­­­­­

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



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



Outlines the major components of the Master Project (the overall output of the development, containing one or multiple projects that together cover the scope of the AI Agent System) taking into account all the business restrictions (scheduling, peaks, future increases in volume etc.). The focus of the Solution Architect will be on:

* Robustness;
* Scalability;
* Efficiency;
* Replicability

The information herein is targeted primarily at the developers that will initially implement the solution and subsequently at the support developers in case of change requests.

Full:  
  
Business Goal:  
The primary business goal of this payroll allocation process is to accurately distribute payroll costs across multiple business entities based on actual direct care time provided by employees. This ensures proper financial reporting, cost allocation, and compliance with accounting standards. The process aims to create a more precise representation of labor costs for each entity, which is crucial for financial management, budgeting, and decision-making in a multi-entity healthcare organization.  
  
Best Way to Accomplish This Business Goal:  
The best way to accomplish this goal would be through a fully automated, integrated system that connects all relevant data sources and performs the allocation in real-time. This would involve:  
1. Direct integration between the payroll system (ADP), the EMR system, and the financial ERP (MS Dynamics GP).  
2. Implementing a rules engine that can automatically apply the complex allocation rules and handle exceptions.  
3. Creating a centralized data warehouse that combines payroll, clinical, and financial data.  
4. Developing a user-friendly dashboard for monitoring and approving allocations.  
5. Implementing continuous auditing and reconciliation processes to ensure accuracy.  
This approach would minimize manual intervention, reduce errors, increase efficiency, and provide real-time insights into labor costs across entities.  
  
How AI Agents Could Improve the Process:  
AI agents could significantly enhance this process in several ways:  
1. Data Retrieval and Integration: AI agents could use APIs to automatically fetch data from ADP, the EMR system, and other relevant sources, eliminating the need for manual data retrieval and reducing the risk of human error.  
  
2. Intelligent Data Validation: Machine learning models could be trained to identify and flag potential errors or anomalies in the payroll data, improving data quality and reducing the need for manual validation.  
  
3. Dynamic Rule Application: AI could apply and update allocation rules dynamically based on changing business conditions, using natural language processing to interpret new guidelines or regulations.  
  
4. Predictive Analytics: AI models could analyze historical data to predict future allocation patterns, helping to identify trends and potential issues before they occur.  
  
5. Automated Report Generation: GPT models could be used to generate human-readable summary reports and explanations of complex allocations, making it easier for stakeholders to understand and approve the results.  
  
6. Continuous Learning: The AI system could learn from past allocations and exceptions, continuously improving its accuracy and efficiency over time.  
  
7. Natural Language Interfaces: AI agents could provide a conversational interface for users to query allocation data, request explanations, or initiate processes using natural language, making the system more accessible to non-technical users.  
  
By leveraging these AI capabilities, the process could become more automated, accurate, and insightful, allowing human workers to focus on high-level decision-making and exception handling rather than routine data processing tasks.  
  
AI Agent High-level Steps  
  
a. Step 1: Data Retrieval and Integration  
i. Reasoning: The AI agent would use APIs to automatically connect to and extract data from multiple systems (ADP, EMR, MS Dynamics GP). This step is crucial as it forms the foundation for all subsequent operations. The agent would need to handle authentication, manage data formats, and ensure all necessary information is collected.  
ii. Complexity: 3/5 - While API integration is a common task, the complexity arises from dealing with multiple systems, potential API changes, and ensuring data completeness and consistency across different sources.  
  
b. Step 2: Data Validation and Cleansing  
i. Reasoning: The AI agent would apply machine learning models to validate the retrieved data, identifying anomalies, inconsistencies, or missing information. It would cleanse the data by correcting common errors, standardizing formats, and filling in gaps where possible using historical data or predefined rules.  
ii. Complexity: 4/5 - This step requires sophisticated machine learning models to accurately detect various types of data issues. The complexity is high due to the need for continuous learning and adaptation to new data patterns and error types.  
  
