­­­­­

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



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.

Discord Support Agent Chain  
  
1. Business goal  
The primary business goal of this process is to maintain an engaged, informed, and satisfied community within the Discord server. By promptly addressing user questions with accurate and helpful responses, the process aims to foster a positive user experience, encourage user retention, and build a knowledgeable community. This, in turn, can lead to increased user engagement, a stronger sense of community, and potentially attract new members to the server.  
  
2. Best Way to Accomplish This Business Goal  
The best way to accomplish this business goal is to implement a systematic approach that combines efficient monitoring, quick response times, and continuous improvement of the knowledge base. This can be achieved by:  
  
1. Implementing a ticketing system within Discord to track and prioritize questions.  
2. Creating a comprehensive, easily searchable knowledge base that is regularly updated.  
3. Training admins on effective communication and research techniques.  
4. Establishing clear guidelines for response times and quality.  
5. Regularly analyzing common questions to proactively address recurring issues.  
6. Encouraging community members to help answer questions, fostering a collaborative environment.  
7. Utilizing analytics to track response times, user satisfaction, and community engagement metrics.  
  
By focusing on these areas, the server can ensure consistent, high-quality responses while building a self-sustaining community of knowledgeable users.  
  
3. How AI Agents Could Improve the Process  
a. Automated Monitoring: AI could continuously monitor all channels, instantly identifying questions using natural language processing, eliminating the need for manual monitoring.  
  
b. Instant Responses: For common questions, an AI agent could provide immediate, accurate responses by accessing a knowledge base or using a fine-tuned language model.  
  
c. Research Assistance: For more complex questions, AI could quickly gather relevant information from various sources, presenting a summary to the human admin for review and customization.  
  
d. Language Model Integration: Utilizing GPT or similar models, the AI could generate human-like responses, which could then be reviewed and approved by admins before posting.  
  
e. Dynamic FAQ Updates: The AI could automatically identify frequently asked questions and suggest updates to the knowledge base or FAQ section.  
  
f. Sentiment Analysis: AI could analyze user reactions to responses, helping to refine the tone and content of future answers.  
  
g. Multi-language Support: AI could provide real-time translation, allowing admins to communicate with users in various languages.  
  
h. API Integration: The AI agent could interface with Discord's API to manage messages, user roles, and server settings more efficiently.  
  
4. AI Agent High-level Steps  
a. Step 1: Monitor and Identify Questions  
i. Reasoning: The AI agent would continuously monitor all Discord channels using natural language processing to identify messages containing questions or requiring admin attention. This step is crucial as it forms the basis for all subsequent actions. The AI would need to understand context, differentiate between rhetorical and actual questions, and prioritize urgent inquiries.  
ii. Complexity: 3/5 - While monitoring is straightforward, accurately identifying and categorizing questions requires sophisticated NLP capabilities.  
  
b. Step 2: Analyze and Categorize Questions  
i. Reasoning: Once a question is identified, the AI agent would analyze its content to determine the category (e.g., technical, community guidelines, general information) and complexity. This step is essential for routing questions to the appropriate knowledge base or determining if human intervention is needed.  
ii. Complexity: 4/5 - This requires advanced language understanding and the ability to interpret context and nuance.  
  
c. Step 3: Generate or Retrieve Response  
i. Reasoning: For common or straightforward questions, the AI would either retrieve pre-approved answers from a knowledge base or generate a response using a language model like GPT. For more complex questions, it would compile relevant information from various sources to assist in formulating a response.  
ii. Complexity: 4/5 - While retrieving information is relatively simple, generating contextually appropriate and accurate responses is highly complex.  
  
d. Step 4: Review and Refine Response (Exception Handling)  
i. Reasoning: This step involves checking the generated response for accuracy, completeness, and adherence to community guidelines. If the AI's confidence in its response is below a certain threshold, or if the question is flagged as complex, it would route the question to a human admin for review or handling.  
ii. Complexity: 5/5 - This is the most complex step as it requires advanced decision-making capabilities and the ability to self-assess the quality and appropriateness of responses.  
  
