TITLE OF PROJECT REPORT

AUTOMATED RESUME CLASSIFIER AND DAILY REPORT BOT

A PROJECT REPORT

Submitted by

SAKTHI SHALINI R

in partial fulfillment for the course

OAI1903 - INTRODUCTION TO ROBOTIC PROCESS AUTOMATION

for the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR

THANDALAM

CHENNAI – 602 105

NOVEMBER 2024

RAJALAKSHMI ENGINEERING COLLEGE

CHENNAI - 602105

BONAFIDE CERTIFICATE

Certified that this project report "…….TITLE OF THE PROJECT……..." is the bonafide work of "……..NAME OF THE CANDIDATE (REGISTER NO.)……" who carried out the project work for the subject OAI1903-Introduction to Robotic Process Automation under my supervision.

Dr. N.Durai Murugan
SUPERVISOR
Associate Professor
Department of Computer Science And Engineering
Rajalakshmi Nagar
Thandalam
Chennai - 602105

Submitted to Project and Viva Voce Examination for the subject OAI1903-Introduction to Robotic Process Automation held on _____.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Thiru. S.Meganathan, B.E., F.I.E.,our Vice Chairman Mr. M.Abhay Shankar, B.E., M.S., and our respectedChairperson Dr. (Mrs.) Thangam Meganathan, M.A., M.Phil., Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N.Murugesan, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr. P.Kumar, M.E., Ph.D., Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guides, Ms. Roxanna Samuel,M.E., Assistant Professor (SG), Ms. U.Farjana, M.E., Assistant Professor andMs. S.Vinothini, M.E., Department of Computer Science and Engineering for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinators, Dr. P.Revathy, M.E., Ph.D., Professor,Dr. N.Durai Murugan, M.E., Ph.D., Associate Professor, andMr. B.Bhuvaneswaran, M.E., Assistant Professor (SG), Department ofComputer Science and Engineering for their useful tips during our review to build our project.

SAKTHI SHALINI R
220701241

TABLE OF CONTENTS

ABSTRACT

The "Automated Resume Sorting and Daily Reporting System" is a cutting-edge solution designed to revolutionize how resumes are managed within HR departments. The growing volume of job applications presents significant challenges to HR teams, as manually sorting resumes and preparing reports is time-intensive and prone to errors. This system, developed using UiPath Studio, leverages Robotic Process Automation (RPA) to automate the classification of resumes based on predefined skill sets and the generation of daily reports. It ensures a seamless, efficient, and accurate workflow for managing recruitment data.The system automates a multi-step process starting with scanning a folder containing resumes. It reads each file, identifies specific keywords such as "Java," "Python," and "C," and moves the files into corresponding subfolders. Simultaneously, it records details of the processed files in an Excel sheet, noting the file name and the category to which it was assigned. At the end of the process, the system sends the daily report to HR via email, ensuring timely delivery of information critical to recruitment decisions.The workflow begins by initializing key variables such as folder paths for resumes and destination categories. The bot verifies the existence of subfolders for each skill category ("Java," "Python," and "C") and creates them if necessary. It then processes each file in the folder, utilizing text reading and keyword matching to classify the resumes accurately. Based on the detected keywords, the bot dynamically moves the files into the respective folders. The bot simultaneously updates a structured Excel sheet to log the sorting results, ensuring traceability and transparency.One of the standout features of this system is its ability to generate and share detailed daily reports automatically. The Excel sheet created during the process contains key information, such as the name of each resume file and its corresponding category. Using UiPath's email automation capabilities, the system attaches this report to an email and sends it to HR teams, ensuring they have updated records without manual intervention.

The benefits of this system are substantial. By automating repetitive tasks, it significantly reduces the time and effort HR professionals spend on administrative work. Its keyword-based classification ensures consistent and error-free sorting, minimizing the risk of misplacing critical resumes.

LIST OF TABLES

Features of the “Automated Resume Classifier and Daily Report Bot"

Feature	Description	Purpose
Resume Classification	The bot reads resumes from a folder and classifies them based on specific keywords (e.g., Java, Python, C). Files are moved to categorized subfolders accordingly.	Automates the process of sorting resumes based on skills, ensuring efficient management of incoming applications.
Keyword-Base Sorting	The bot scans resumes for specific keywords related to programming languages or skills (e.g., Java, Python, C)	Ensures resumes are placed in the correct folder based on the skillset mentioned, enhancing recruitment accuracy.
Automated Data Logging	As resumes are processed, details such as the file name and the assigned folder name are logged into an Excel sheet.	Creates a record of processed resumes, ensuring transparency and easy tracking of classified applications.
Email Report Generation	The bot generates a daily Excel report summarizing the resume classification process. The report includes file names and their corresponding categories.	Automates the generation of daily reports, ensuring timely updates for HR regarding processed resumes.
Email Notification to HR	After the report is generated, the bot sends the daily report to HR via email, attaching the Excel file.	Ensures that HR receives an updated and detailed resume classification report without manual intervention.

Workflow Activities of the Automated Resume Classifier and Daily Report Bot"

Step No	Activity	Description	Tools/Features Used
1	Read Resumes from Folder	The bot scans the designated folder for all resume files and stores their paths for further processing.	Directory.GetFiles, For Each Activity
2	Read Resume Content	The bot reads the content of each resume file to search for specific keywords (Java, Python, C).	Read Text File Activity
3	Check for Keywords	The bot checks if the resume content contains any of the pre-defined keywords (Java, Python, C).	If Activity, Contains Function
4	Move Resume to Appropriate Folder	Based on the keyword found, the bot moves the resume to the corresponding folder (Java, Python, C).	Move File Activity, Path.Combine
5	Log Classification Data	After classifying the resume, the bot logs the file name and the folder it was moved to into an Excel sheet.	Add Data Row Activity, Excel Application Scope
6	Generate Daily Report	The bot generates a report summarizing the classified resumes, including file names	Write Range Activity, Excel Application Scope
7	Send Report to HR via Email	After the daily report is generated, the bot sends it to HR via email with the Excel report as an attachment.	Send SMTP Mail Message
8	Log Process Completion	The bot logs the completion of the process, including success and failure messages, for monitoring and debugging purposes.	Log Message Activity.

LIST OF FIGURES

Introduction

In today's fast-paced recruitment landscape, managing and sorting through the growing number of resumes can be a time-consuming and labor-intensive task. Human Resources (HR) departments often face challenges in efficiently handling, categorizing, and tracking candidate resumes, especially when there are large volumes to process. To streamline this process and ensure greater accuracy and efficiency, automation is becoming an essential solution.The **Automated Resume Classifier and Daily Report Bot** is a powerful tool designed to automate the classification of resumes based on specific skill sets and to generate daily reports summarizing the classification process. This bot uses predefined keywords such as programming languages (e.g., Java, Python, C) to sort resumes into categorized folders, eliminating manual sorting and the possibility of human error. Additionally, it automates the process of generating and sending daily reports to HR, saving time and ensuring that relevant stakeholders are consistently updated with the latest resume classifications.This bot operates through a series of automated steps, from reading resumes, identifying the skill set keywords, sorting them into appropriate folders, logging the results into an Excel sheet, to sending daily email reports. The use of automation tools like UiPath ensures the process is fast, efficient, and free from human oversight, allowing HR teams to focus on more strategic tasks rather than routine administrative ones.