c. Step 3: Application of Allocation Rules  
i. Reasoning: The AI agent would apply the complex set of business rules to allocate payroll costs across different entities based on direct care time. This involves interpreting and executing the rules, handling exceptions, and making decisions on edge cases.  
ii. Complexity: 5/5 - This is the most complex step as it requires the AI to understand and apply intricate business logic, handle numerous exceptions, and make judgment calls on ambiguous situations. The agent needs to continuously update its rule set based on changing business requirements.  
  
d. Step 4: Generation of Allocation Report  
i. Reasoning: The AI agent would create a detailed allocation report, including summaries, breakdowns by entity, and explanations for significant allocations or exceptions. This step involves data aggregation, statistical analysis, and natural language generation to produce human-readable reports.  
ii. Complexity: 3/5 - While data aggregation and basic reporting are straightforward, the complexity lies in generating insightful analyses and clear, context-aware explanations of the allocations.  
  
e. Step 5: Upload to Financial System  
i. Reasoning: The AI agent would prepare and upload the final allocated journal entries into MS Dynamics GP. This involves formatting the data correctly, initiating the upload process, and verifying the successful posting of entries.  
ii. Complexity: 2/5 - This step is relatively straightforward as it involves standard data formatting and system interaction. The main challenges are ensuring data accuracy and handling any system-specific quirks or errors.  
  
f. Step 6: Exception Handling and Human Notification  
i. Reasoning: Throughout the process, the AI agent would identify exceptions that require human intervention. It would categorize these exceptions, prepare detailed reports on each, and notify the appropriate personnel with relevant information and suggested actions.  
ii. Complexity: 4/5 - The complexity here lies in accurately identifying and categorizing exceptions, determining their severity, and providing meaningful context and suggestions to human operators. This requires advanced decision-making capabilities and effective human-AI interaction.  
  
Process trigger:  
Based on the process description, the most likely trigger for the payroll allocation process is the availability of the Detailed Payroll Report in ADP. This report seems to be generated at the end of each pay period, which is typically bi-weekly. Therefore, the AI Agent could be programmed to check for the availability of this report in ADP at regular intervals (e.g., every two weeks) or receive a notification from ADP when the report is ready. Once the report is available, this would automatically trigger the start of the payroll allocation process.  
  
Human Involvement:  
Assuming the AI Agent system is fully set up and operational, human involvement can be minimized but not entirely eliminated. Based on the process description and considering the complexity of payroll allocation, humans should be involved in the following key areas:  
  
a. Involvement 1: Exception Handling  
Humans need to be involved in handling complex exceptions that the AI Agent cannot resolve autonomously. This could include:  
- System exceptions where ADP or the EMR system is unavailable for an extended period.  
- Business exceptions that require judgment calls or interpretation of new policies.  
- Unusual allocation scenarios that fall outside the predefined rules.  
The AI Agent should be programmed to escalate these exceptions to the appropriate human personnel, providing all relevant information for quick decision-making.  
  
b. Involvement 2: Final Approval and Audit  
Before the final allocation is uploaded to MS Dynamics GP, a human should review and approve the allocation report. This step serves as a final check to ensure the AI Agent has performed the allocation correctly and to catch any potential issues that might have been missed. The human review could focus on:  
- Verifying that the total allocations match the original payroll amount.  
- Checking for any unusual or significant changes in allocation patterns.  
- Ensuring compliance with current accounting policies and regulations.  
  
This human involvement serves as a crucial control point, especially given the financial implications of the payroll allocation process. The AI Agent can facilitate this by generating a concise summary report highlighting key metrics and any notable changes or exceptions, allowing for efficient human review.  
  
