IT ASSESSMENT OF IT SUPPORT SPECIALIST

Mehedi Hasan

Email: mailto:rkrefat01994@gmail.com

Note

Email: mehedi2trial@gmail.com

password:886633#Trial

Please access with these credentials in all of (Make.com, Zoho SalesIQ's) my works.

Task 1: Make.com Integration

Objective:

The goal of this automation process is to streamline calendar event creation based on incoming emails. The automation ensures data accuracy and handles incomplete or erroneous data effectively by routing errors and notifying the team promptly. This improves operational efficiency and reduces manual effort.

Planning:

Core goal is to automate the process of creating calendar events based on incoming emails while ensuring minimal errors and effective data handling. With accuracy in event creation, timely notifications for missing data, and robust error-handling mechanisms.

Requirement Analysis

- Email Data: Analyze the structure of incoming emails to identify:
- Key fields (e.g., subject line, date, consultant ID).
- Potential inconsistencies in data formatting.

System Integration

- Compatibility with Google Calendar for event creation.
- Use of Google Sheets for intermediate data handling and storage.
- Notification system integration for error reporting.

Workflow Overview

The automation consists of the following key steps, utilizing various modules to process, validate, and act upon incoming data.

Email Filtering (Gmail - Watch Emails**):**

The system monitors incoming emails and filters them based on specific subject keywords to ensure only relevant emails are processed.

Date Parsing and Formatting (Text Parser + Set Variable):

Extracts the date information from the email content. Converts the date into a Google Calendar-compatible format, as Google Calendar does not accept MMMM (e.g., "January") date formats directly. This step ensures the extracted date is usable for calendar event creation.

Regex Parsing for Data Extraction (Multi Regex Parser):

Parses the email content to extract key information such as: Consultant ID. Other essential details required for event creation.

Consultant Information Lookup (Google Sheets):

Uses the Consultant ID extracted in the previous step to retrieve the associated consultant data (e.g., name, email address) from a stored Google Sheet.

Email Address Parsing (Text Parser):

Extracts the consultant's email address from the retrieved data. This ensures that Google Calendar can send event invitations without errors, as ID-based data alone would not suffice for sending invitations.

Validation and Routing (Set Variable + Router):

A validation check is performed to ensure all required fields (e.g., date, consultant details) are filled with valid data.

Based on the validation:

Path 1 (Incomplete Data):

Logs the incomplete email data into a Google Sheet, including missing fields and timestamp. Sends an automated email to the team, notifying them of the issue with detailed error information for manual review and reprocessing.

Path 2 (Complete Data):

Passes validated data to the Google Calendar module to create the event.

Calendar Event Creation (Google Calendar):

Uses the validated data to create a calendar event. Sends invitations to the attendees (consultants and other participants).

Error Handling (Error Handler + Google Sheets):

If an error occurs during calendar event creation, the error handler captures the issue. Logs the error details into a Google Sheet and triggers an email to notify the team of the failure.

Detailed Process Breakdown

Step 1: Email Monitoring and Filtering

Module: Gmail (Watch Emails).

Functionality: Monitors incoming emails continuously. Filters emails by specific subject keywords to ensure only task-relevant emails are processed.

Step 2: Date Parsing and Formatting

Modules Used:

Text Parser: Extracts the date from email content.

Set Variable: Reformats the date into a compatible format for Google Calendar.

Purpose: Ensures smooth integration with Google Calendar by providing correctly formatted dates.

Step 3: Data Extraction via Regex Parsing

Module: Multi Regex Parser.

Functionality: Extracts structured data such as Consultant ID and other details required for downstream processing.

Step 4: Consultant Information Lookup

Module: Google Sheets.

Functionality: Uses Consultant ID to fetch detailed consultant information (e.g., name, email address).

Step 5: Email Address Extraction

Module: Text Parser.

Purpose: Ensures the consultant's email address is extracted for accurate calendar invitations.

Step 6: Validation and Routing

Modules Used: Make built in tool.

Set Variable: Checks for missing fields.

Router: Routes data based on validation results.

Functionality: Path 1 (Incomplete Data): Logs missing fields and notifies the team via email for manual intervention.

Path 2 (Complete Data): Proceeds to calendar event creation.

Step 7: Calendar Event Creation

Module: Google Calendar.

Functionality: Creates calendar events using validated data. Sends invitations to attendees.

Step 8: Error Handling

Modules Used:

Error Handler: Captures errors occurring during calendar event creation.

Google Sheets: Logs error details.

Gmail: Sends email notifications to the team about the error.

Business Value:

Improved Efficiency: Automates repetitive tasks, reducing manual effort and time consumption.

Error Mitigation: Captures and routes errors effectively, minimizing disruptions to operations.

Scalability: Designed to handle large volumes of data without manual intervention.