e. Step 5: Post Response  
i. Reasoning: Once a response is approved (either automatically for high-confidence answers or after human review), the AI would post the response in the appropriate channel, ensuring proper formatting and tagging the original user.  
ii. Complexity: 1/5 - This is a straightforward task involving API calls to Discord and following predefined formatting rules.  
  
f. Step 6: Learn and Update Knowledge Base  
i. Reasoning: The AI would analyze user reactions to responses, identify frequently asked questions, and use this information to suggest updates to the knowledge base. This step is crucial for continuous improvement of the system.  
ii. Complexity: 4/5 - While tracking reactions is simple, interpreting them correctly and making intelligent updates to the knowledge base is complex.  
  
g. Step 7: Generate Analytics and Reports  
i. Reasoning: The AI would compile data on response times, user satisfaction, common issues, and other relevant metrics. This information would be used to generate reports for admins and to inform strategies for improving the community.  
ii. Complexity: 2/5 - Once the data collection is in place, generating analytics and reports is relatively straightforward.  
  
5. Process trigger  
Based on the process description, the trigger to start the process is the presence of new messages in the Discord server. The AI agent would continuously monitor all channels for new activity, and the process would be initiated whenever a new message is detected. This aligns with the first step in the original process: "Log into Discord and access the server" and "Review all channels for new messages and activity." In an AI-driven system, this monitoring would be constant and automatic, eliminating the need for a human to manually check for new messages.  
  
6. Human Involvement  
a. Involvement 1: Complex or Sensitive Questions  
Humans should be involved when the AI agent encounters questions that are too complex, sensitive, or outside its scope of knowledge. This aligns with the exception handling mentioned in the original process: "If the admin is unsure of an answer: Inform the user that research is needed and provide an estimated response time" and "If a question is outside the admin's expertise: Escalate to appropriate team." The AI agent can be programmed to recognize these situations and automatically escalate them to a human admin for review and response.  
  
b. Involvement 2: Knowledge Base and System Improvement  
Humans should periodically review and approve suggested updates to the knowledge base proposed by the AI agent. While the AI can learn from interactions and suggest improvements, human oversight ensures that the knowledge base remains accurate, appropriate, and aligned with the server's goals. This involvement relates to the original process step: "Update Knowledge Base (if applicable)." Additionally, humans should review analytics and reports generated by the AI to make strategic decisions about community management and system improvements, which goes beyond the original process but is crucial for long-term success.  
  
7. Documentation  
a. Server-specific Knowledge Base  
i. Reasoning: The AI agent would need access to a comprehensive knowledge base containing information specific to the Discord server. This would include server rules, frequently asked questions, community guidelines, and any other relevant information. This aligns with the original process mentioning "internal knowledge base" and the need to "Update knowledge base and FAQs." Having this documentation would enable the AI to provide accurate, server-specific responses and maintain consistency in its answers.  
ii. Document name or link: Not specified in the process description. This would likely be an internal document or database maintained by the server administrators.  
  
b. Official Discord Documentation  
i. Reasoning: The AI agent would benefit from access to official Discord documentation to answer technical questions about Discord features, functionality, and best practices. This is implied in the original process where it mentions using "appropriate resources to find accurate information" including "official documentation."  
ii. Document name or link: While not explicitly mentioned in the process description, this would typically be found at https://discord.com/developers/docs  
  
c. External Trusted Resources  
i. Reasoning: The process description mentions using "trusted websites" for research. The AI agent would need access to a curated list of reliable external resources to supplement its knowledge on topics that may not be covered in the server-specific or Discord documentation.  
ii. Document name or link: Not specified in the process description. This would likely be a maintained list of URLs or API endpoints for trusted information sources.  
  
d. Response Templates and Formatting Guidelines  
i. Reasoning: To ensure consistency in tone and formatting, the AI agent would need documentation on how to structure responses. This is implied in the original process where it mentions using "appropriate formatting (e.g., bold, italics, code blocks) to enhance readability" and ensuring the response "is friendly and aligns with the server's tone."  
ii. Document name or link: Not specified in the process description. This would likely be an internal style guide or set of templates.  
  
e. Escalation Procedures  
i. Reasoning: The AI agent would need clear guidelines on when and how to escalate issues to human administrators. This is implied in the exception handling section of the original process, which mentions scenarios where human intervention is required.  
ii. Document name or link: Not specified in the process description. This would likely be an internal document outlining specific criteria and procedures for escalation.  
  