1.1 General

The **Automated Resume Classifier and Daily Report Bot** is an advanced automation tool designed to simplify the recruitment process by automating the classification of resumes based on predefined criteria and generating detailed reports. In today's recruitment industry, organizations often face the challenge of managing large volumes of resumes from job applicants. Manually sorting these resumes to identify candidates with specific skills can be time-consuming, prone to human error, and inefficient.This bot uses **UiPath**, a leading Robotic Process Automation (RPA) platform, to automate the entire process. It is specifically designed to read resumes from a designated folder, classify them into categories based on keywords like programming languages (e.g., Java, Python, C), and move them into respective folders (Java, Python, and C). Additionally, the bot logs the classification results into an Excel file and generates a daily report summarizing the activity, which is then automatically emailed to the HR department for review

1.2 Objectives

1.3 Existing System

1. Resume Collection and Storage

The process begins when resumes are collected through various channels such as job portals, emails, career websites, and direct submissions. These resumes are typically stored in a centralized folder or directory, where they are accessible by HR staff for processing. However, this collection process can be chaotic, especially when resumes come from multiple sources in different formats (PDF, Word, etc.). HR professionals have to ensure that the resumes are appropriately saved and stored in a location where they can be easily retrieved.

2. Manual Resume Sorting and Categorization

Once the resumes are collected, HR personnel manually sort and classify each resume based on specific skills, qualifications, or job requirements. For example, resumes may be sorted into different folders depending on programming languages (such as Java, Python, or C) or other job-specific criteria. This sorting is often done by reviewing the content of each resume manually, which is both time-consuming and prone to human error. The process involves assessing a candidate's qualifications, skills, years of experience, and education, among other factors.

This manual approach to classification leads to inefficiencies, particularly when there are a large number of applications. HR professionals often need to spend hours reviewing and categorizing resumes, which delays the recruitment process. Additionally, there is a risk that qualified candidates might be overlooked due to the subjective nature of manual classification or due to errors made during the sorting process.

3. Keyword-based Filtering

In the existing system, HR teams often use specific keywords or phrases to filter resumes based on the required job skills, such as programming languages (Java, Python, C), educational qualifications, or professional certifications. However, this process is still manual, requiring HR staff to manually search for and identify these keywords within each

resume. This method can sometimes result in misclassification, particularly if candidates do not use the exact keywords or phrasing that HR professionals are looking for.

Additionally, resumes that mention multiple skills or have keywords in a non-standard format may be overlooked, leading to the exclusion of potentially qualified candidates.

4. Report Generation and Data Entry

After categorizing and sorting the resumes, HR teams typically record information from the resumes into an Excel spreadsheet or database. This process involves manually entering data such as the candidate's name, job role, skills, and experience. Once the data is recorded, HR staff generate reports based on this information, often summarizing the number of resumes received, categorized, and stored.

The generation of these reports is another area where manual effort is required. HR teams often have to prepare daily or weekly reports that are shared with hiring managers or team leads. These reports are typically in Excel format and contain detailed data on the resumes categorized by job skills, making them useful for tracking progress, but also time-consuming to create.

5. Communication and Email Notifications

Once the reports are prepared, HR staff manually email the reports to hiring managers, recruiters, or other stakeholders. This process may involve attaching the reports in Excel format or providing updates through email communication. The lack of automation in this step means that HR personnel are responsible for sending individual emails to each recipient, which can become repetitive and inefficient, particularly when sending reports to multiple people daily.

1.4 Proposed System

The proposed **Automated Resume Classifier and Daily Report Bot** is an automated solution designed to address the limitations of existing systems by streamlining subscription management and web data extraction processes. This system is built using **UiPath Studio**, a powerful tool for **Robotic Process Automation (RPA)**, to ensure seamless and efficient automation of tasks that are typically manual and error-prone.

Key Features of the Proposed System:

1. **Automated Subscription Tracking:** The Subscription Tracking Bot will automatically track the renewal dates of all subscriptions listed in an Excel sheet. It will calculate the difference between the current date and the next renewal date. If the difference is less than three days, the bot will automatically send email reminders to the subscribers, ensuring that no subscription is missed and that reminders are sent on time. This feature eliminates the risk of human error and significantly reduces manual workload.
2. **Web Scraping for Data Extraction:** The system will include an integrated **web scraping** function that automatically collects valuable data from e-commerce websites. This includes data on the most-sold products, frequently bought-together items, and overall product reviews. The bot will extract this data in real-time and store it in an organized format within an Excel sheet for easy analysis and tracking.
3. **Daily Report Generation:** Once the data is collected through web scraping, the bot will generate daily reports in **DOC format**. These reports will summarize the key insights from the collected data, such as sales trends, popular products, and customer reviews, providing businesses with actionable insights to refine inventory management and marketing strategies.
4. **Email Notification System:** In addition to the subscription reminders, the system will be capable of sending notifications regarding the status of the web scraping and the generated reports. This ensures that businesses are kept informed in real-time about both the subscription renewals and the collected data.
5. **Scalability and Flexibility:** The system will be designed with scalability in mind. It will be capable of handling increasing numbers of subscriptions and scraping data from multiple websites, making it ideal for growing businesses. The automation will remain efficient even as the workload expands.
6. **Error-Free Automation:** The use of UiPath Studio ensures that the automation is error-free, with tasks being performed in a consistent and predictable manner. The bot will handle all subscription tracking, data extraction, and report generation without requiring human intervention, allowing employees to focus on more strategic tasks.
7. **Integration with Existing Systems:** The proposed system can easily integrate with other business systems, such as CRM or customer support platforms, to provide a unified approach to managing subscriptions and customer data.

Benefits of the Proposed System

Efficiency: The automated system significantly reduces the time spent on manual resume classification, sorting, and data entry.

Accuracy: The system uses NLP techniques to ensure that resumes are accurately classified based on relevant skills and job requirements.

Scalability: The system can easily handle an increasing number of resumes, making it suitable for both small and large-scale hiring efforts.

Real-Time Access: Hiring managers and HR professionals will have access to real-time data, enabling faster decision-making.

Improved Communication: Automated email notifications ensure that stakeholders receive timely updates on recruitment progress.

Error Reduction: By automating processes, the system reduces human error, ensuring consistent and accurate resume categorization.

2. Literature Review

The field of **Robotic Process Automation (RPA)** has seen significant advancements in recent years, particularly in automating business processes such as subscription management, data collection, and reporting. This section explores existing research, methodologies, and tools related to RPA, subscription management systems, and web scraping technologies. It aims to provide a comprehensive understanding of the current state of these technologies and their applications in subscription tracking and data extraction.

2.1 Robotic Process Automation (RPA) in Business Operations

RPA is the use of software robots or "bots" to automate repetitive tasks typically performed by human workers. According to **Avasarala et al. (2021)**, RPA is revolutionizing business operations by improving efficiency, accuracy, and scalability.

Bots can execute a range of tasks, including data entry, data extraction, and report generation, which would otherwise be performed manually. The key advantages of RPA, such as cost reduction and error minimization, have made it a preferred solution in industries like finance, healthcare, and customer service.

Several studies, including those by **Michael and Tiwari (2019)**, demonstrate that RPA tools like **UiPath**, **Blue Prism**, and **Automation Anywhere** have been successfully applied in a variety of business operations, including subscription management. These tools allow businesses to automate tasks such as invoice processing, order management, and subscription tracking. Automation of subscription management, as noted by **Gupta and Yadav (2020)**, helps businesses reduce the risk of human error and ensure timely reminders for renewals, which improves customer retention.

2.2 Subscription Management Systems

Subscription-based business models are becoming increasingly popular across various industries, including media, software, and e-commerce. According to **Hoch et al. (2020)**, subscription management systems allow businesses to track renewals, manage billing cycles, and ensure customer satisfaction through timely communications. Traditional subscription management systems involve manual data tracking, which can be inefficient and prone to mistakes, especially as subscription volumes increase.