Transparency: Logs all errors and missing data for accountability and future improvements.

Time Log for Task Completion (10 Hours)

Day 1: Planning and Initial Setup (5 Hours)

Hour 1 (9:00 AM – 10:00 AM): Requirements analysis and task breakdown.

Hour 2 (10:00 AM – 11:00 AM): Workflow mapping and identification of key modules (email parsing, data handling, router logic).

Hour 3 (11:00 AM – 12:00 PM): Research and configuration of email parser and data extraction logic.

Hour 4 (12:00 PM – 1:00 PM): Initial testing of email-to-Google Sheets integration and troubleshooting parsing issues.

Hour 5 (1:00 PM – 2:00 PM): Refinement of parsed data formatting to align with Google Calendar requirements (e.g., date conversion).

Day 2: Implementation and Testing (5 Hours)

Hour 6 (9:00 AM – 10:00 AM): Implementation of router logic for handling missing and complete data cases.

Hour 7 (10:00 AM – 11:00 AM): Integration of Google Calendar for event creation and invitations.

Hour 8 (11:00 AM – 12:00 PM): Testing complete workflow with simulated email data (complete, missing, and erroneous).

Hour 9 (12:00 PM – 1:00 PM): Development of error-handling mechanisms (logging to Google Sheets and email notifications).

Hour 10 (1:00 PM – 2:00 PM): Final system validation and preparation of documentation for workflow and implementation.

Total Time Spent: 10 Hours

This time log is based on my free time that I could manage with my current work life.

Task 2: Virtual Environment Setup

Objective:

Create a virtual machine (VM)-based environment where agents can log in remotely to run property analysis software. The environment must:

- Prevent agents from accessing the host machine outside their allocated VM.
- Ensure that 1 to 5 agents can simultaneously access individual VMs securely.
- Use Windows-compatible software (Windows 7 or later).

Tools/Software Needed

Host Machine Software:

Hypervisor Software:

- VMware Workstation Pro (recommended for robust features).
- Oracle VirtualBox (free alternative).

Operating System for Host Machine:

- Windows 11 Pro (preferred for compatibility with advanced features like Hyper-V).

Agent Machines:

Remote Desktop Protocol (RDP) Client:

- Native Microsoft Remote Desktop (Windows or macOS).
- Alternative: AnyDesk or TeamViewer.

VPN Software:

- OpenVPN (to securely connect to the host machine).

Additional Tools:

- Windows Remote Desktop Services (RDS): For managing multiple RDP connections.
- Monitoring Tool: e.g., SolarWinds RMM for tracking VM performance.

Best-Suited Windows Edition for VMs

Windows 10 Pro or Windows 11 Pro:

- Lightweight and stable for property analysis software.
- Supports Remote Desktop and Hyper-V features.

Host Machine Specifications

To run 5 simultaneous VMs efficiently:

- Processor: Intel Core i7 (12th Gen or higher) or AMD Ryzen 7.
- RAM: Minimum 32 GB (6 GB allocated per VM). Note- 64GB will be the very secure option but will be added cost so the best valued approach is considered here.
- Storage: 1 TB SSD (to store VM disk files and ensure fast performance).
- GPU: Dedicated GPU (e.g., NVIDIA GeForce RTX 3060) if analysis software requires graphics.
- Network Connectivity: Gigabit Ethernet port for stable RDP connections.

VM workstation building guide

Step 1: Prepare the Host Machine

- Install Windows 11 Pro on the host laptop.
- Update all drivers and ensure the machine is secure with the latest OS updates.
- Install Hypervisor Software (VMware Workstation Pro or VirtualBox).
- Configure a separate admin account for managing the VMs.

Step 2: Create Virtual Machines

Launch the hypervisor software.

Create a new VM template:

- OS: Install Windows 10 Pro/Windows 11 Pro.

  Allocate resources:

  - RAM: 4–6 GB.

- Storage: 60–80 GB.
- Processors: 2 cores.

Configure network settings to use NAT or Bridged Networking.

Install the property analysis software on the VM.

Secure the VM:

- Enable password protection for the user account.
- Restrict guest OS access to host resources.
- Clone the VM template to create 5 instances for agents.
- Assign unique names (e.g., "AgentVM1," "AgentVM2").

Step 3: Configure Network and Remote Access

Set up VPN Software on the host machine:

- Use OpenVPN to create secure tunnels for remote connections.

Assign static IPs for each VM.

Enable Remote Desktop on each VM:

- Navigate to System Settings > Remote Desktop > Enable Remote Access.

Create unique RDP credentials for each agent and share securely.

Step 4: Restrict Host Machine Access

Use hypervisor settings to isolate VMs from the host:

- Disable shared folders and USB passthrough.

Set permissions to prevent agents from accessing the hypervisor directly.