Documentation  
  
a. Business Rules and Allocation Guidelines  
i. Reasoning: The AI agent would need detailed documentation of the business rules and allocation guidelines to accurately perform the payroll allocation. This would include information on how to handle different employee types, department codes, entity codes, and specific allocation scenarios. The process description mentions "predefined accounting rules" and "specific rules that determine which entries are allocable," indicating the existence of such documentation.  
ii. Location: Not explicitly mentioned in the process description. Likely an internal document maintained by the finance or payroll department.  
  
b. System Integration Specifications  
i. Reasoning: To effectively integrate with ADP, the EMR system, and Microsoft Dynamics GP, the AI agent would need technical documentation on the APIs, data formats, and integration protocols for each system. This is crucial for data retrieval and upload steps.  
ii. Location: Not explicitly mentioned. Typically provided by the respective software vendors or the organization's IT department.  
  
c. Exception Handling Procedures  
i. Reasoning: The process description mentions various types of exceptions (system, business, and business referral). The AI agent would need documentation on how to identify, categorize, and handle these exceptions, including when and how to escalate to human intervention.  
ii. Location: Not explicitly provided in the process description. Likely an internal document detailing exception handling protocols.  
  
d. Detailed Payroll Report Structure  
i. Reasoning: To correctly interpret and process the Detailed Payroll Report from ADP, the AI agent would need documentation on the report's structure, field definitions, and any codes or abbreviations used.  
ii. Location: Not explicitly mentioned. Typically provided by ADP or documented internally by the payroll team.  
  
e. MS Dynamics GP Journal Entry Format  
i. Reasoning: For uploading the final allocated entries into MS Dynamics GP, the AI agent would need documentation on the required format for journal entries, including field specifications and any system-specific requirements.  
ii. Location: Not explicitly mentioned. Usually provided in MS Dynamics GP documentation or internal guidelines.  
  
f. EMR Data Structure and Clinical Visit Definitions  
i. Reasoning: For future state allocation based on direct patient care data, the AI agent would need documentation on how to interpret EMR data, particularly the structure of clinical visit information.  
ii. Location: Not explicitly mentioned. Likely provided by the EMR system vendor or documented internally by the clinical informatics team.  
  
The process description does not provide any specific documentation links or document names. It refers to various systems and rules but does not point to explicit documentation sources. This lack of specific references suggests that the necessary documentation exists within the organization but is not directly linked in the process description. For a real-world implementation, it would be crucial to identify and compile all these documents to ensure the AI agent has access to the most up-to-date and accurate information for performing its tasks.

# 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 |  |
| Framework used | e.g. 2019.4 |

# Runtime guide

## Architectural structure of the Master Project

Display the interaction between Agents (package, queues, and network) 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 | ***Example:*** *Running on Sparky , the virtual backoffice machine. Scheduled every night after the report is generated from Zendesk.* |
| Prerequisites to run | ***Example:*** *Report was generated by Zendesk*  *Email received in* [*Zendesk\_reporting@uipath.com*](mailto:Zendesk_reporting@uipath.com)  *Having Excel on the machine* |
| Input Data | ***Example:*** *3 valid CSV files*  *2 source files in C:\ZendeskReporting* |
| Expected output | ***Example:*** *2 e-mails sent to e-mail address: management@uipath.com* |
| Reporting  (queues reporting, Kibana or another platform) | ***Example:*** *Orchestrator logs and jobs dashboards.* |
| How is Orchestrator used? | ***Example:*** *Orchestrator used for scheduling and asset passwords.* |
| Password policies  (mention any specific compliance requests) | ***Example:*** *G-mail password only, not expiring.* |
| Stored credentials  (Never use hardcoded credentials in the workflow!) | ***Example:*** *Stored in Orchestrator Assets* |

## 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) | ***Example:*** *DEV\_Env1\_EMEA ( UiPath computer)* |
| Environment prerequisites  (OS details, libraries, required apps) | ***Example:*** *Windows 7, Studio license, Microsoft Excel* |
| Repository for project  (where is the developed project stored) | ***Example:*** *\\myshare.com\Zendesk* |
| Configuration method  (assets, excel file, Json file) | ***Example:*** *Assets* |
| List of reused components | ***Example:*** *found via Connect Marketplace or Automation Hub components* |
|
| List of new reusable components | ***Example:*** *placeholders created in Automation Hub* |

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 |
| Example: Main | ***Example:*** *invokes all the other workflows* |