Ahrens (2021) suggests that automating the tracking of subscription renewals and notifications can significantly improve operational efficiency. RPA-based solutions, like the one proposed in this project, have the potential to enhance subscription management by integrating with existing systems to track due dates and automatically notify customers of upcoming renewals.

Moreover, **Sadeghi et al. (2022)** explore the concept of intelligent subscription management systems that not only track renewals but also offer insights into customer preferences and behaviors. These systems can provide businesses with more granular data to help optimize their subscription models and customer engagement strategies. Integrating such intelligence with RPA can further enhance the functionality and effectiveness of the system.

2.3 Web Scraping and Data Extraction Technologies

Web scraping has emerged as a powerful tool for data collection, enabling businesses to gather real-time information from websites for analysis and decision-making. **Zhang et al. (2018)** provide an overview of various web scraping techniques used for data extraction from websites, including parsing HTML and XML documents, utilizing APIs, and employing automation tools like Selenium. In recent years, web scraping has become a popular method for businesses to collect product-related data, including sales trends, reviews, and frequently bought items.

According to **Almeida et al. (2020)**, web scraping tools can be integrated with RPA solutions to automate the process of extracting data from e-commerce platforms. This is particularly valuable for businesses that rely on real-time data to monitor market trends and customer behavior. The use of RPA in conjunction with web scraping allows for continuous, automated data extraction and analysis, reducing the need for manual intervention.

Additionally, **León et al. (2019)** highlight the importance of data accuracy when using web scraping for business applications. They emphasize the role of regular updates and error handling in ensuring that scraped data remains relevant and accurate. In the proposed system, automated web scraping will allow businesses to gather up-to-date information on product sales and customer feedback, which will be vital for creating daily reports and making informed business decisions.

2.4 Report Generation and Automation

Automated report generation has become an essential feature of modern business intelligence tools. **Adams et al. (2018)** describe how report automation software can significantly reduce the time and resources required for generating accurate business reports. These tools often integrate with existing business systems, allowing for the automatic generation of reports based on live data.

In the context of subscription management and web scraping, **Singh et al. (2021)** suggest that generating reports automatically can provide businesses with timely insights without requiring manual data processing. Automated reports can summarize key metrics such as

sales performance, customer behavior, and product reviews, helping businesses track their progress and make quick adjustments to their strategies.

The proposed will incorporate a report generation feature that automatically compiles scraped data and subscription renewal information into daily reports. These reports will be delivered in **DOC format** and will highlight critical insights such as popular products, customer purchasing trends, and subscription renewal statuses. Automating this process will enable businesses to make data-driven decisions without the need for manual report generation.

2.5 Challenges and Opportunities

Despite the benefits of RPA and web scraping, implementing these technologies in business operations presents certain challenges. **Pereira and Mendes (2022)** highlight the complexities involved in setting up RPA systems, including the need for careful process mapping, integration with existing software, and employee training. In the case of subscription tracking, ensuring the accuracy of renewal data and integration with customer management systems can pose difficulties.

Web scraping also presents challenges, such as handling anti-scraping mechanisms deployed by websites, ensuring compliance with data privacy regulations, and managing large volumes of data. **Martinez and Sandoval (2020)** discuss these challenges and suggest the use of error handling techniques and adaptive scraping algorithms to address issues such as CAPTCHAs and data inconsistencies.

Despite these challenges, the growing demand for automation in business operations presents significant opportunities for RPA and web scraping technologies. The ability to automate subscription tracking and web data collection can lead to enhanced operational efficiency, improved customer engagement, and data-driven decision-making.

2.6 Conclusion

In conclusion, RPA, subscription management systems, and web scraping technologies have the potential to greatly enhance business operations, making them more efficient, accurate, and scalable. The proposed **Automated Resume Classifier and Daily Report Bot** combines these technologies to automate the tracking of subscription renewals, data collection from e-commerce websites, and report generation. By reviewing existing

literature, it is evident that the integration of RPA with web scraping and subscription management systems can offer significant benefits to businesses, improving both operational efficiency and customer satisfaction. The proposed system will build on these advancements to deliver a fully automated, error-free, and scalable solution.

3. SYSTEM DESIGNS

The System Design section outlines the architecture, components, and the design approach for the Automated Resume Classifier and Daily Report Bot. The system is built using UiPath Studio, leveraging RPA to automate the subscription tracking and web scraping processes. This section provides a detailed overview of how the system components interact with each other to ensure smooth and efficient functioning.

3.1 GENERAL

3.1.1. System Architecture

User Interface Layer (UI)

- **Purpose**: The UI layer allows HR personnel or administrators to configure the bot, monitor its status, and interact with reports. Users can adjust the bot's settings, set folders for resume processing, and configure the frequency of report generation.
- **Key Features**:
    - Configuration settings (input folders, output reports, classification criteria)
    - Viewing logs and system performance
    - Triggering manual report generation or process runs

2. Data Ingestion Layer

- **Purpose**: This layer handles the collection and monitoring of resumes. The bot can fetch resumes from cloud storage (e.g., Google Drive), email inboxes, or local file systems.
- **Key Features**:
    - Monitors specific directories for new resume files
    - Supports multiple resume formats (PDF, DOCX, TXT)
    - Collects resumes from cloud-based storage or local sources

3. Data Classification Layer

- **Purpose**: The core of the system that performs resume classification based on specific keywords and job-related criteria (such as skills and experience). This layer uses NLP (Natural Language Processing) and predefined keywords to categorize resumes into relevant folders (e.g., Java, Python, C).
- **Key Features**:
    - Resume parsing and content extraction (skills, education, experience)
    - Classification of resumes into categories based on skill matching (Java, Python, etc.)
    - Keyword-based classification engine

4. Reporting Module

- **Purpose**: This module compiles daily summary reports, which include key insights from the resume classification process. It formats and generates reports in DOC, PDF, or Excel format, summarizing the daily activities and categorized data.
- **Key Features**:
    - Generates daily summary reports
    - Formats reports in DOC, PDF, or Excel
    - Includes statistics like the number of resumes classified per category, top skills identified, and trends

5. Email Notification System

- **Purpose**: This layer is responsible for sending the generated reports via email to HR professionals or stakeholders. It uses SMTP protocol to send reports as attachments in the desired format.
- **Key Features**:
    - Sends automated daily emails with generated reports
    - Customizable email templates (subject, body)
    - Sends reports as DOC, PDF, or Excel attachments

3.1.2. Functional Design

The Automated Resume Classifier and Daily Report Bot is designed to automate the process of classifying resumes based on specific skills (e.g., Java, Python, C) and generating daily reports

Resume Folder and File MonitoringPurpose: To ensure the system operates on a regularly updated set of resumes and that files are processed correctly.

1. **Folder Setup**: The bot monitors a folder where resumes are saved. Subfolders are created for each skill set (Java, Python, C) to ensure proper classification.
2. **Input Source**: The system continuously checks for newly added resumes in the specified directory

Resume Content Analysis and Classification

Purpose: To identify the presence of certain keywords (Java, Python, C) in resumes and classify them into respective folders

Activities

For Each File: The bot loops through each resume in the monitored folder.

Read Text from Resume: The bot extracts text from each resume (using Optical Character Recognition (OCR) or simple text extraction techniques, depending on the file format).

3. **Data Logging and Record Keeping**
4. **Purpose**: To maintain a record of all resumes processed, including classification results, and prepare data for reporting.

Activities:

Create Data Table: A data table is initialized to store the resume name, classification folder, and status (processed, pending, etc.).