Step 5: Testing and Validation

- Perform a test login for each VM using RDP to ensure remote access works.
- Check that multiple agents can log in simultaneously without conflicts.
- Validate that agents cannot access resources outside their assigned VM.

Step 6: Monitoring and Maintenance

Use a monitoring tool (e.g., SolarWinds RMM) to:

- Track CPU, RAM, and network usage of each VM.
- Monitor uptime and address performance issues.

Regularly update the OS and software on both the host and VMs.

Back up VM templates and critical data weekly.

How to Set Up and Access Virtual Machines for Property Analysis-

Purpose: This document outlines how agents can securely access virtual machines for running property analysis software.

Steps for Agents:

Set Up VPN:

- Download the OpenVPN client.
- Import the configuration file provided by the admin.
- Connect to the VPN.

Log in to the Virtual Machine:

- Open the RDP client on your device.
- Enter the IP address and credentials for your assigned VM.
- Click "Connect" to access your virtual desktop.

Using the VM:

- Run the pre-installed property analysis software.
- Save all work within the VM's allocated drives.

Logout Safely:

- Close the RDP session when finished.

Troubleshooting:

- Can’t connect to the VPN: Ensure your internet is stable and retry.
- VM is slow: Notify IT support for resource reallocation.
- Forgot credentials: Contact the admin for assistance.

Task 3: (AI Chatbot)

Objective:

The main goal of this task is to reduce the workload on the Tier-1 support team by implementing an AI chatbot on the company intranet. This chatbot will handle common queries, enabling support teams to focus on more complex issues. The objectives include:

1. **Researching AI Chatbot Platforms**: Identifying tools that meet the business requirements while being cost-effective and scalable.
2. **Providing a Comparative Summary**: Highlighting the differences in features, costs, and pros/cons of popular chatbot platforms.
3. **Recommending the Best Platform**: Choosing one chatbot platform and defending the choice based on its suitability for Tier-1 support needs.
4. **Defining Core FAQs and Queries**: Establishing the common Tier-1 queries the chatbot should handle.
5. **Designing a Flowchart**: Outlining the user interaction flow with the chatbot.
6. **Proposing Key Metrics**: Identifying measurable outcomes to evaluate the chatbot's performance.
7. **Creating a Demo**: Demonstrating a basic prototype or UI flow for the chatbot.

Comparative Summary of AI Chatbot Platforms

Aspect	Microsoft Power Virtual Agents	Google Dialogflow	Rasa Open Source	Zoho SalesIQ's Zobot
Initial Cost	Free (Copilot Chat)	Free (within usage limits)	Free	Free (limited features)
Advanced Features	$30/user/month	Pay-as-you-go	Development resources required	Available in paid plans
Integration	Seamless with Microsoft 365	Integrates via APIs	Fully customizable	Integrates with Zoho apps and third-party apps
Scalability	High	High	Depends on infrastructure	Suitable for SMBs; scalable with plans
Maintenance	Low	Low	High (self-managed)	Low (cloud-based)

Recommendation: Zoho SalesIQ's Zobot

Rationale

Zoho SalesIQ's Zobot is the most suitable platform for Tier-1 support needs, especially for small to medium-sized businesses (SMBs) like St Trinity Property Group. Here's why:

1. **Cost-Effectiveness**:
    - Offers a free plan with essential features, making it budget-friendly.
    - Advanced capabilities can be unlocked with paid plans if needed.
2. **Ease of Integration**:
    - Seamlessly integrates with Zoho's suite of tools, which may already be in use by the organization.
    - Supports third-party app integrations, enabling flexibility.
3. **Low Maintenance**:
    - Being cloud-based, it requires minimal maintenance, freeing up IT resources.
4. **Scalability**:
    - Well-suited for SMBs and offers scalability through its flexible pricing tiers.

5. **Feature Set**:
    - Includes capabilities such as natural language processing (NLP), predefined workflows, and analytics to enhance Tier-1 support efficiency.