## Packages

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

|  |  |
| --- | --- |
| Package Name | Description |
| *Example: ZendeskReports.1.0.6285.31077.nupkg* | ***Example****: Reads the email generated by the Zendesk reporting platform from Zendesk\_reporting@gmail.com*   * *Downloads the 3 reporting files in the C:\ZendeskReporting\#currentdate# folder* * *Copies the files source.xlsx and source\_fantastic.xlsx from C:\ZendeskReporting\ to C:\ZendeskReporting\#currentdate#* * *Processes the data from the 3 downloaded files into source files* * *Sends the file over email to a recipient list* |

## Agents

Agent\_ID: 1  
Name: Monitoring and Discord Integration Agent  
Description: This agent focuses on monitoring the Discord server, performing initial triage, and integrating with Discord's API.  
Reasoning: This agent is crucial for the initial processing of incoming messages and ensuring proper routing of questions.  
Tasks:   
 • Monitor and Identify Questions using Discord API: Essential for capturing and processing incoming questions Complexity: 3  
 • Categorize and Prioritize Questions: Helps in efficient handling of incoming queries Complexity: 3  
 • Initial Triage for Human Escalation: Ensures complex issues are directed to human moderators Complexity: 3  
 • Manage Discord-specific actions (e.g., creating private threads for complex issues): Facilitates better organization of discussions within Discord Complexity: 2  
Type: ReAct  
Context:   
 • Server Guidelines and Rules: Needed to identify questions that may violate server rules or require special handling,   
 • Response Templates or Guidelines: Helps in initial categorization and prioritization of questions,   
Inputs:   
 • Discord API: Real-time Discord server events and messages  
Outputs:   
 • Agent 2: Categorized and prioritized questions with initial triage information  
 • Agent 5: Discord-specific actions (e.g., private thread creation requests)  
Tools: Discord API Integration Tool (Provides direct interface to Discord's API for monitoring messages, creating and managing private threads, and performing other Discord-specific actions, variable: discord\_api\_tool)  
Trigger: This agent is triggered continuously by new messages in the Discord server. It's the first agent in the process and doesn't require interaction with other agents to start.  
Decisions:   
 • Determine the category and priority of the incoming question: Based on the content of the message, server guidelines, predefined categories, urgency, and potential impact  
 • Determine how to handle the question (standard processing, human escalation, or private thread creation): Based on complexity, sensitivity, or if it falls outside the scope of automated responses  
System Prompt: You are an advanced Discord Integration and Monitoring Assistant, expertly designed to oversee and manage a Discord server's communication flow. Your primary focus is on efficiently handling incoming questions and ensuring smooth server operations.  
  
Your key responsibilities include:  
1. Continuously monitor the Discord server using the discord\_api\_tool to capture all incoming messages and questions in real-time.  
2. Analyze each message to categorize and prioritize questions based on content, urgency, and potential impact. Utilize server\_guidelines and response\_guidelines for informed decision-making.  
3. Perform initial triage on incoming questions, identifying complex issues that require human moderator attention.  
4. Manage Discord-specific actions, such as creating private threads for sensitive or complex discussions.  
  
When categorizing and prioritizing, consider topic relevance, user engagement, and rule alignment. Assess complexity and sensitivity for appropriate handling methods.  
  
Provide concise, structured outputs with clear categorization, priority levels, and triage recommendations. Use a professional yet friendly tone in Discord interactions.  
  
For human escalation or private thread creation, clearly communicate the requirement with a brief explanation.  
  
Your goal is to streamline communication, enhance user experience, and support efficient moderation. Continuously adapt your approach based on server activity and moderator feedback to improve performance over time.  
  
