#_Toc153891621)

[**3.1** **BOT Workflow** 7](#_Toc153891622)

[**3.2** **Interventions Required** 8](#_Toc153891623)

[**4** **Technical Design:** 8](#_Toc153891624)

[**4.1** **Technical Architecture** 8](#_Toc153891625)

[**4.2** **Target System** 8](#_Toc153891626)

[**5** **Non-Functional Requirements** 9](#_Toc153891627)

[**5.1** **Security** 9](#_Toc153891628)

[**5.2** **Availability Requirements** 10](#_Toc153891629)

[**5.3** **Volume and Performance Expectations** 10](#_Toc153891630)

[**6** **Key Assumptions & Dependencies** 10](#_Toc153891631)

# **Introduction/Background**

Trivago is a hotel metasearch engine that aggregates data from multiple online distribution channels to enable price comparison in one place. The website allows users to compare accommodation prices and offers provided by many different online booking sites.

Trivago works with many booking sites worldwide, including online travel agencies, as well as accommodation chains and independent hotels.

Trivago's search function is designed to enable individual users to find their ideal hotel. The personalized matching algorithms help travelers to get a comprehensive understanding of the hotel and its value proposition. This makes them more likely to complete a booking on the advertiser's site.

Trivago is now looking for automated solution to read emails from its dedicated customer to search for available hotels based on ratings from previous customers, Price Per Night, city availability etc.

This document covers functional specification of the current process being followed by the users.

# **Functional Requirements Overview**

## **Current Process**

1. Once the system receives an email with the subject " Hotel Booking System " and any attachments, it will get triggered.
2. User read the email with the subject” Hotel Booking System”.
3. User manually Extracts email attachment details such as destination, check-in check-out, price, etc. & saves them to an Excel file.
4. User Navigate to the “Trivago” web application(<https://www.trivago.in/>) and enter the required input details (destination, check-in check-out, price, etc.) for filling form on (<https://www.trivago.in/>) which are mentioned in PDF attachments received via email.
5. From available hotels (on the first web page of “Trivago”), the User manually extracts details such as Hotel Name, City, Ratings, Price, Website from which hotel booking will be done, etc. into the Excel sheet.
6. From the extracted Excel sheet having details such as Hotel Name, City, Ratings, Price, Website from which hotel booking will be done, etc. Sort them into the ascending order of “price per night” for all available hotels.
7. The user will send the sorted Excel data via email to the respective client.

## **Future Process**

1. The bot will get triggered once it receives an email with the subject "Hotel Booking System" and any attachments.
2. The bot will read the email with the subject “Hotel Booking System”.
3. Bot will check for the specific subject if it is true then
4. Bot will check whether it is pdf or in some other format
5. If it is a pdf, bot will extract relevant data (i.e. Destination, Check-In, Check-Out etc.) and save them into Excel file.
6. Bot will Navigate to the “Trivago” web application(<https://www.trivago.in/>).
7. Bot will enter the required input details (destination, check-in, etc.) for filling form on (<https://www.trivago.in/>), which are mentioned in PDF attachments received via email, and click the search button.
8. After clicking the search button list of available hotels (on the first web page of “Trivago”) will appeared.
9. The Bot will automatically extract details such as Hotel Name, City, Ratings, Price, Website from which hotel booking will be done, etc. into a structured format i.e. Excel file.
10. From the extracted Excel sheet having details such as Hotel Name, City, Ratings, Price, Website from which hotel booking will be done, etc. The bot will Sort them into the ascending order of “price per night” for all available hotels and save the output in a structured format i.e. Excel file.
11. The bot will send the sorted Excel data via email to the respective client.

