

Article

Bridging ERP Complexity Through Retrieval-Augmented Generation: Design and Evaluation of an Intelligent Question-Answering System for SMEs

Pongsathon Pookduang  and Wirapong Chansanam * 

Department of Information Science, Faculty of Humanities and Social Sciences, Khon Kaen University, Khon Kaen 40002, Thailand; pongsathon.po@kkumail.com

* Correspondence: wirach@kku.ac.th

Abstract

Purpose/Background: Small and medium-sized enterprises (SMEs) that deploy Enterprise Resource Planning (ERP) systems face a persistent paradox: although ERP centralises organisational data, frontline users frequently lack the technical expertise to navigate complex menu structures, preventing efficient information retrieval for decision-making. This study presents a preliminary Research and Development (R&D) effort to design, build, and qualitatively evaluate a prototype intelligent question-answering system that connects to Odoo ERP through application programming interfaces (APIs) via Retrieval-Augmented Generation (RAG), enabling natural-language access to real SME business data. **Methods:** A single-organisation R&D case study was conducted at an SME in Khon Kaen Province, Thailand. The development cycle comprised problem analysis and requirement specification, system architecture design, prototype construction, integration and deployment, iterative testing and refinement, and multidimensional evaluation. The prototype was implemented with Chainlit (conversational interface), FastAPI (orchestration and tool-calling layer), and Odoo XML-RPC/JSON-RPC APIs (structured data retrieval). A fixed set of 20 test questions spanning four complexity levels (easy, moderate, complex, out-of-scope) was evaluated by three automated tools (OpenAI Evals, DeepEval, Ragas), by real-task verification against live ERP data, by five domain experts using 5-point Likert-scale questionnaires, and by three end users from the case-study organisation, who additionally completed the System Usability Scale (SUS). Given the very small expert and user samples, all human evaluation results are reported descriptively as preliminary, exploratory indicators rather than as statistically generalisable measures. **Results:** Automated evaluation achieved indicative pass rates of 95.00% (OpenAI Evals, 19/20), 90.00% (Ragas, 18/20), and 85.00% (DeepEval, 17/20). Descriptive expert feedback ($n = 5$) yielded an overall mean of 3.82 (high level), and descriptive end-user feedback ($n = 3$) yielded an overall satisfaction mean of 4.33 (highest level). The SUS score was 66.67/100, sitting at the boundary between ‘OK’ and ‘Good’ and revealing a divergence between high stated satisfaction and lower confidence in independent system use (item 9, raw mean = 2.33) and a stronger perceived need for expert assistance (item 4, raw mean = 2.67). These results are interpreted as preliminary diagnostic signals for further development rather than as confirmatory evidence. **Conclusions:** This preliminary R&D study suggests that a RAG-based ERP chatbot can meaningfully simplify ERP data access for SME users, while exposing persistent gaps in multi-step reasoning, user confidence, and data privacy boundaries that must be addressed in subsequent development cycles. The SUS pattern, in particular, suggests a ‘novelty effect’ in which users are enthusiastic about the natural-language interface yet remain anxious about correctness and stability during real tasks. **Originality/Value:** This work contributes



Academic Editor: Ming Liu

Received: 8 June 2026

Revised: 27 June 2026

Accepted: 1 July 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

a transparent, replicable preliminary R&D blueprint that combines (i) live API-mediated RAG over structured ERP data, (ii) a complementary multi-tool automated evaluation set (OpenAI Evals + DeepEval + Ragas), (iii) descriptive expert and end-user feedback, and (iv) SUS-based usability assessment, all documented for a single Thai SME using Odoo 18. The study explicitly positions itself as an early step toward larger, multi-site, and on-premise deployments of trustworthy ERP-integrated conversational agents.