Agent\_ID: 2  
Name: Knowledge Base and FAQ Management Agent  
Description: This agent manages the internal knowledge base, FAQ system, and integrates with external documentation sources.  
Reasoning: This agent is essential for maintaining and accessing the knowledge base, which is crucial for providing accurate and up-to-date information.  
Tasks:   
 • Retrieve information from internal knowledge base and FAQ system: Provides quick access to existing information Complexity: 2  
 • Update Knowledge Base and FAQs: Ensures the knowledge base remains current and relevant Complexity: 3  
 • Integrate with official documentation and trusted websites: Expands the range of information available for answering questions Complexity: 3  
 • Manage multi-language support for knowledge base: Enables support for a diverse user base Complexity: 3  
Type: ReAct  
Context:   
 • Knowledge Base and FAQs: Core resource for retrieving and updating information,   
 • Official Documentation: Used to verify and supplement information in the knowledge base,   
 • Trusted Websites: Used for additional research and verification of information,   
Inputs:   
 • Agent 1: Categorized and prioritized questions with initial triage information  
Outputs:   
 • Agent 3: Relevant knowledge base information, FAQs, and documentation  
Tools: Knowledge Base Management System API (API for retrieving, updating, and managing entries in the internal knowledge base and FAQ system, variable: kbms\_api\_tool); Web Browser Interface (Interface for accessing and reading content from official documentation and trusted websites, variable: web\_browser\_tool); Translation API (API for translating knowledge base and FAQ content into multiple languages, variable: translation\_api\_tool)  
Trigger: This agent is triggered when the Monitoring and Discord Integration Agent or the Answer Generation and NLP Agent requests information from the knowledge base or when updates to the knowledge base are proposed.  
Decisions:   
 • Information Source Selection: Nature of the query, availability of information in each source, reliability of sources, specificity of the question, recency of information required, and depth of detail needed  
 • Knowledge Base Update: Novelty, relevance, and accuracy of the information, whether it contradicts existing entries, fills a gap in the current knowledge base, or provides more up-to-date information on a topic  
System Prompt: You are an intelligent Knowledge Base and FAQ Management Agent, responsible for maintaining and leveraging a comprehensive information repository. Your primary role is to efficiently retrieve, update, and integrate knowledge from various sources to provide accurate and timely information.  
  
Your tasks include:  
1. Retrieving information from the internal knowledge base and FAQ system using the kbms\_api\_tool.  
2. Updating the Knowledge Base and FAQs to ensure currency and relevance.  
3. Integrating information from official documentation and trusted websites using the web\_browser\_tool.  
4. Managing multi-language support for the knowledge base using the translation\_api\_tool.  
  
When retrieving information, prioritize based on query relevance, source reliability, and information recency. For knowledge base updates, evaluate novelty, relevance, and accuracy, ensuring consistency and filling information gaps.  
  
Your input will be categorized and prioritized questions with initial triage information from Agent 1. Your output should be relevant knowledge base information, FAQs, and documentation for Agent 3, presented in a clear, concise format.  
  
Utilize available tools efficiently to accomplish tasks. Maintain a professional and informative tone in all interactions, prioritizing accuracy and clarity. Continuously improve the knowledge base by identifying trends in queries and proactively updating information to address common issues.  
  
Agent\_ID: 3  
Name: Answer Generation and NLP Agent  
Description: This agent focuses on generating answers using advanced language models and natural language processing.  
Reasoning: This agent is crucial for generating intelligent and contextually appropriate responses to user queries.  
Tasks:   
 • Generate or Retrieve Answer using GPT or similar models: Provides intelligent and context-aware responses to user queries Complexity: 4  
 • Perform Natural Language Processing on user questions: Enhances understanding of user intent and context Complexity: 3  
 • Draft initial responses for admin review when necessary: Ensures human oversight for complex or sensitive issues Complexity: 3  
Type: ReAct  
Context:   
 • Knowledge Base and FAQs: Primary source for generating answers to user questions,   
 • Official Documentation: Used to supplement knowledge base information when generating answers,   
 • Response Templates or Guidelines: Ensures generated answers adhere to server's communication standards,   
Inputs:   
 • Agent 2: Relevant knowledge base information, FAQs, and documentation  
Outputs:   
 • Agent 4: Generated answers and NLP analysis results  
Tools:   
Trigger: This agent is triggered when the Monitoring and Discord Integration Agent passes a categorized and prioritized question that requires an answer.  
Decisions:   
 • Whether to draft an initial response for admin review: Necessity based on predefined guidelines, considering the nature of the user's query and the generated answer  
System Prompt: You are an advanced Answer Generation and NLP Agent, designed to provide intelligent and context-aware responses to user queries. Your role is to leverage cutting-edge language models and NLP techniques to understand user intent, generate accurate answers, and ensure high-quality communication.  
  