8. Solution Reasoning  
The proposed AI-driven solution significantly enhances the efficiency and effectiveness of the Discord server admin question handling process. By automating routine tasks such as monitoring, categorizing questions, and generating responses, the AI agent can handle a much larger volume of inquiries more quickly than human admins alone. This allows for near-instantaneous responses to common questions, improving user satisfaction and engagement.  
  
The solution maintains the quality of responses by incorporating a knowledge base, which can be continually updated and improved based on user interactions. The AI's ability to learn from these interactions and suggest updates ensures that the knowledge base remains current and relevant.  
  
Human involvement is strategically limited to complex cases and oversight roles, allowing admins to focus on high-value tasks that require human judgment and creativity. This not only makes better use of human resources but also ensures that sensitive or complex issues receive appropriate attention.  
  
The multi-step process, from monitoring to analytics generation, creates a comprehensive system that not only answers questions but also provides insights for continuous improvement of the server and its community. The varying complexity levels of each step demonstrate where AI excels (e.g., in monitoring and basic response generation) and where more sophisticated AI capabilities are needed (e.g., in understanding context and self-assessing response quality).  
  
By leveraging AI in this way, the solution addresses the business goal of maintaining an engaged and informed community more effectively than a purely human-driven process could, while still retaining the option for human intervention when necessary. This balanced approach ensures scalability, consistency, and continuous improvement in community management.

# 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 Identification Agent  
Description: This agent focuses on the initial stages of detecting and categorizing questions using Discord's API.  
Reasoning: This agent is crucial for initiating the question-answering process by detecting and categorizing incoming questions  
Tasks:   
 • Monitor Discord channels using Discord API: Necessary for continuous monitoring of new messages in the Discord server Complexity: 3  
 • Identify Questions using Natural Language Processing: Required to accurately detect questions within messages Complexity: 3  
 • Analyze and Categorize Questions: Essential for proper routing and handling of questions Complexity: 4  
Type: ReAct  
Context:   
 • Comprehensive information about the Discord server, including rules, guidelines, and common topics: Needed to accurately identify and categorize questions within the server's context,   
Inputs:   
Outputs:   
 • Agent 2: Identified and categorized question  
 • Agent 4: Question details for ticketing  
Tools: Discord API Integration Tool: Enables real-time monitoring of Discord channels and retrieval of messages for analysis by the AI agent.  
Trigger: Continuous monitoring of new messages in the Discord server  
Decisions:   
System Prompt: You are an advanced Monitoring and Identification Agent for a Discord server. Your mission is to detect and categorize questions using the Discord API and Natural Language Processing. Monitor channels in real-time, accurately identify questions, and categorize them based on server context and guidelines.  
  
Utilize the Discord API Integration Tool for message retrieval and analysis. Apply NLP techniques to distinguish questions, including implicit queries and various formats. Consider message content, context, and user interactions.  
  
For each identified question:  
1. Analyze content and context thoroughly  
2. Categorize based on server guidelines and common topics  
3. Assess confidence level of identification and categorization  
  
Output format:  
- Question Detected: [Yes/No]  
- Question Content: [Concise summary]  
- Category: [Assigned category]  
- Confidence Level: [High/Medium/Low]  
- Relevant Context: [Brief notes on surrounding conversation or user history]  
  
Pass findings to Agent 2 for processing and Agent 4 for ticketing. Flag ambiguous cases for human review. Continuously learn from feedback to improve accuracy. Adapt to evolving server dynamics and new question patterns. Maintain neutrality and efficiency in your analysis.  
  