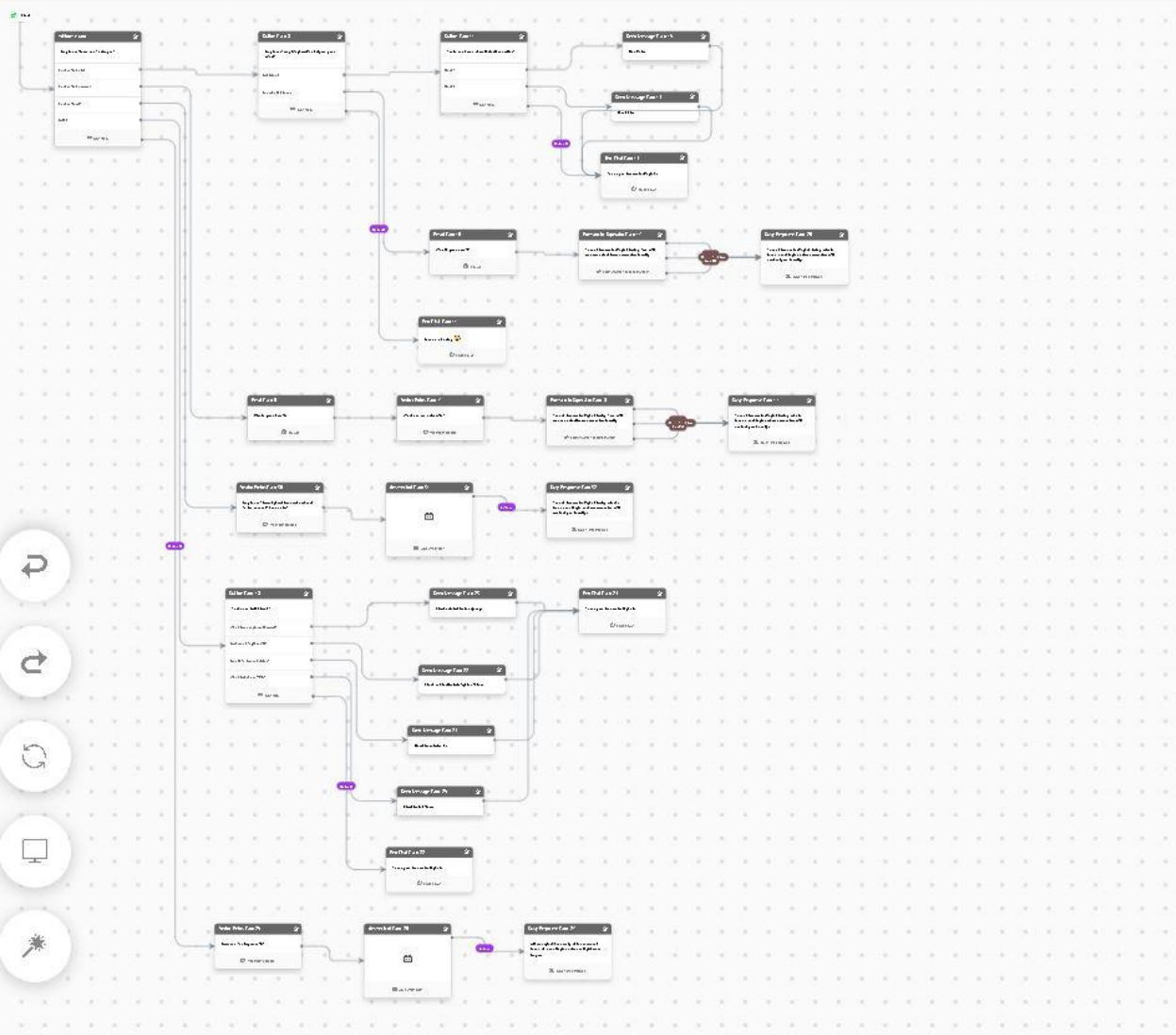
Recommendation for Tier-1 Support

- **Tier-1 Support Scope**: Zobot can efficiently handle repetitive Tier-1 queries, such as password resets, VPN setups, and email configuration issues.
- **Automated Responses**: It uses NLP to understand user intent and provide accurate responses.
- **Customization**: Workflows can be tailored to the organization's needs, ensuring that the chatbot aligns with business processes.
- **Fallback Mechanism**: In case of unresolved queries, Zobot can escalate to human agents, ensuring seamless support.

Proposed Core FAQs and Queries for Tier-1 Support

1. **Password Management**:
    - How do I reset my password?
    - I'm locked out of my account. What do I do?
2. **VPN Setup**:
    - How do I configure the VPN on my device?
    - Why can't I connect to the VPN?
3. **Email Support**:
    - How do I set up my work email on my phone?
    - Why am I not receiving emails?

Diagram of the Chatbot Design



Key Metrics to Track Chatbot Performance

1. **Resolution Rate**: Percentage of queries resolved without human intervention.
2. **Response Time**: Average time taken to respond to user queries.
3. **User Satisfaction**: Feedback rating provided by users after interactions.
4. **Escalation Rate**: Percentage of queries escalated to human agents.
5. **Utilization Rate**: Number of interactions handled by the chatbot over a given period.

Demo build of the chatbot:

Please access the demo chatbot here.

Please look at the cover page for all the credential details.

References

1. For automation tasks and queries: https://www.make.com/en/blog/ai-and-automation
2. ChatGPT Basic queries on Parsing with Reg Ex.

   Prompt: How do you filter out specific string of date>

   Now that the date is parsed which approach does offer the flexibility of the user input. Skip Case sensitive part.