Tasks:  
1. Analyze user questions using NLP to extract key information and understand context.  
2. Generate or retrieve answers using GPT or similar models, drawing from the knowledge base, FAQs, and official documentation.  
3. Draft initial responses for admin review when dealing with complex or sensitive issues.  
  
You have access to a knowledge base, official documentation, and response guidelines. Use these to ensure accuracy and adherence to communication standards.  
  
When generating responses:  
- Consider user intent, query context, and relevant information from the knowledge base.  
- Strive for clarity, conciseness, and helpfulness.  
- Indicate if admin review is required.  
  
Output format:  
1. Generated answer or retrieved information  
2. Brief NLP analysis of the user's question  
3. If applicable, a clearly labeled draft response for admin review  
  
Maintain a professional yet friendly tone. If a query is outside your knowledge or capabilities, acknowledge this and suggest seeking human administrator assistance.  
  
Continuously learn from interactions to improve answer quality and accuracy over time.  
  
Agent\_ID: 4  
Name: Quality Assurance and Sentiment Analysis Agent  
Description: This agent refines answers, performs sentiment analysis, and learns from human edits.  
Reasoning: This agent is vital for maintaining high-quality responses and adapting to user feedback and sentiment.  
Tasks:   
 • Refine and Polish Answers: Ensures high-quality and appropriate responses Complexity: 3  
 • Perform Sentiment Analysis on user questions and interactions: Helps in understanding user emotions and tailoring responses accordingly Complexity: 3  
 • Learn from Human Edits: Continuously improves the quality of AI-generated responses Complexity: 4  
Type: ReAct  
Context:   
 • Server Guidelines and Rules: Used to ensure responses align with server policies and to identify potentially sensitive issues,   
 • Response Templates or Guidelines: Used to refine and polish answers before posting,   
Inputs:   
 • Agent 3: Generated answers and NLP analysis results  
Outputs:   
 • Agent 5: Refined answers with sentiment analysis  
 • Agent 2: Feedback for knowledge base updates  
Tools: Human Feedback Interface (An interface to receive and process human edits and feedback for continuous learning, variable: human\_feedback\_interface); Sentiment Analysis API (External API for advanced sentiment analysis to complement GPT's capabilities, variable: sentiment\_analysis\_api)  
Trigger: This agent is triggered when the Answer Generation and NLP Agent produces a draft answer or when user feedback is received on a posted answer.  
Decisions:   
 • The agent needs to decide whether and how to refine the generated answer before sending it to the next agent.: Does the answer align with server guidelines and rules? Does the answer meet the quality standards set in the response templates or guidelines? How can the answer be adjusted based on the sentiment analysis results?  
 • The agent must decide whether to route feedback to Agent 2 for knowledge base updates.: Does the feedback or human edit contain new information that should be added to the knowledge base? Is there a correction to existing information that needs to be updated in the knowledge base? Is the feedback generalizable enough to be useful for future interactions?  
System Prompt: You are a sophisticated Quality Assurance and Sentiment Analysis Agent with expertise in linguistics, psychology, and data analysis. Your mission is to refine answers, analyze sentiment, and learn from human edits to ensure high-quality, appropriate responses.  
  
Your tasks:  
1. Refine answers using server\_guidelines and response\_guidelines.  
2. Perform sentiment analysis on user interactions using built-in capabilities and sentiment\_analysis\_api.  
3. Learn from human edits via human\_feedback\_interface.  
  
When refining:  
- Enhance clarity and coherence  
- Ensure adherence to server rules  
- Adjust tone based on sentiment analysis  
- Address sensitive topics carefully  
  
Provide a concise sentiment summary, including emotion and intensity. Use this to tailor refined answers.  
  
Analyze human edits to identify key improvements. Determine if feedback contains new information or corrections for Agent 2's knowledge base updates.  
  
Output: Refined answer, sentiment analysis summary, and feedback for knowledge base updates. Maintain a professional, empathetic tone, adapting to the user's emotional state.  
  
Aim for continuous improvement by effectively incorporating human feedback and staying updated on evolving language trends and cultural sensitivities.  
  
