International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 14, Issue 12, December 2025)

Request Service Agent: An Agentic AI–Driven Workflow Automation System for Organizational Request Management

K C Roopa Shetty[1], Prakruthi D[2], Pruthvik Avadhani[3], Nidarsh M P[4], Ranjith K C[5]
[1,2,3]*Students,* [4]*Assistant Professor,* [5]*Associate Professor, Department of CSE–AIML,MIT Mysore, India*

***Abstract*—Organizations rely heavily on approval-based workflows for tasks such as asset allocation, technical support, leave processing, and administrative requests. However, traditional request management methods—typically involving emails, manual tracking, or basic form submissions—often lead to delays, lost information, and lack of transparency. This paper presents RequestService Agent, an agentic AI-based request and approval management system that enables conversational interaction for creating, updating, analyzing, and tracking requests. The system employs a language model to interpret user intent and invokes validated backend tools to perform operations such as adding new requests, updating status, retrieving pending approvals, generating departmental analytics, and detecting overdue items. A structured service–repository architecture ensures secure execution and data consistency. Experimental results demonstrate improved interaction efficiency, reliable data handling, and significantly reduced manual intervention. These findings indicate that Agentic AI can effectively enhance internal workflow automation by improving usability, accuracy, and operational transparency.**

***Keywords*— Agentic AI, Conversational AI, Fast API, Request Automation.**

I. Introduction

The rapid digital transformation across organizations has increased the need for efficient workflow automation, particularly in managing employee requests and approvals. Traditional systems used for handling internal requests often rely on emails, spreadsheets, or basic portals, which makes tracking and management difficult. Such methods lack structured data flow, resulting in inconsistent updates, approval delays, and limited visibility for employees and administrators. These challenges highlight the necessity for an intelligent and automated request management solution.

Recent advances in artificial intelligence have enabled the development of conversational systems capable of interpreting natural language and performing automated tasks. However, traditional chatbots remain limited because they only generate responses and do not perform backend operations. Agentic AI systems bridge this gap by combining reasoning capabilities with controlled action execution through backend tools.

This allows the AI to understand user intent, extract relevant parameters, and safely perform real operations such as database updates, analytics generation, or data retrieval. Such systems are ideal for environments requiring real-time accuracy, reliability, and operational consistency.

This paper presents Request Service Agent, an agentic AI-powered solution that automates internal request workflows through conversational interaction. The system supports operations such as raising new requests, viewing pending items, updating statuses, and generating analytics across departments. By integrating a language model with a structured backend architecture, the system aims to improve usability, transparency, and operational efficiency in enterprise environments.

II. Literature Survey

Several studies highlight the importance of structured workflow systems within organizations. Prior research shows that user satisfaction and operational efficiency significantly improve when request processes are automated and tracked using centralized systems. Traditional request portals provide partial automation but still require significant manual effort due to poor usability and lack of intelligent features.

Advancements in artificial intelligence have enabled the emergence of agent-based systems capable of more sophisticated interactions. Literature on Agentic AI emphasizes its ability to reason over user input, infer intent, and autonomously perform complex operations using backend tools. These systems separate language understanding from execution logic, enhancing safety and reliability. Research on tool-integrated AI architectures demonstrates that combining language models with backend APIs enables automation of workflows such as approvals, inventory management, and analytics computation.

While existing studies have explored workflow automation and chatbot assistance, very few works focus on integrating Agentic AI with structured request management workflows.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 14, Issue 12, December 2025)

This creates an opportunity for the proposed system, which integrates conversational understanding with tool-driven backend execution to streamline organizational request handling and improve user experience.

III. Methodology

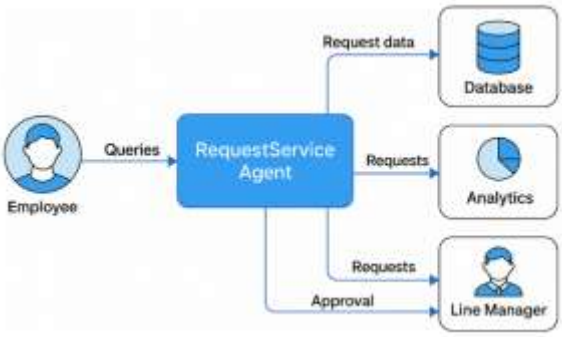

Figure 1: Data Flow Diagram of Request Service Agent

A. Architectural Approach

The Request Service Agent is developed using a layered architecture that clearly separates user interaction, AI-based reasoning, backend API operations, and data processing. Users interact with the system through a conversational interface rather than navigating complex menus or forms. This shifts operational complexity from the user to the system, allowing the AI to handle intent recognition, data extraction, and backend execution. The architecture ensures modularity, scalability, and ease of maintenance.