Keywords: Retrieval-Augmented Generation; Enterprise Resource Planning; conversational AI; Odoo; small and medium-sized enterprises; question-answering systems

1. Introduction

The global ERP software market is projected to exceed USD 117 billion by 2030, yet industry surveys consistently report that a majority of ERP users in SMEs engage with fewer than one-third of available system functions [1]. This utilisation gap is not primarily a technological deficiency; it is an access-complexity problem. Most frontline employees in SMEs lack the time, training, or system familiarity to navigate layered module hierarchies, define multi-field search conditions, and synthesise results from several ERP screens—even when the required data are present in the system [2,3]. The consequence is that expensive, data-rich ERP investments deliver less value than they could, and data-driven decision-making remains the preserve of a small number of technically skilled staff.

Advances in Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) now offer a credible pathway toward democratising ERP data access. LLMs provide flexible natural-language understanding and fluent response generation, while RAG improves factual reliability by conditioning generation on retrieved external evidence and verifiable context [4,5]. When RAG is extended from document corpora to live structured databases exposed through APIs, the result is a system that can answer questions such as ‘Which products are low in stock this week?’ or ‘What are the total sales for January?’ by retrieving real-time ERP data and summarising it in plain language—without requiring the user to touch a menu.

Despite rapid growth in RAG research [5,6], three important gaps remain unaddressed: (a) most RAG applications target unstructured document corpora, and relatively few studies examine RAG over structured relational or API-based enterprise data sources [7]; (b) evaluation of RAG chatbots in SME ERP contexts has not been conducted using multi-tool automated assessment alongside practitioner and end-user validation; and (c) the full R&D design cycle—from organisational process analysis through prototype construction and multidimensional evaluation—has rarely been documented in a published study using real organisational ERP data.

This study addresses all three gaps through an R&D case study conducted in a Thai SME that uses Odoo ERP version 19.3 as its primary operational platform. Odoo was selected because it is an open-source ERP suite that covers core business applications and provides documented external API mechanisms for integration with XML-RPC and JSON-RPC services, while also supporting the products, sales, inventory, purchasing, and contacts modules relevant to typical SME decision-making queries [8,9].

The study is guided by three research questions: (1) What are the characteristics and limitations of the existing ERP data-access process in the case-study organisation? (2) How can an AI chatbot using RAG be designed to connect to Odoo ERP through APIs to support natural-language business data access? (3) To what extent can the developed prototype

correctly answer questions from ERP data, and how do automated, expert, and end-user evaluations compare in assessing system quality?

This article makes four preliminary contributions, which together frame the present manuscript as a documented design-and-evaluation blueprint rather than as a fully validated production system:

- A replicable R&D-grounded architecture for ERP-integrated RAG chatbots, specifying design decisions at the level of API integration, intent analysis, and tool-calling orchestration over live structured ERP data.
- Formal as-is and to-be process models documenting the transition from a conventional menu-driven ERP access process to a natural-language question-driven process, with an explicit five-step to two-step workflow reduction.
- A four-level test-question taxonomy (easy, moderate, complex, out-of-scope) that operationalises complexity for systematic RAG-ERP evaluation and that can be reused by other studies as a starting benchmark.
- A practical demonstration of the complementary diagnostic value of simultaneously applying three automated RAG evaluation tools (OpenAI Evals, DeepEval, Ragas) alongside descriptive expert and end-user assessments, supplemented by a SUS usability check.

The remainder of the paper is organised as follows. Section 2 presents the theoretical and conceptual background covering IS success, Task-Technology Fit, conversational agent adoption, and the RAG paradigm. Section 3 reviews related literature on ERP data-access challenges in SMEs, enterprise conversational AI, RAG over structured data, and RAG evaluation frameworks. Section 4 describes the methodology, including the R&D development cycle, case-study context, system architecture, test-question design, and evaluation protocol. Section 5 reports the results across the four evaluation dimensions. Section 6 discusses principal findings, theoretical and practical implications, methodological implications, and comparison with prior work. Section 7 details limitations and future research directions, and Section 8 concludes.