Add Data Row: Each resume's classification result (including the folder name it was moved to) is added as a new row in the data table.

Store Data: The data is saved in an Excel sheet, which will later be used to generate a report.

---

4. Excel Report Generation

Purpose: To generate daily reports summarizing the classification results of all resumes processed during the day.

Activities:

- **Write Data to Excel**: At the end of each day, the bot writes the collected data (resumes and their classifications) into a designated Excel file.
    - The Excel file has columns such as:
        - **File Name**
        - **Folder Name (Java/Python/C)**
        - **Date Processed**
        - **Status**
- **Format Report**: The Excel file is formatted for easy readability (e.g., column headers, date formatting).

Email Notification and Report Delivery

Purpose: To automate the process of sending the daily report to HR or relevant stakeholders via email.

Activities:

- **Send Email**: The bot sends an email to the designated recipients (e.g., HR) with the daily report attached as an Excel file.
    - The email includes a brief summary of the day's resume classification activity.
    - The Excel file with the results is attached for further review.

Error Handling and Logging

Purpose: To ensure the bot runs smoothly by identifying and addressing any errors or issues during execution.

Activities:

- **Log Errors**: If the bot encounters an issue (e.g., unable to read a file, or no keywords found), an error message is logged.
- **Manual Review**: Resumes that cannot be automatically classified are flagged for manual review and added to a separate folder for follow-up.
- **Notifications**: In case of critical errors (e.g., failure to send the email), the bot triggers an alert or notification to inform the user about the issue.

Scheduling and Automation

Purpose: To automate the execution of the bot on a regular basis.

Activities:

- **Task Scheduler**: The bot is scheduled to run at the same time each day (e.g., at midnight or after business hours) using Windows Task Scheduler or Orchestrator.
    - This ensures that the bot runs without manual intervention, keeping the process automated and consistent.

Reporting and Insights Generation

Purpose: To provide stakeholders with detailed insights about the resume processing activity and trends.

Activities:

- **Daily Insights**: The bot generates a daily summary of all processed resumes, including the number of resumes classified under each skill set.
    - For example: "Today, 50 resumes were classified into the Java folder, 30 into Python, and 20 into C."
- **Trends Analysis**: Over time, the bot can generate reports analyzing trends (e.g., more resumes are coming in for Java compared to Python, etc.).

Data Backup and Storage

Purpose: To ensure the data generated is safely stored and accessible for future analysis or reference.

Activities:

- **Backup Reports**: The daily Excel reports are backed up to a secure cloud storage or local directory for long-term retention.
- **Data Archiving**: Older reports are archived periodically to maintain a manageable file size and ensure quick access to recent data.

3.1.2.3. Technical Design

Below is the detailed technical design of the bot based on the chosen tools and technologies:

1. UiPath Studio

Role in the System: UiPath Studio is used as the primary tool to design, develop, and automate the workflows for resume classification and daily report generation.

Key Features:

- **Drag-and-Drop Interface**: UiPath provides a user-friendly interface for designing automation workflows without requiring extensive coding.

- **Prebuilt Activities**: The tool comes with built-in activities such as 'Read Text File,' 'Move File,' 'Send SMTP Mail Message,' and 'Write Range' to handle the automation processes seamlessly.
- **Error Handling**: UiPath's robust exception handling framework ensures that any errors or exceptions in the workflow are caught and logged for analysis.

---

2. Microsoft Excel

Role in the System: Excel is used to store and maintain the records of resumes processed by the bot, including their names, classification folders, and processing statuses.

Key Features:

- **Data Storage**: Excel serves as a reliable platform for storing data in tabular format. It stores the information such as file names, folder classifications, and processing status.
- **Excel Automation**: UiPath integrates well with Excel, allowing the bot to read from and write to Excel files. Activities like 'Read Range' and 'Write Range' are used to interact with Excel files.
- **Report Generation**: Daily reports are generated and formatted within Excel, offering easy readability and accessibility for stakeholders.

---

3. Windows Task Scheduler / Orchestrator

Role in the System: Windows Task Scheduler or UiPath Orchestrator is used to schedule and automate the execution of the bot on a daily or predefined basis.

Key Features:

- **Task Scheduling**: Both tools allow for scheduling the bot to run at specified times, ensuring that the automation process runs consistently without manual intervention.

- **Orchestrator (Optional)**: If using UiPath Orchestrator, it provides additional features like centralizing robot management, tracking logs, and scheduling tasks across multiple environments.

---

4. SMTP (Simple Mail Transfer Protocol)

Role in the System: SMTP is used to send daily email notifications and reports to stakeholders, such as HR.

Key Features:

- **Email Sending**: UiPath uses the 'Send SMTP Mail Message' activity to automatically send the daily report (Excel file) to the recipient (e.g., HR).
- **Email Customization**: The email's subject, body, and attachments can be dynamically customized within the bot workflow.
- **Error Notifications**: In case of failure to send the email, error notifications are logged, and fallback mechanisms can be triggered.
- **5. Regular Expressions (Regex)**
- **Role in the System**: Regex is used for keyword matching within resume files to classify them into categories like Java, Python, and C.
- **Key Features**:
- **Pattern Matching**: Regular expressions allow the bot to search for specific patterns such as programming languages (e.g., 'Java', 'Python', 'C') within the resume text.
- **Flexible Classification**: By using regex, the bot can identify variations of keywords, such as 'JavaScript' or 'Python3,' and classify resumes more accurately.
- ---
- **6. Data Scraping Techniques**

- **Role in the System**: If resumes are provided in different formats (e.g., PDFs or web-based applications), data scraping tools or OCR (Optical Character Recognition) can be used to extract resume content for classification.
- **Key Features**:
- **PDF Data Extraction**: UiPath's 'Read PDF Text' activity or third-party tools can be used to extract text from PDFs, ensuring the bot can process resumes regardless of the file format.
- **OCR**: For scanned images or PDFs that contain text as images, OCR techniques (like UiPath's 'Read OCR Text' activity) can be employed to extract text data.
- ---
- **7. File System (Directory Management)**
- **Role in the System**: The bot interacts with the file system to move, organize, and classify resumes into folders based on the detected keywords (Java, Python, C).
- **Key Features**:
- **Folder Management**: Using activities like 'Create Directory' and 'Move File,' the bot ensures that resumes are categorized into the correct folders.
- **Automated Folder Creation**: Subfolders for each programming language (Java, Python, C) are created if they don't already exist, ensuring that the system stays organized.

Technical Design of the Automated Resume Classifier and Daily Report Bot

The **Automated Resume Classifier and Daily Report Bot** is designed to function efficiently using a set of tools, technologies, and frameworks that support automation, data processing, and reporting. Below is the detailed technical design of the bot based on the chosen tools and technologies:

---

1. UiPath Studio

Role in the System: UiPath Studio is used as the primary tool to design, develop, and automate the workflows for resume classification and daily report generation.

Key Features:

- **Drag-and-Drop Interface**: UiPath provides a user-friendly interface for designing automation workflows without requiring extensive coding.
- **Prebuilt Activities**: The tool comes with built-in activities such as 'Read Text File,' 'Move File,' 'Send SMTP Mail Message,' and 'Write Range' to handle the automation processes seamlessly.
- **Error Handling**: UiPath's robust exception handling framework ensures that any errors or exceptions in the workflow are caught and logged for analysis.

---

2. Microsoft Excel

Role in the System: Excel is used to store and maintain the records of resumes processed by the bot, including their names, classification folders, and processing statuses.