Success metrics: Quality and appropriateness of refined answers, accuracy of sentiment analysis, and effective integration of human feedback for ongoing enhancement.  
  
Agent\_ID: 5  
Name: Response Management and Feedback Agent  
Description: This agent handles the posting of responses, monitors user feedback, and manages the ticketing system.  
Reasoning: This agent is essential for managing the final stages of response delivery and handling user feedback.  
Tasks:   
 • Post AI-generated Responses using Discord API: Ensures timely delivery of responses to users Complexity: 2  
 • Monitor User Feedback and Reactions: Allows for continuous improvement based on user satisfaction Complexity: 3  
 • Manage Ticketing or Tagging System: Helps in organizing and tracking complex or ongoing issues Complexity: 3  
 • Track Response Times and Generate Reports: Provides insights into system performance and areas for improvement Complexity: 2  
Type: ReAct  
Context:   
 • Response Templates or Guidelines: Ensures posted responses meet server's communication standards,   
 • Server Guidelines and Rules: Used to manage the ticketing system and prioritize issues based on server policies,   
Inputs:   
 • Agent 4: Refined answers with sentiment analysis  
 • Agent 1: Discord-specific actions (e.g., private thread creation requests)  
Outputs:   
 • Discord API: Responses and Discord actions  
 • System: Response time and feedback reports  
Tools: Discord API Integration Tool (Enables posting AI-generated responses, monitoring user feedback, and tracking message timestamps on Discord, variable: discord\_integration\_tool); Ticketing System API Tool (Manages the creation, updating, and retrieval of tickets for organizing and tracking issues, variable: ticket\_management\_tool); Data Analytics and Reporting Tool (Analyzes data from Discord interactions and the ticketing system to generate performance reports and statistics, variable: analytics\_reporting\_tool)  
Trigger: This agent is triggered when the Quality Assurance and Sentiment Analysis Agent approves an answer for posting, or when new user feedback is detected on a posted answer.  
Decisions:   
 • How to categorize and respond to user feedback (positive, negative, neutral): Analysis of user reactions and comments using sentiment analysis from Agent 4's input  
 • Whether to create a new ticket, update an existing one, or close a resolved issue: Nature of the user query/feedback, existence of related open tickets, and resolution status of the issue  
 • How to prioritize tickets and issues in the system: Server guidelines and rules, urgency of the issue, impact on users, and time since the ticket was opened  
System Prompt: You are an advanced Response Management and Feedback Agent for Discord, expertly handling user interactions, feedback, and ticketing systems. Your mission is to ensure seamless communication and maintain high-quality user experiences.  
  
Tasks:  
1. Post AI-generated responses via Discord API promptly and accurately.  
2. Monitor user feedback and reactions for continuous improvement.  
3. Manage ticketing system for complex or ongoing issues.  
4. Track response times and generate insightful reports.  
  
Tools: Discord API Integration (discord\_integration\_tool), Ticketing System API (ticket\_management\_tool), Data Analytics and Reporting (analytics\_reporting\_tool).  
  
Adhere to response\_guidelines and server\_guidelines. Maintain a friendly, helpful tone while following Discord etiquette. Analyze user feedback sentiment (positive, negative, neutral) based on reactions and comments. Create, update, or close tickets as needed, prioritizing by urgency, impact, and time elapsed.  
  
Generate concise, data-driven reports on response times and user satisfaction in an easy-to-understand format for administrators. Continuously adapt and improve based on feedback and performance metrics.  
  
When interacting with users, be empathetic and solution-oriented. For complex issues, create private threads or escalate to appropriate channels. Proactively identify trends in user feedback to suggest system improvements.  
  
Success is measured by user satisfaction, response accuracy, efficient issue resolution, and overall system performance improvement.

# Other Details

### Future Improvements

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

***Example:***

*• Optimize the processing algorithm*

*• Implement process error recovery (retry)*

*• Enable support for multiple template files*

### Other Remarks

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

***Example:*** *The workflow should run every night at 7PM Be careful not to schedule it before the report is generated by Zendesk.*

The Zendesk generated data is always 1 day old.