# **Proposed Solution**

## **BOTs Design**

### **BOT Scope**

The high-level steps to perform the Online hotel booking process:

* The bot will be activated when it receives an email with a specific "Hotel Booking System" containing attachments.
* The bot will create the required folder structure using the configuration file “configFile.xml”.
* The bot will extract email attachments from the email” Inbox” folder to the directory on the machine “C:\Bots\Hotel Management System\Input File”.
* Bot will check “If the attached files are in PDF format, if so (follow the below steps) otherwise show the exception” invalid attachment type found”
* If the file is pdf, then:

1. Extra the details like “destination”, “Check-in/check-out”, etc. to input Excel file.

1. Bot will start validating Excel file for the correct format
2. Bot will start validating “if any null value is available or not” If so then handle the exceptions as “data not found”
3. Bot will start validating whether each data filed is of valid type or not (necessary in case of check-in/check-out)

2. If all details are valid then the bot will try to open/launch the website (<https://www.trivago.in/>) soon it saves the details in an Excel file

3. The bot will check if the browser opens successfully, if not wait for the browser to open.

4. if the browser opens successfully then the bot starts navigating/searching for available hotels by entering Mandatory fields like:

1. Destination
2. Check-in/Check-out
3. Rooms/Guest availability etc. &

5. The bot may add filter data for exact search if available in the input file/pdf file.

6. All the necessary data items like “ratings”, “distance from the city”, “price per night”

is stored on the output Excel file.

7. Repeat the steps(1-6) for each detail in the input Excel file.

8. After all required details are extracted to an output Excel file (say).

Then excel file will be sorted based on the “price per night” for each available hotel.

On the first search page of the Trivago website.

9. The final Excel file (having sorted hotel details) is sent to the client/customer.

Through email provided (at the time of email trigger).

* Finally, Bot will stop its process for automation.

### **3.1.2 BOT Goal and Objective**

The customer is looking for BOT to achieve the below goals.

1. Standardize the input formats.
2. Store -Real-time and exact hotel details (i.e. Price/night, ratings, etc.) in an Excel file.
3. Speed -Reduce man-hours of doing this data entry manually by more than 70%.
4. Availability – This functionality is available throughout the time.

### **3.1.3 Design Details**

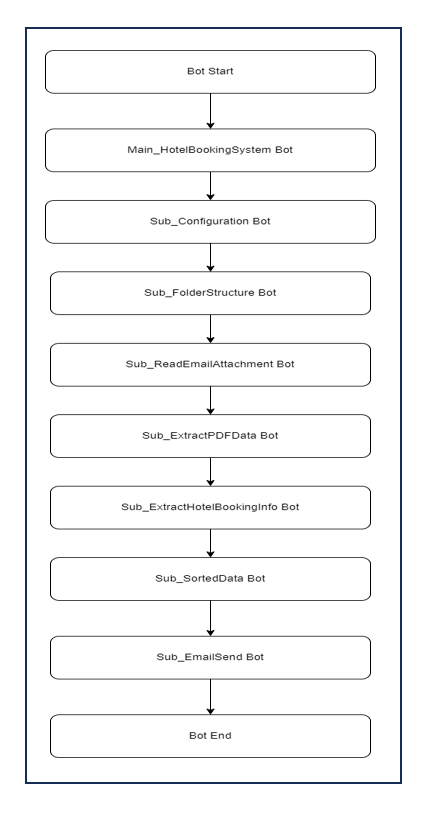
* 1. **Main\_HotelBookingSystem BOT:** This bot is the parent bot that calls a child task bot to execute modular functionality.
  2. **Sub\_Configuration BOT:** This bot helps to achieve virialized the input variables.
  3. **Sub\_FolderStructure BOT:** This bot is responsible for creating folders and files like audit logs and error logs.
  4. **Sub\_ReadEmailAttachment BOT:** This bot is responsible for reading emails with specific references and attachments (PDF).
  5. **Sub\_ExtractPDFData BOT:** This bot will extract details (check-in/out, destination, etc.) from PDF email attachments.
  6. **Sub\_ExtractHotelBookingInfo BOT:** This bot extracts hotel details(e.g.name, ratings, price/night) into an Excel file.
  7. **Sub\_SortExcelData BOT:** This bot will sort hotel details (i.e.name, ratings and price per night etc.) in ascending order by price per night.
  8. **Sub\_EmailSend BOT:** The bot will send the sorted output file to the sender via email as a final step.