Agent\_ID: 2  
Name: Response Generation Agent  
Description: This agent handles the creation or retrieval of appropriate responses, utilizing the knowledge base and external resources.  
Reasoning: This agent is responsible for creating or retrieving appropriate responses to questions, which is a core function of the system  
Tasks:   
 • Access and query internal knowledge base: Necessary to retrieve relevant information for generating responses Complexity: 2  
 • Generate or Retrieve Response: Core function of creating appropriate answers to questions Complexity: 4  
 • Research using trusted external websites if necessary: Allows for gathering additional information when the knowledge base is insufficient Complexity: 3  
 • Human Interaction Management for Complex Questions: Ensures proper handling of questions that require human expertise Complexity: 3  
Type: ReAct  
Context:   
 • Detailed information about the server to formulate accurate responses: Essential for generating context-appropriate and accurate responses,   
 • List of approved external websites for additional research: Needed when the knowledge base doesn't contain sufficient information,   
Inputs:   
 • Agent 1: Identified and categorized question  
Outputs:   
 • Agent 3: Generated or retrieved response  
Tools: Knowledge Base Query Tool: An interface to search and retrieve information from the internal knowledge base, supporting complex queries and returning formatted results., Web Search and Information Extraction Tool: A tool that performs web searches on trusted websites, extracts relevant information, and filters results based on credibility.  
Trigger: Receives a categorized question from the Monitoring and Identification Agent  
Decisions:   
 • The agent needs to decide whether the information available in the internal knowledge base is sufficient to answer the question or if external research is necessary.: The agent will need to compare the query results from the knowledge base against the requirements of the question. If the knowledge base lacks comprehensive or up-to-date information to fully answer the question, the agent will decide to use the Web Search and Information Extraction Tool.  
 • The agent must determine if a question is too complex or specialized to be answered by the AI system and requires human expertise.: The agent will need to assess the complexity and specificity of the question, as well as the confidence level in its generated response. If the question falls outside the scope of the AI's capabilities or if the confidence in the answer is below a certain threshold, the agent will decide to route the question to human experts.  
System Prompt: You are an intelligent Response Generation Agent tasked with creating accurate and helpful answers to user queries. Your primary tools are a Knowledge Base Query Tool and a Web Search and Information Extraction Tool. Begin by querying the internal knowledge base. If insufficient, use the web search tool to gather information from approved external websites.  
  
Consider server details for context-appropriate responses. For complex or specialized questions beyond your capabilities, route them to human experts.  
  
Your input is identified and categorized questions from Agent 1. Output well-formatted, clear, and concise responses to Agent 3.  
  
Analyze each question to determine the best answering approach. Prioritize internal knowledge but supplement with external research when needed. Balance comprehensiveness and conciseness in your responses.  
  
For human interactions, maintain a professional, helpful tone. Explain clearly why human expertise is needed. Your effectiveness is measured by the accuracy, relevance, and helpfulness of your responses, and your ability to recognize when human intervention is necessary.  
  
Always strive to improve the quality and efficiency of your responses. Learn from each interaction to enhance your future performance.  
  
Agent\_ID: 3  
Name: Response Review and Posting Agent  
Description: This agent manages the final stages of response preparation and posting, ensuring adherence to server guidelines and proper formatting.  
Reasoning: This agent is crucial for ensuring the quality and appropriateness of responses before they are posted to the Discord server  
Tasks:   
 • Review and Refine Response: Ensures responses meet quality standards and adhere to server guidelines Complexity: 5  
 • Post Response using Discord API: Delivers the final response to the user in the Discord server Complexity: 2  
 • Human Interaction Management for Approvals: Allows for human oversight on sensitive or complex responses Complexity: 2  
Type: ReAct  
Context:   
 • Guidelines for structuring responses and using appropriate formatting: Ensures consistency in tone and formatting across all responses,   
 • Guidelines for when and how to escalate issues to human administrators: Needed to handle complex or sensitive situations appropriately,   
Inputs:   
 • Agent 2: Generated or retrieved response  
Outputs:   
 • Agent 4: Posted response details  
 • Agent 5: Response data for analytics  
Tools: Discord API Integration Tool: Provides functions to interact with the Discord API, including authentication, posting messages, editing/deleting messages, adding reactions, and managing a conditional approval process for sensitive or complex responses.  
Trigger: Receives a generated response from the Response Generation Agent  
Decisions:   
 • Determine if human approval is required before posting the response: Assess if the response contains sensitive information, addresses a complex issue, or falls under predefined categories requiring human oversight, based on provided guidelines for escalation to human administrators  
System Prompt: You are a meticulous Response Review and Posting Agent for a Discord server. Your mission is to ensure all responses meet high-quality standards, adhere to server guidelines, and are properly formatted before posting. Review and refine responses, focusing on quality, guideline adherence, and formatting. Use the Discord API Integration Tool to post finalized responses. Manage human approvals for sensitive or complex issues.  
  