2. Theoretical and Conceptual Background

This study is grounded in three converging streams: information systems success theory, the RAG paradigm, and enterprise chatbot adoption research.

2.1. Information Systems Success and Task-Technology Fit

Contemporary ERP success research drawing on the DeLone and McLean IS Success Model indicates that system quality, information quality, service quality, use, user satisfaction, and net benefits remain central dimensions for assessing ERP success [10,11]. In an ERP context, this model implies that even a technically sound system will fail to deliver value if the information-quality dimension—in terms of accuracy, timeliness, and accessibility—is insufficient. The prototype in the present study directly targets information quality: by retrieval-grounding answers in real ERP data, it aims to raise both accuracy and accessibility.

Recent Task-Technology Fit (TTF) research on AI-based chatbots indicates that technology utilisation depends on the fit between task requirements, user characteristics, and technology capabilities [12]. ERP data-access tasks performed by non-technical SME staff are poorly matched to conventional menu-driven interfaces but are well matched to conversational question-answering systems. The end-user evaluation component of the present study can therefore be read as a TTF operationalisation: high user satisfaction scores suggest strong Task-Technology Fit, while the SUS score reveals residual fit gaps related to user confidence and system stability.

2.2. Conversational Agent Adoption in Organisational Contexts

Adam et al. [13] showed in a large-scale experiment that users' willingness to adopt chatbots is influenced by perceived service quality and expected task fit. Their findings suggest that chatbots with access to accurate, domain-specific information are more likely to be adopted than general-purpose bots. Recent chatbot adoption and trust studies similarly indicate that perceived information quality, system competence, transparency, and trustworthy interaction design influence user trust and adoption intention [14,15]. For RAG-based chatbots specifically, source attribution and grounding have been shown to affect perceived trust and transparency, directly motivating the API-mediated RAG approach in this study [16].

2.3. The Retrieval-Augmented Generation Paradigm

Contemporary RAG literature describes RAG as a framework that augments LLM generation with retrieved external evidence at inference time, improving factual grounding while introducing trade-offs related to retrieval quality, grounding fidelity, latency, and robustness [4,5]. Subsequent architectural refinements—Fusion-in-Decoder [17], in-context RAG [18], and Self-RAG [6]—establish RAG as a design space rather than a monolithic technique. Gao et al. [5] categorise approaches as naive RAG, advanced RAG, and modular RAG, identifying structured data retrieval as an important frontier.

Agentic RAG architectures have emerged in which the language model orchestrates retrieval, tool use, planning, and reasoning across multiple steps [19,20]. Singh et al. [21] provide a 2026 survey of agentic RAG that systematically categorises agent cardinality, control structure, autonomy, and knowledge representation, and identifies tool-calling over structured enterprise data as one of the most active and least mature design spaces. Recent enterprise-focused work [22] further argues that hybrid agentic patterns combining tool calls to structured APIs with traditional document retrieval outperform either approach alone for compliance-sensitive deployments. The prototype in the present study sits in this design space: it implements a simplified agentic RAG pattern in which the LLM selects among Odoo API endpoints based on question intent, executes the relevant call, and grounds its response in the returned structured data. This design trades embedding-based retrieval flexibility for higher factual accuracy—a rational trade-off in a domain where data precision is critical.

For evaluation, Es et al. [23] formalised four RAG quality dimensions in the Ragas framework: faithfulness, answer relevancy, context precision, and context recall. DeepEval documentation describes LLM-as-a-judge metrics such as G-Eval for evaluating outputs against custom criteria, while Saad-Falcon et al. [24] proposed ARES as a complementary evaluation framework [24,25]. These frameworks collectively inform the multi-tool evaluation design of the present study.