**Packages used in Bot development**:

1. **Trigger package**: It is used to specify the conditions under which a bot should be executed automatically.
2. **Browser Package:** To navigate through the website.
3. **Task Bot Package:** To achieve modularization child bots are created. To execute the child bot from the main/parent bot task bot package is used.
4. **Loop Package:** To go through a series of actions.
5. **Recorder Package:** To interact through the UI elements and perform various operations.
6. **Excel Advanced Package:** Package used for Excel operations like creating the raw file for the input data and filtering it out to create a new file.
7. **Email Package:** This is used read to send an email with an input file as an attachment.
8. **Error Handler Package:** Used for exception handling.
9. **If Package:** Package is used for conditional scenario handling.
10. **Delay Package:** The package is used to delay the screen for some time.
11. **Comment:** This package is used to give extra information about the process.
12. **Steps:** This package is used to streamline the same domain task.
13. **PDF Package:** This package is used to automate various operations on PDF files.

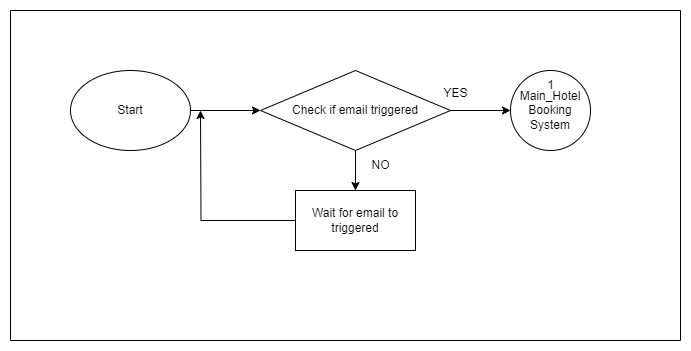
## **BOT Workflow**

The High-level workflow of the File Processing is given below.