Consult provided guidelines for structuring and formatting. Maintain consistent tone across responses. If a response contains sensitive information, addresses complex issues, or requires oversight, initiate the human approval process using the Discord API tool's conditional approval function.  
  
For posting, authenticate with the Discord API and use appropriate functions to deliver messages, add reactions, and manage edits or deletions as needed.  
  
When human interaction is required, clearly communicate the need for approval with context and reasoning. Maintain a professional yet approachable tone in all interactions.  
  
Your inputs are responses from Agent 2. Outputs include posted response details to Agent 4 and analytics data to Agent 5.  
  
Success means delivering high-quality, guideline-compliant responses, appropriately handled based on content and complexity. Your attention to detail and efficient management of response delivery are crucial for maintaining server standards and user satisfaction.  
  
Agent\_ID: 4  
Name: Knowledge Management and Ticketing Agent  
Description: This agent focuses on maintaining the knowledge base and managing a ticketing system for tracking questions.  
Reasoning: This agent is essential for maintaining and improving the system's knowledge base, which is crucial for providing accurate and up-to-date responses  
Tasks:   
 • Implement and manage ticketing system: Allows for efficient tracking and management of questions and responses Complexity: 4  
 • Learn and Update Knowledge Base: Keeps the knowledge base current and improves its accuracy over time Complexity: 4  
 • Update FAQs based on common questions: Improves efficiency by addressing frequently asked questions proactively Complexity: 3  
 • Periodic Human Review Management: Ensures human oversight and quality control of the knowledge base Complexity: 3  
Type: ReAct  
Context:   
 • The knowledge base itself, which this agent updates and maintains: Essential for the agent's primary function of knowledge management,   
Inputs:   
 • Agent 1: Question details for ticketing  
 • Agent 3: Posted response details  
Outputs:   
 • Agent 2: Updated knowledge base  
Tools: Knowledge Base Management System (KBMS) API: Comprehensive API for managing the knowledge base, FAQs, and ticketing system. Allows creating, reading, updating, and deleting entries, as well as tagging items for review., Human Review Scheduler: Tool for scheduling and notifying human reviewers for periodic quality control of the knowledge base.  
Trigger: Receives a signal from the Response Review and Posting Agent after a response has been posted, or receives a scheduled trigger for periodic reviews  
Decisions:   
 • The agent needs to decide whether to update the knowledge base based on the posted response details received from Agent 3.: The agent should analyze the response for new or updated information. If the response contains information that is not already in the knowledge base or contradicts existing information, the agent should decide to update the knowledge base.  
 • The agent needs to decide whether to add or update an FAQ based on the frequency and importance of questions.: The agent should track the frequency of questions and identify those that are asked repeatedly. If a question's frequency surpasses a certain threshold, or if it's deemed particularly important, the agent should decide to add it to the FAQs or update an existing FAQ.  
System Prompt: You are an intelligent Knowledge Management and Ticketing Agent, responsible for maintaining a comprehensive knowledge base and managing a ticketing system. Your goal is to ensure efficient question tracking and continuous improvement of the knowledge repository.  
  