Key Features:

- **Data Storage**: Excel serves as a reliable platform for storing data in tabular format. It stores the information such as file names, folder classifications, and processing status.
- **Excel Automation**: UiPath integrates well with Excel, allowing the bot to read from and write to Excel files. Activities like 'Read Range' and 'Write Range' are used to interact with Excel files.
- **Report Generation**: Daily reports are generated and formatted within Excel, offering easy readability and accessibility for stakeholders.

---

3. Windows Task Scheduler / Orchestrator

Role in the System: Windows Task Scheduler or UiPath Orchestrator is used to schedule and automate the execution of the bot on a daily or predefined basis.

Key Features:

- **Task Scheduling**: Both tools allow for scheduling the bot to run at specified times, ensuring that the automation process runs consistently without manual intervention.
- **Orchestrator (Optional)**: If using UiPath Orchestrator, it provides additional features like centralizing robot management, tracking logs, and scheduling tasks across multiple environments.

---

4. SMTP (Simple Mail Transfer Protocol)

Role in the System: SMTP is used to send daily email notifications and reports to stakeholders, such as HR.

Key Features:

- **Email Sending**: UiPath uses the 'Send SMTP Mail Message' activity to automatically send the daily report (Excel file) to the recipient (e.g., HR).
- **Email Customization**: The email's subject, body, and attachments can be dynamically customized within the bot workflow.
- **Error Notifications**: In case of failure to send the email, error notifications are logged, and fallback mechanisms can be triggered.

---

5. Regular Expressions (Regex)

Role in the System: Regex is used for keyword matching within resume files to classify them into categories like Java, Python, and C.

Key Features:

- **Pattern Matching**: Regular expressions allow the bot to search for specific patterns such as programming languages (e.g., 'Java', 'Python', 'C') within the resume text.
- **Flexible Classification**: By using regex, the bot can identify variations of keywords, such as 'JavaScript' or 'Python3,' and classify resumes more accurately.

---

6. Data Scraping Techniques

Role in the System: If resumes are provided in different formats (e.g., PDFs or web-based applications), data scraping tools or OCR (Optical Character Recognition) can be used to extract resume content for classification.

Key Features:

- **PDF Data Extraction**: UiPath's 'Read PDF Text' activity or third-party tools can be used to extract text from PDFs, ensuring the bot can process resumes regardless of the file format.
- **OCR**: For scanned images or PDFs that contain text as images, OCR techniques (like UiPath's 'Read OCR Text' activity) can be employed to extract text data.

---

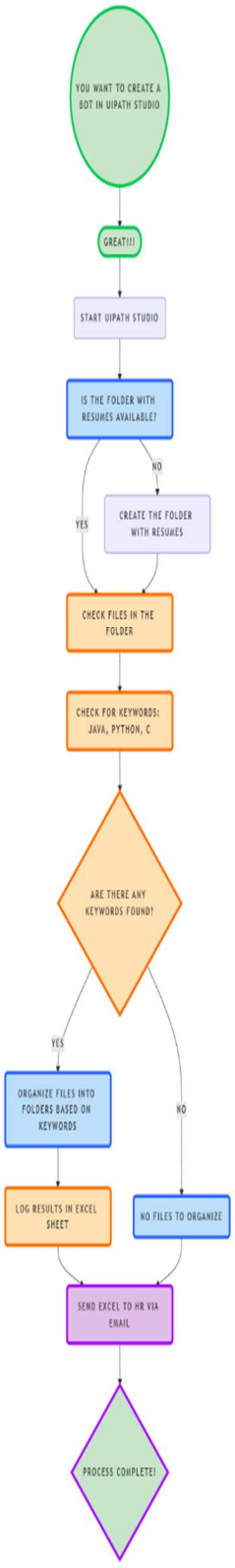
7. File System (Directory Management)

Role in the System: The bot interacts with the file system to move, organize, and classify resumes into folders based on the detected keywords (Java, Python, C).

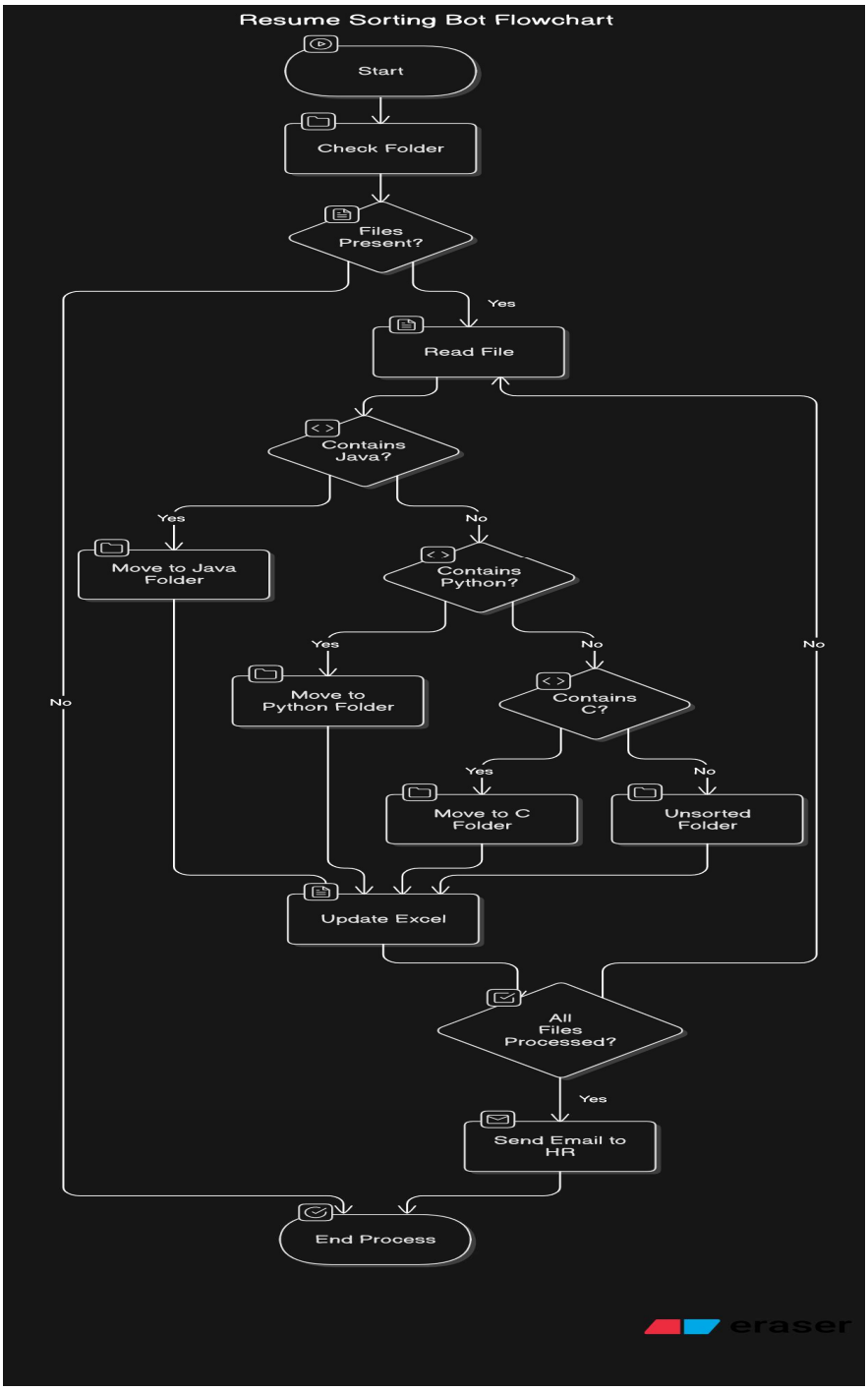
Key Features:

- **Folder Management**: Using activities like 'Create Directory' and 'Move File'

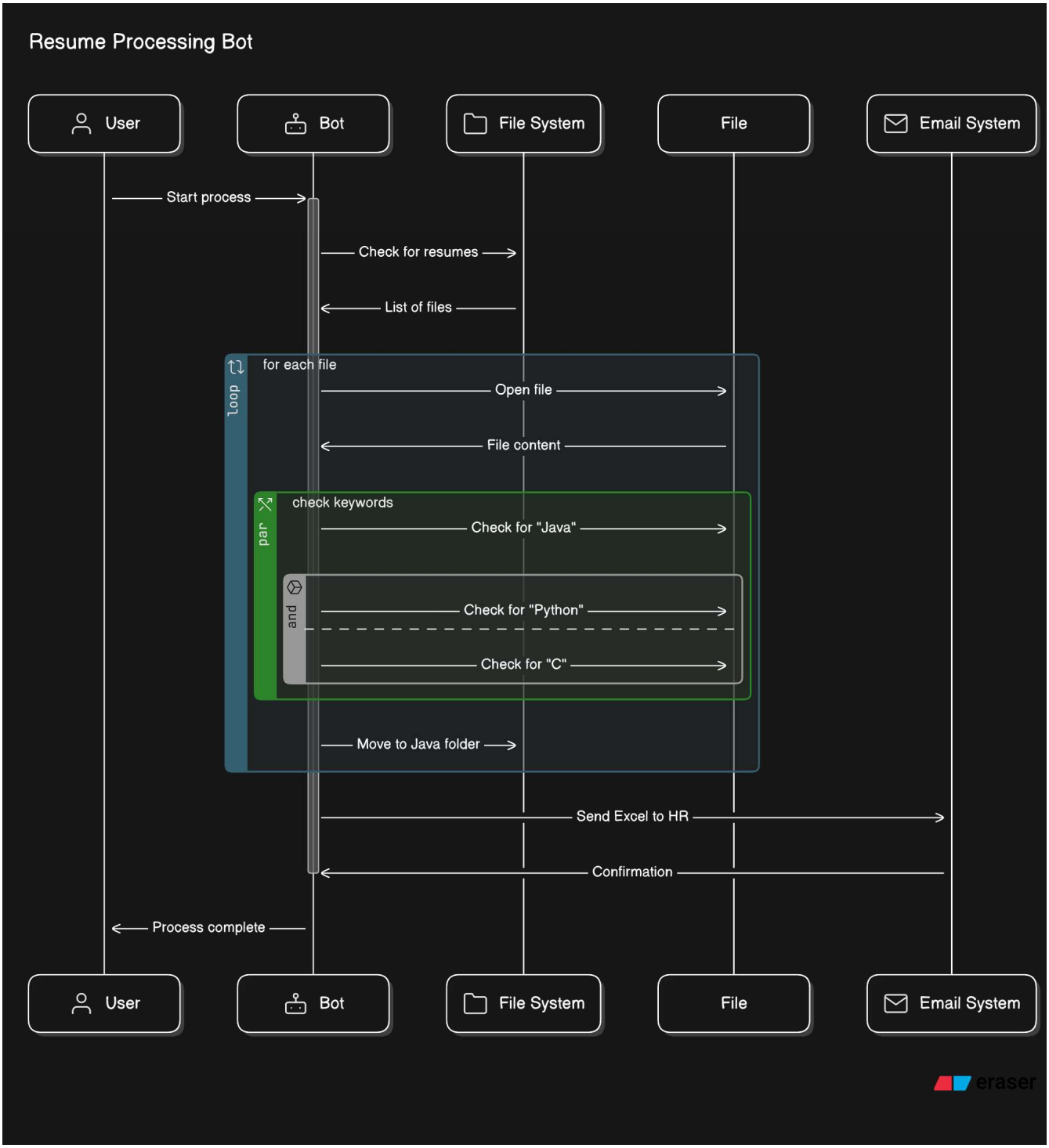
3.1. SYSTEM FLOW DIAGRAM



ARCHITECTURE DIAGRAM:



3.3 SEQUENCE DIAGRAM:



4.Project Description

The bot will process resumes from a folder, search for specific keywords (Java, Python, C), and move the resumes to respective folders based on these keywords. The bot will also generate a report in Excel with the name of the file, the folder it was moved to, and a score based on the presence of the keywords. Finally, it will send this Excel report to HR via email on a daily basis.

Detailed Flow:

1. **Retrieve Resume Files from Folder**
    - **Activity:** Directory.GetFiles
    - **Purpose:** Get a list of all resume files (PDF, DOCX, or TXT) from a specified folder.
    - **Implementation:**
        - Set the path of the folder containing resumes (e.g., "C:\Resumes").
        - Use Directory.GetFiles("C:\Resumes") to get a list of file paths.
2. **For Each Resume in the Folder**
    - **Activity:** For Each
    - **Purpose:** Iterate through each resume file to perform further actions.
    - **Implementation:**
        - Loop through the list of resume file paths.
        - For each file, perform the following steps.
3. **Read Content of Each Resume**
    - **Activity:** Read PDF Text, Read Text File, or Read Word Document
    - **Purpose:** Extract text content from each resume file to check for keywords.
    - **Implementation:**
        - Use Read PDF Text for PDF files.
        - Use Read Text File for text files.
        - Use Word Application Scope and Read Text for DOCX files.
        - Store the extracted content in a string variable (e.g., resumeText).
4. **Keyword Search (Java, Python, C)**
    - **Activity:** If (Multiple Conditions)
    - **Purpose:** Check if the extracted text contains any of the specified keywords (Java, Python, C).
    - **Implementation:**

        - Use an If statement to check if the resumeText contains the keywords:

vb
Copy code
resumeText.Contains("Java")
resumeText.Contains("Python")
resumeText.Contains("C")

5. **Move Resume to the Appropriate Folder**
    - **Activity:** Move File
    - **Purpose:** Move the resume file to the folder based on the keyword detected (Java, Python, or C).
    - **Implementation:**
        - If Java is found in the resume, move it to the Java folder.
        - If Python is found, move it to the Python folder.
        - If C is found, move it to the C folder.
        - Use the Move File activity to move the files to the respective folders.
6. **Assign Score Based on Keywords**
    - **Activity:** Assign
    - **Purpose:** Assign a score based on the presence of keywords in the resume.
    - **Implementation:**
        - For each resume, assign a score (e.g., 1 for Java, 2 for Python, 3 for C, etc.), depending on which keyword was found.
        - Keep track of the score for each resume in a variable (e.g., score).
7. **Store Data in Excel**
    - **Activity:** Excel Application Scope, Write Range
    - **Purpose:** Create an Excel file and write the data (resume file name, folder name, and score) into it.
    - **Implementation:**
        - Use Excel Application Scope to open an Excel file.
        - Use Write Range to write data to the Excel sheet (file name in column 1, folder name in column 2, and score in column 3).
        - Add headers like "File Name," "Folder Name," and "Score."
8. **Send Email with Excel Attachment**
    - **Activity:** Send SMTP Mail Message
    - **Purpose:** Send the Excel file as an email attachment to HR.

- **Implementation:**
  - Use Send SMTP Mail Message to send the email.
  - Set the recipient (HR's email address), subject, and body content.
  - Attach the generated Excel file with the resume sorting results.