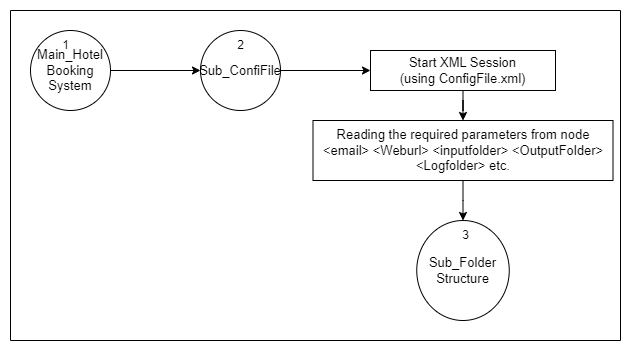


**3.2 Bot Detailed Workflow**

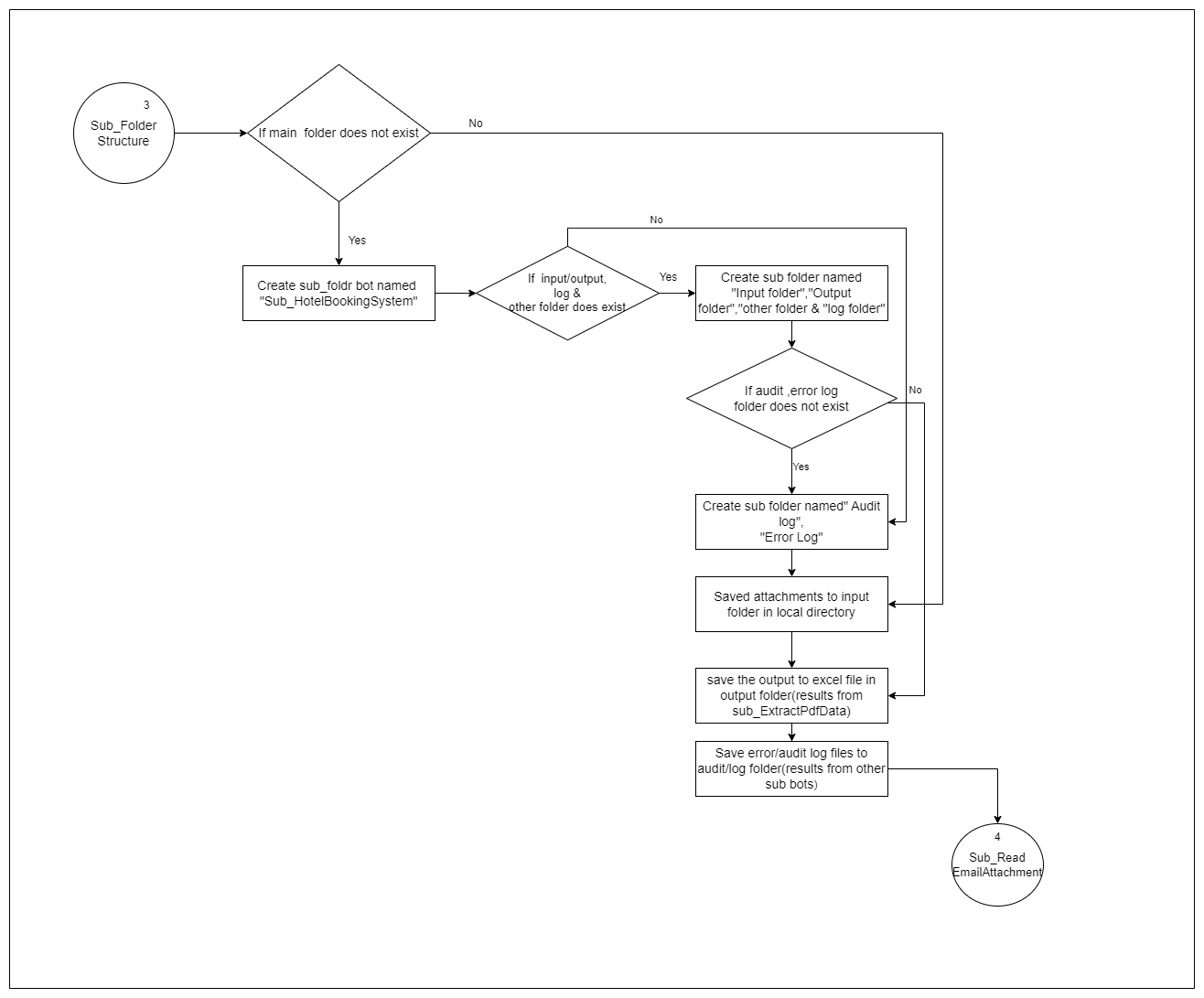
**1. Flow diagram for Main\_HotelBookingSystem Bot is given below:**

****

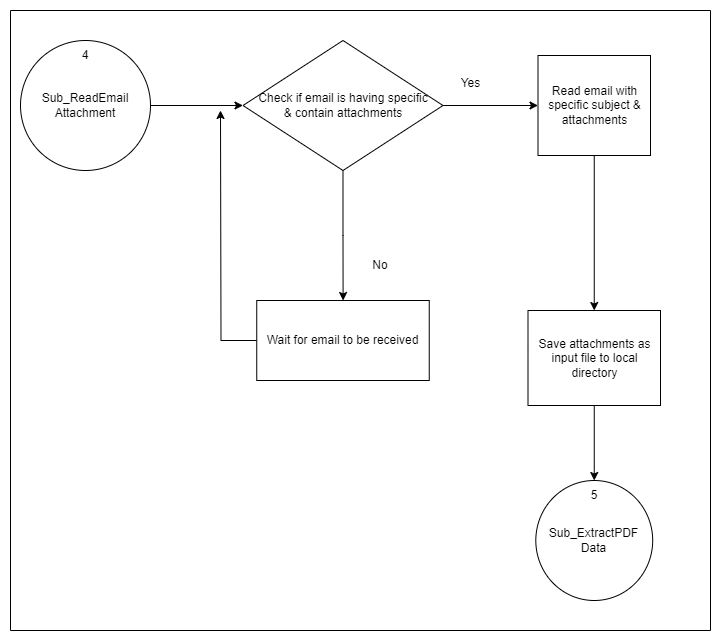
1. **Flow diagram for Sub\_ConfigFile Bot is given below:**