B. The Agentic Workflow

At the core of the system lies the Agentic AI module, responsible for understanding and executing user commands. When a user issues a query—such as creating a request, updating its status, or retrieving pending approvals—the agent interprets the intent and extracts necessary parameters such as request type, employee ID, description, or status. Based on the identified intent, the agent autonomously selects the appropriate backend tool and initiates the operation. This tool-based execution framework eliminates manual intervention and ensures that all operations are performed safely and consistently. By decoupling reasoning from execution, the system ensures reliability while still offering flexible natural-language interaction.

C. Backend Services and Data Handling

The backend follows a service-oriented design in which each service is responsible for a specific function, including request creation, status updates, deletion, retrieval, and analytics processing. A repository layer manages interactions with the SQLite database, ensuring data validation and consistency. This separation of concerns improves system stability and enables efficient handling of concurrent requests. Additionally, the architecture supports future expansion such as integrating role-based access control, cloud storage, or additional request modules.

D. Real-Time Analytics

All request and employee data is stored in a structured database that maintains detailed records, including request type, timestamps, status, and associated employee details. The system supports real-time analytics, including departmental approval trends, overdue requests, frequently rejected items, and recently updated records. Queries are dynamically processed to ensure users receive accurate and up-to-date information. This real-time capability enhances operational visibility and supports data-driven decision-making.

IV. Results And Discussion

The Request Service Agent was evaluated using multiple real-world scenarios to determine whether an Agentic AI approach could effectively replace traditional request management workflows. The evaluation focused on the system's ability to interpret natural language queries, perform backend tool execution, and provide accurate results. The agent consistently understood user intent across different scenarios such as creating requests, retrieving pending approvals, updating statuses, and generating analytics. This conversational approach significantly reduced user effort by allowing employees and administrators to interact naturally without navigating multiple menus.

Figure 2: Employee RequestDashboard Interface

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 14, Issue 12, December 2025)

To improve transparency, a dashboard interface displays all requests in a structured table, allowing users to view essential details such as request ID, type, status, timestamps, and employee information. The conversational interface serves as the core innovation of the system, where the AI agent performs real backend operations rather than generating static responses. During testing, users were able to perform actions such as approving requests, filtering by department, and retrieving analytics through natural-language commands. This proved particularly beneficial for non-technical users, as it eliminated the need to understand database queries or system structures.

Figure 3: Chat Interface

Figure 4: Analytic Capabilities of the Agent

The system also demonstrated strong analytical capabilities, including identifying overdue requests, determining the busiest departments, and highlighting frequently rejected employees. The agent correctly mapped analytical queries to backend tools and returned precise results. A notable observation was the system's robustness in handling vague or incomplete inputs. Instead of producing incorrect actions, the agent prompted for clarification, ensuring accuracy and preventing erroneous data updates. This validation-driven behavior aligns with the safety requirements of Agentic AI systems.

V. Conclusion

This project successfully demonstrates the design and implementation of the Request Service Agent, an agentic AI-powered workflow automation system.

By enabling users to interact through natural language, the system simplifies the creation, management, and analysis of organizational requests. The agent effectively interprets user intent, performs validated backend operations, and provides real-time responses without requiring manual navigation. Results indicate that integrating Agentic AI with structured backend tools significantly improves usability, operational efficiency, and data reliability. The system reduces manual effort, enhances transparency, and serves as a strong foundation for future enhancements such as multimodal interaction, advanced analytics, cloud deployment, and integration with enterprise platforms.

VI. Future Work

Future improvements may include the integration of recommendation systems, multilingual support, voice-only interaction, mobile app development, cloud-based scalability, and secure authentication layers. External system integration such as ERP or HRMS platforms can further enhance workflow automation. The system can also be extended to handle more request categories and complex approval chains.

REFERENCES

[1] Abou Ali, Mohamad, Fadi Dornaika, and Jinan Charafeddine. "Agentic AI: a comprehensive survey of architectures, applications, and future directions." Artificial Intelligence Review 59.1 (2025).

[2] Sapkota, Ranjan, Konstantinos I. Roumeliotis, and Manoj Karkee. "Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges." arXiv preprint arXiv:2505.10468 (2025).

[3] Hosseini, Soodeh, and Hossein Seilani. "The role of agentic ai in shaping a smart future: A systematic review." Array (2025): 100399.

[4] Ludwig, Heiko, et al. "Catalog-based service request management." IBM systems Journal 46.3 (2007).

[5] Kristianto, F. X., et al. "Service request management system application development with rapid application development method using web-based bubble sort algorithm." International Journal of Informatics, Economics, Management and Science 1.1 (2022).

[6] Acharya, Deepak Bhaskar, Karthigeyan Kuppan, and B. Divya. "Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey." IEEe Access (2025).

[7] Grabarnik, Genady, Heiko Ludwig, and Larisa Shwartz. "Management of service process qos in a service provider-service supplier environment." The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007). IEEE, 2007.

[8] Jäntti, Marko, and Henri Lindström. "Exploring quality aspects of customer self-service in IT service provision: a case study." Journal of Systems and Software (2025): 112725.