9. **Automate Daily Execution (Optional)**
   - **Activity:** Use UiPath Orchestrator or Task Scheduler
   - **Purpose:** Automate the bot to run daily without manual intervention.
   - **Implementation:**
     - Schedule the process in UiPath Orchestrator to trigger daily.
     - Alternatively, use Task Scheduler to trigger the process at a set time each day.

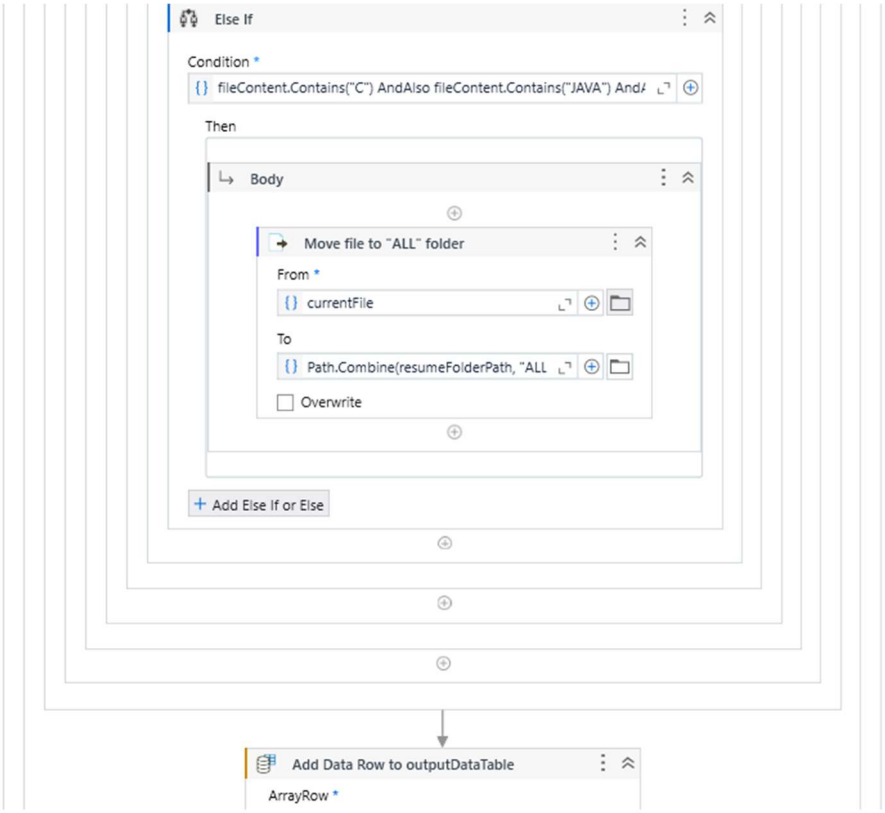
Excel Structure:

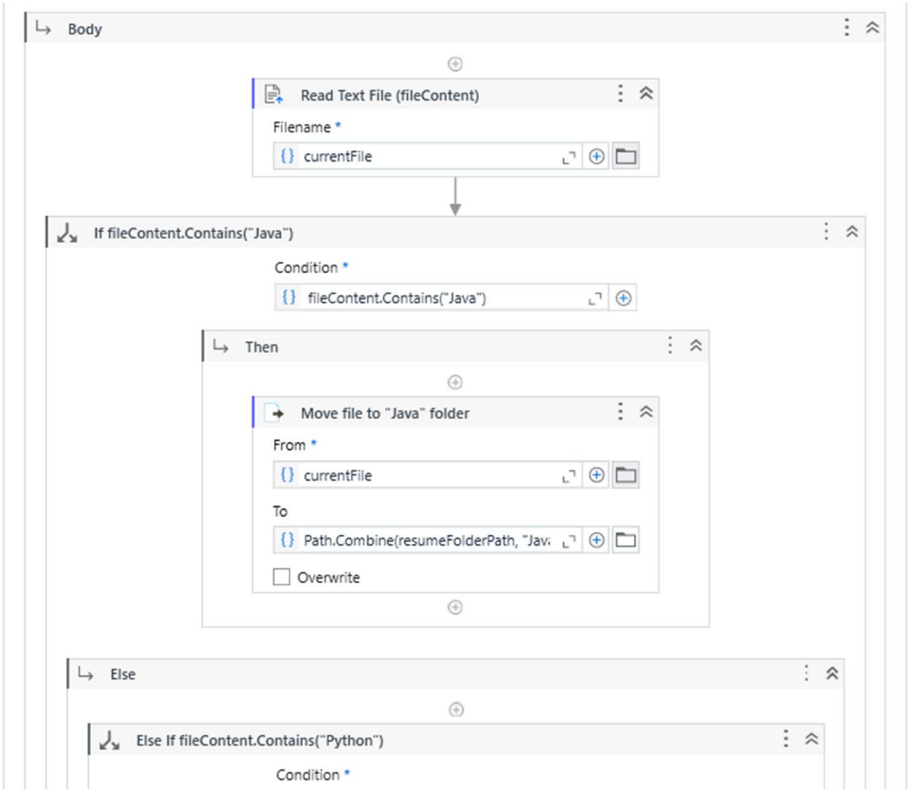
File Name	Folder Name	Score
resume1.pdf	Java	1
resume2.docx	Python	2
resume3.txt	C	3