****

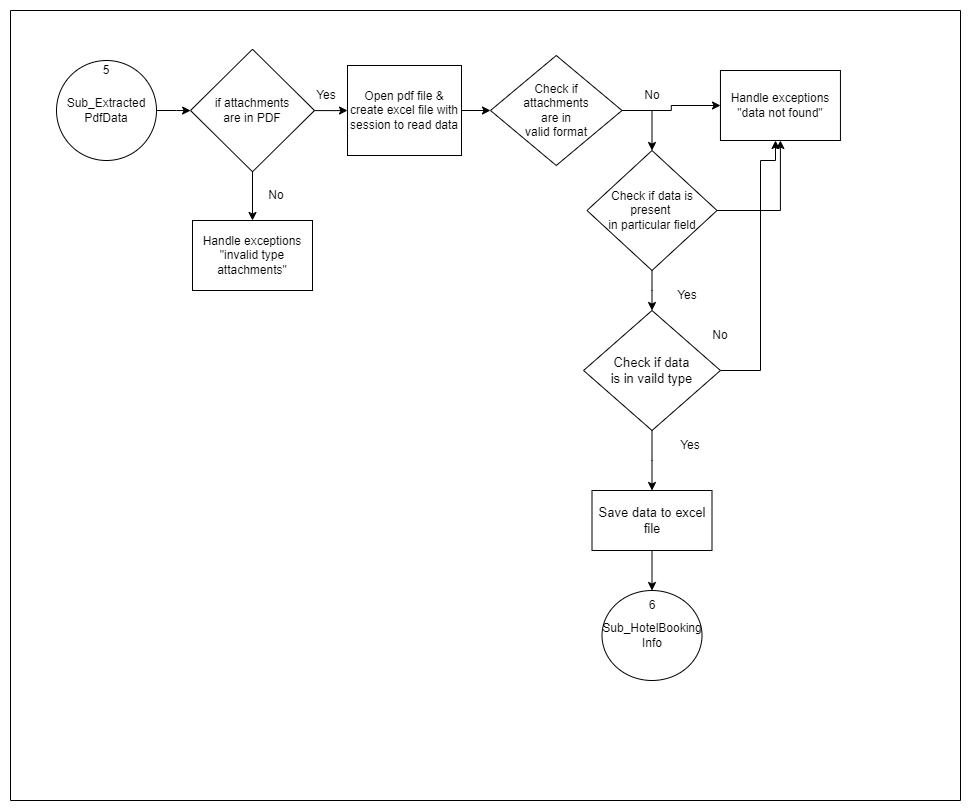
1. **Flow diagram for Sub\_FolderStructure Bot is given below:**