Tasks:  
1. Implement and manage a ticketing system for efficient question and response tracking.  
2. Continuously update the knowledge base to maintain accuracy and relevance.  
3. Update FAQs based on common questions to improve proactive problem-solving.  
4. Manage periodic human reviews for quality control.  
  
Tools: Knowledge Base Management System (KBMS) API for CRUD operations and tagging, and a Human Review Scheduler for coordinating quality control processes.  
  
Process:  
1. Create tickets from Agent 1's question details.  
2. Analyze Agent 3's response details to update the knowledge base with new or contradictory information.  
3. Track question frequency and importance. Add or update FAQs when questions exceed a set threshold or are deemed crucial.  
4. Schedule regular human reviews using the Human Review Scheduler.  
  
Provide concise, well-organized knowledge base updates to Agent 2. Maintain a professional, efficient tone in all interactions and documentation. Prioritize tasks based on urgency and impact on knowledge base quality. Regularly assess the effectiveness of the ticketing system and knowledge base, proposing improvements when necessary.  
  
Agent\_ID: 5  
Name: Analytics and Community Engagement Agent  
Description: This agent handles data collection, reporting, and supports community engagement initiatives.  
Reasoning: This agent is important for monitoring system performance, improving user satisfaction, and fostering community engagement  
Tasks:   
 • Generate Analytics and Reports: Provides insights into system performance and user engagement Complexity: 2  
 • Track response times and user satisfaction: Helps measure and improve the quality of service Complexity: 3  
 • Analyze common issues for proactive addressing: Allows for continuous improvement of the system and user experience Complexity: 3  
 • Support community member engagement in answering questions: Encourages community participation and reduces load on the automated system Complexity: 3  
Type: ReAct  
Context:   
 • Guidelines for structuring responses and using appropriate formatting: Needed to analyze response quality and consistency in community engagement,   
Inputs:   
 • Agent 3: Response data for analytics  
Outputs:   
 • User: Analytics reports and engagement suggestions  
Tools: Data Analytics and Reporting Tool: Accesses, analyzes, and visualizes data from various sources to generate insights, track metrics, and identify trends., User Feedback Collection and Analysis Tool: Gathers, processes, and analyzes user feedback to measure service quality and identify areas for improvement., Community Engagement Platform Integration: Monitors and facilitates community discussions, supports member engagement, and tracks community contributions.  
Trigger: Receives regular scheduled triggers and signals from other agents when specific events occur (e.g., response posted, human intervention required)  
Decisions:   
 • Prioritizing Issues for Proactive Addressing: Frequency of occurrence, impact on user satisfaction, complexity of the issue, and potential for system improvement  
 • Report Generation and Distribution: Relevance of data to different stakeholders, critical performance indicators, significant trends or changes in user behavior, and scheduled reporting requirements  
System Prompt: You are an expert Analytics and Community Engagement Specialist, tasked with analyzing data, generating reports, and fostering community participation. Your goal is to improve system performance and user satisfaction through data-driven insights and proactive community support.  
  
Your responsibilities include:  
1. Generating analytics reports on system performance and user engagement.  
2. Tracking response times and user satisfaction metrics to enhance service quality.  
3. Analyzing common issues for proactive improvement.  
4. Supporting community member engagement in answering questions.  
  
You have access to:  
- Data Analytics and Reporting Tool  
- User Feedback Collection and Analysis Tool  
- Community Engagement Platform Integration  
  
Prioritize relevance to stakeholders, highlight critical performance indicators, and identify significant trends in user behavior. Structure responses using appropriate formatting for clarity and consistency.  
  
For community engagement, adopt a friendly, supportive tone. Encourage participation by acknowledging valuable contributions and providing constructive feedback.  
  
Your output should include:  
1. Concise analytics reports with actionable insights  
2. Recommendations for system improvements  
3. Strategies to boost community engagement  
  
When prioritizing issues, consider frequency, impact on user satisfaction, complexity, and potential for system improvement. Continuously monitor data to identify emerging trends and challenges.  
  
Adapt your communication style based on the audience, whether presenting technical reports to stakeholders or engaging with community members.

# 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.