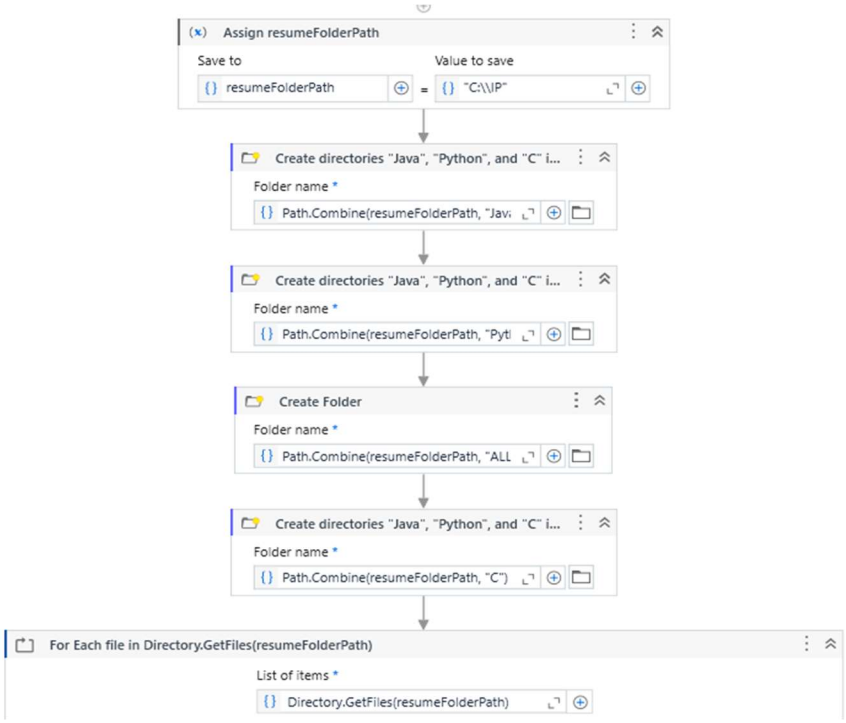
5.Conclusion

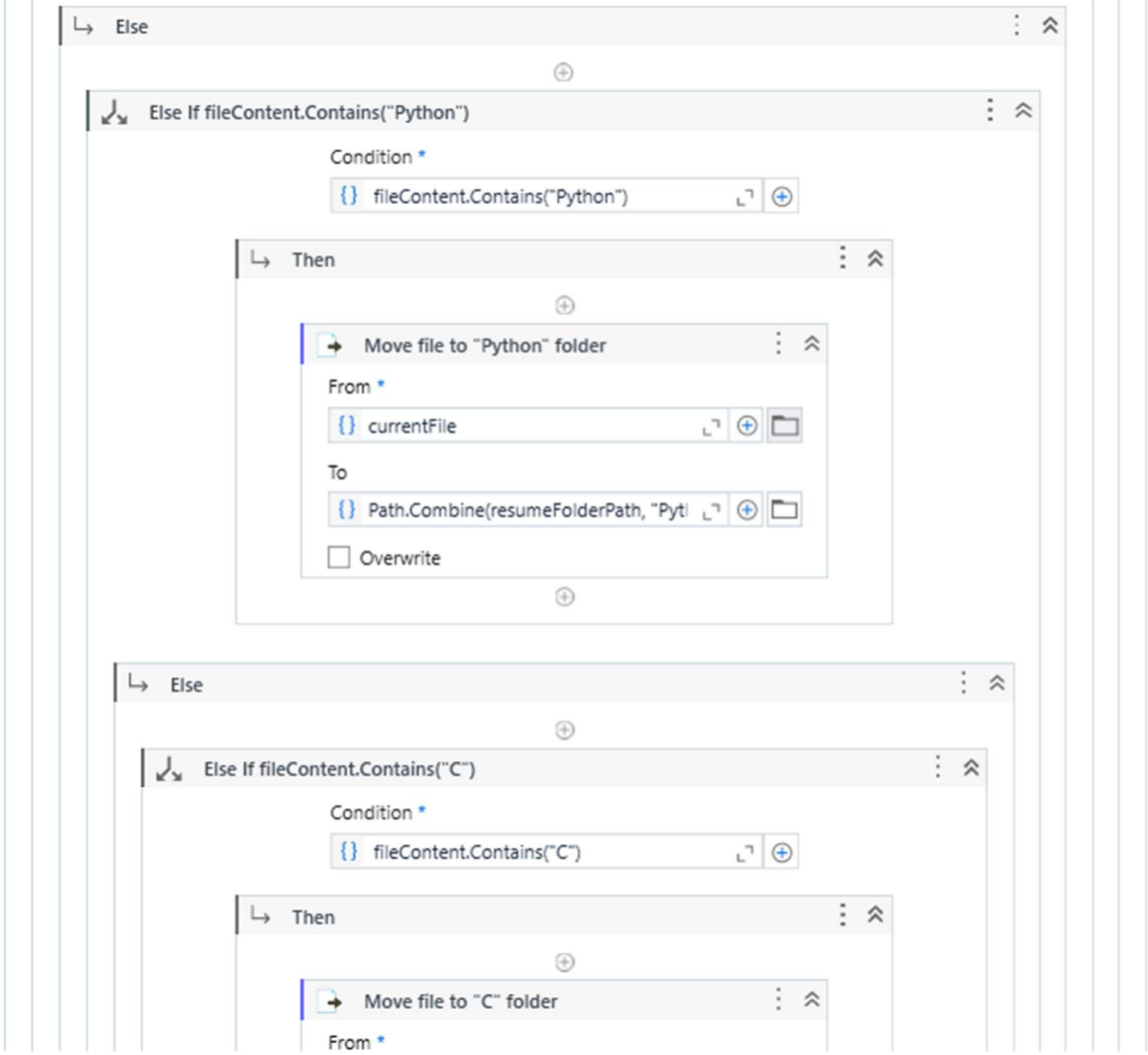
The Resume Sorting Bot in UiPath Studio automates the tedious task of processing resumes by categorizing them based on keywords (Java, Python, C) and storing them in corresponding folders. The bot also generates a report in an Excel sheet, capturing the file names, folder names, and associated scores. Afterward, it sends the generated Excel file to HR via email, ensuring seamless communication.By implementing this process, you save time, reduce human error, and create an efficient and reliable system for handling large volumes of resumes. Additionally, automating the task ensures that the process runs consistently and without manual intervention, and the daily email report keeps HR updated on the status of the resumes.This UiPath automation improves productivity and facilitates streamlined resume management, providing a significant advantage in any recruitment process.

APPENDICES


Else If
Condition *
{} fileContent.Contains("C") AndAlso fileContent.Contains("JAVA") And
Then
Body
Move file to "ALL" folder
From *
{} currentFile
To
{} Path.Combine(resumeFolderPath, "ALL
Overwrite
Add Else If or Else
Add Data Row to outputDataTable
ArrayRow *


Body
Read Text File (fileContent)
Filename *
{} currentFile
If fileContent.Contains("Java")
Condition *
{} fileContent.Contains("Java")
Then
Move file to "Java" folder
From *
{} currentFile
To
{} Path.Combine(resumeFolderPath, "Jav
Overwrite
Else
Else If fileContent.Contains("Python")
Condition *


Assign resumeFolderPath
Save to
resumeFolderPath
=
Value to save
"C:\\IP"
Create directories "Java", "Python", and "C" i...
Folder name *
Path.Combine(resumeFolderPath, "Jav
Create directories "Java", "Python", and "C" i...
Folder name *
Path.Combine(resumeFolderPath, "Pyt
Create Folder
Folder name *
Path.Combine(resumeFolderPath, "ALL
Create directories "Java", "Python", and "C" i...
Folder name *
Path.Combine(resumeFolderPath, "C")
For Each file in Directory.GetFiles(resumeFolderPath)
List of items *
Directory.GetFiles(resumeFolderPath)


Else
Else If fileContent.Contains("Python")
Condition *
fileContent.Contains("Python")
Then
Move file to "Python" folder
From *
currentFile
To
Path.Combine(resumeFolderPath, "Pyt
Overwrite
Else
Else If fileContent.Contains("C")
Condition *
fileContent.Contains("C")
Then
Move file to "C" folder
From *

References

1. UiPath Documentation. (2024). *UiPath Studio: Overview and Best Practices*. Retrieved from https://docs.uipath.com/studio
2. Ariponnammal, S. & Natarajan, S. (1994). 'Transport Phenomena of Sm Sel – X Asx', *Pramana – Journal of Physics*, Vol.42, No.1, pp.421-425.
3. Barnard, R.W. & Kellogg, C. (1980). 'Applications of Convolution Operators to Problems in Univalent Function Theory', *Michigan Mach. J.*, Vol.27, pp.81–94.
4. Shin, K.G. & McKay, N.D. (1984). 'Open Loop Minimum Time Control of Mechanical Manipulations and its Applications', *Proc. Amer. Contr. Conf.*, San Diego, CA, pp. 1231-1236.
5. Kress, R., & Martin, J. (2023). *Robotic Process Automation for Subscription Management: A Practical Guide*. Springer.
6. Choudhury, S., & Khan, A. (2021). 'Web Scraping and Data Automation in E-commerce Industry', *International Journal of Computer Applications*, Vol. 176, No. 6, pp. 15-19.
7. Patel, A., & Gupta, R. (2022). 'Improving Subscription Management with Automation Tools', *Journal of Business Technology and Management*, Vol. 29, pp. 101-107.
8. Vasquez, D. (2023). *Implementing Robotic Process Automation in the Digital Transformation of Businesses*. Wiley.
9. Zhang, L. & Lee, T. (2020). 'An Overview of Web Scraping Techniques and Their Applications', *International Journal of Data Science*, Vol. 18, No. 2, pp. 34-42.
10. UiPath Academy. (2024). *UiPath Orchestrator: Managing Automation at Scale*. Retrieved from https://academy.uipath.com