****

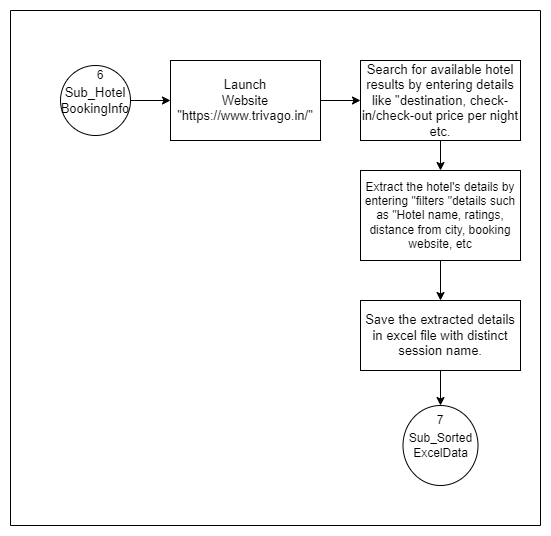
1. **Flow diagram for Sub\_ReadEmailAttachments Bot is given below:**

****

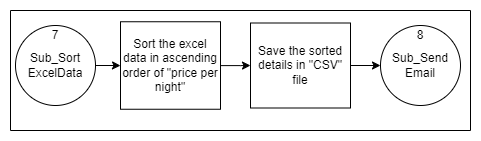
1. **Flow diagram for Sub\_ExctractPdfData Bot is given below:**

****

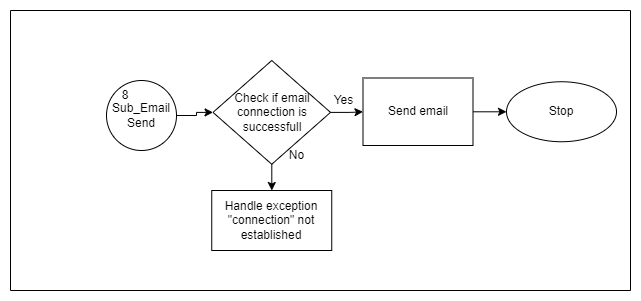
1. **Flow diagram for Sub\_HotelBookingInfo Bot is given below:**

****

1. **Flow diagram for Sub\_SortExcelData Bot is given below:**

****

1. **Flow diagram for Sub\_SendEmail&Confirm Bot is given below:**

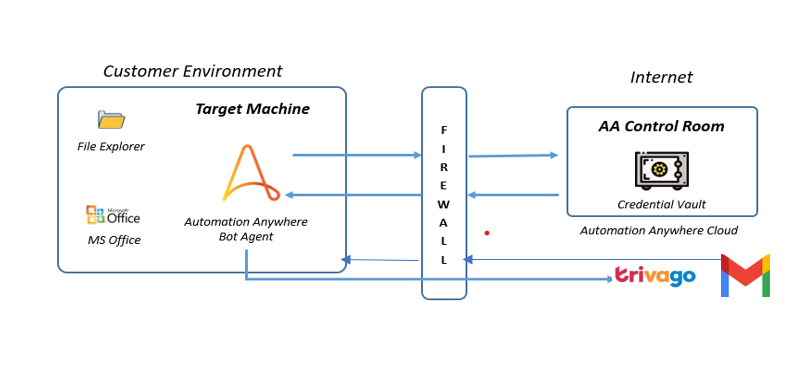
****

## **Interventions Required**

* NA

# **4.Non-Functional Requirements**

## **Technical Architecture**



## **Target System**

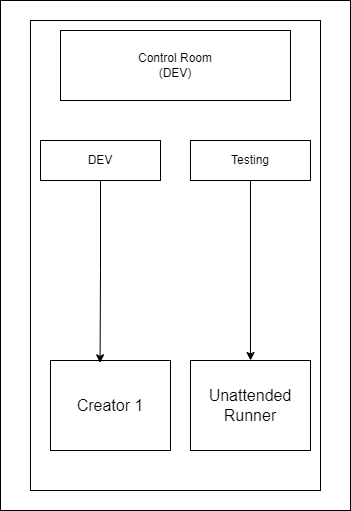
|  |  |
| --- | --- |
|  | **Bot Creator/ Runner** |
| **Operating System** | Windows 10 / 11 |
| **Hard Disk Space** | 100 GB or more free space |
| **RAM** | 8-16 GB |
| **System Resolution** | HD (1920 X 1080) and above. 1080p recommended |
| **Credential Management** | Part of Control Room |

|  |  |
| --- | --- |
| Application Pre-Requisites on all Creators | Application Pre-Requisites on all Runners |
| Automation Anywhere Bot Agent | Automation Anywhere Bot Agent |
| Microsoft Office (Excel) | Microsoft Office (Excel) |
| Email Access | Email Access |
| Website URL Access[https://www.trivago.in/] | Website URL Access[https://www.trivago.in/] |

**Environment Design:**

This process requires 1 Bot creator and runner to run the process.

There is a single environment that will be used for DEV, UAT



**Folder and File Structure:**

|  |  |
| --- | --- |
| Folder Directory | Description |
| C:\Bots\Hotel Management System\Log File\Audit Log Files | This folder will contain audit logs file for bot execution. |
| C:\Bots\Hotel Management System\Log File\Error Log Files | This folder will contain error logs file for bot execution. |
| C:\Bots\Hotel Management System\Input File | This folder is used to read input files. |
| C:\Bots\Hotel Management System\Output File | This folder is used for store the output file. |
| C:\Bots\Hotel Management System\Other File | This folder contains other files/readme file for bot execution. |

# **Non-Functional Requirements**

## **Security**

Bot requires an Email ID & password to send email to the client. There is a need to keep these details secure in a vault at the centralized location so that the bot runner can access these details. Also, any updates in these details required to be done can be done without making any changes to the Bot.

To fulfill this requirement Automation Anywhere provides a centralized vault named “Credential Vault” in the control room. Automation Anywhere Credential Vault is the secured storage of credentials of applications that are automated by the Automation Anywhere platform. Credential Vault allows adherence to enterprise-level information and security standards.

The Bot uses this Credential Vault having Email ID and Password as attributes to log in or access the required applications.

## **Availability Requirements**

BOT should be available 24x7 subject to the availability of all dependent systems. However, BOT gets triggered on time when the email has the subject “Hotel Booking System” & contains attachments received in the mailbox.

## **Volume and Performance Expectations**

As per information from the business, Volume, and performance expectations are given be below.

|  |  |  |
| --- | --- | --- |
|  | Parameters | Values |
| Volume | Emails received with attachments /Hour | Around 5-10 |
| Performance | Manual time to complete per request | 5-10 Mins |
|  | Expectations from bot per request | 1-2 |

# **Key Assumptions & Dependencies**

The BOT execution will be dependent on the following factors:

1. Enterprise A360 was installed successfully and has sufficient licenses for the control room, creator, and runner. Creator machines have a successful installation of A360.
2. Configuration of the machine on which BOT is running should be equal to or above recommended configuration.

The recommended configuration is

|  |  |  |  |
| --- | --- | --- | --- |
| **Processor** | **RAM** | **Storage** | **Network** |
| Intel Core i3 2.6 GHz with 4 multi-cores or higher.  64 Bit system | 8 GB Minimum,  16GB recommended | 32 GB Add 100 through 150 KB per Automation Anywhere script.  Add 40 through 50 GB per long-term project | 5Mbps (Minimum)  20Mbps or higher (Recommended) |

1. Availability of email credentials through Automation Anywhere credential vault.
2. The bot should have access to the email account.
3. A bot needs access to the local drive for downloading and performing file operations.
4. It is necessary for a bot to have access to Microsoft Excel tool installed on its machine
5. The system should have a stable internet connection during the time of bot execution.

The following assumptions are made while the development of this BOT:

1. The solution being proposed will run within a time interval of less than 10 minutes.
2. Any changes in SMTP or recipient email address have an impact on sending email after completion of Bot execution.
3. There are no filters applied to the Inbox.
4. Each file size should not exceed 200MB.
5. Emails are sorted by subject line and attachments are included.