3. Related Literature

3.1. ERP Adoption and Data-Access Challenges in SMEs

ERP systems integrate core organisational processes into a unified data architecture, reducing redundancy and supporting cross-functional reporting [26]. For SMEs, cloud-hosted and open-source solutions such as Odoo have reduced the total cost of ownership, making ERP accessible to organisations without dedicated IT departments [2,3,9]. Recent ERP implementation reviews identify user competence, training, management support, organisational readiness, and usability as persistent determinants of ERP success and failure in SMEs and other organisational contexts [27,28]. Rutz et al. [29], studying ERP appropriation in SMEs, similarly found that ERP systems remain underutilised when users lack adequate training and understanding. Zhang et al. [30] reviewed AI-augmented

ERP systems and identified natural-language interfaces as a high-priority development direction, noting that current ERP vendors' NLP capabilities remain immature for SME deployment contexts.

3.2. Conversational AI and LLM-Based Assistants in Enterprise Contexts

The evolution of enterprise chatbots from rule-based systems [31] to LLM-powered assistants [32,33] has dramatically expanded the range of questions these systems can handle. Achiam et al. [32] demonstrated that GPT-4 and successors can follow complex multi-step instructions and generate structured outputs. Adam et al. [13] found that perceived service quality was the primary driver of continued chatbot adoption, while novelty effects faded rapidly, underscoring the importance of factual correctness over fluency. Ravi et al. [16] showed that trust and transparency in RAG systems are affected by grounding and source-use design, a principle implemented in the present study through RAG grounding.

3.3. RAG Architectures and Extension to Structured Enterprise Data

Contemporary RAG architectures generally combine retrieval and generation components while differing in how retrieval evidence is selected, filtered, structured, and coordinated with the generator [4,5]. Subsequent advances [5,6,17,18] establish a rich design space. Wang et al. [7] examined RAG over tabular data and showed that API-mediated retrieval substantially outperforms embedding-based retrieval over serialised structured text for factual numerical queries—providing theoretical support for the present study's API-first design. Chen et al. [20] benchmarked multiple LLMs in structured data RAG settings and found faithfulness to be the most sensitive performance dimension for enterprise use cases.

Studies on LLM-based text-to-SQL [34,35] demonstrate that LLMs can generate database queries from natural-language questions with increasing accuracy, but these approaches require database access permissions that may be restricted in cloud ERP deployments—a constraint motivating the API-based retrieval design of the present study. Zhang et al. [30] specifically identified ERP + RAG conversational chatbots as an important understudied area, calling for empirical studies with real deployment data.

3.4. Evaluation Frameworks for RAG and Intelligent Question-Answering Systems

Systematic evaluation of RAG systems requires multidimensional assessment. Es et al. [23] formalised faithfulness, answer relevancy, context precision, and context recall in the Ragas framework. DeepEval documentation describes LLM-as-a-judge metrics for custom output evaluation, including G-Eval-style assessment [25]. Saad-Falcon et al. [24] proposed ARES. Gan et al. [36] provide the most comprehensive recent survey of RAG evaluation, identifying multi-tool combination as best practice. For usability assessment, recent SUS benchmarking studies confirm the continuing utility of SUS and provide empirical guidance for interpreting SUS scores [37,38]. Koo [39] summarises the use of Likert-type scales for measuring perceptions and attitudes in research contexts.

3.5. Synthesis and Research Gap

Table 1 positions the present study relative to closely related prior work.

No prior study has combined: (i) live API-mediated RAG over structured ERP data, (ii) simultaneous multi-tool automated evaluation across three complementary RAG assessment frameworks, (iii) SUS-based usability assessment with real SME employees, and (iv) a complete R&D-grounded design cycle documented in a published study using real organisational ERP data. The present study fills this compound gap.

Table 1. Synthesis of related studies and positioning of the present study.

Study	Data Source	Evaluation	Domain	SME/Real Data
Sharma [4]	Survey (RAG)	N/A	RAG architecture survey	No/No
Gao et al. [5]	Survey (RAG)	N/A	RAG survey	No/No
Asai et al. [6]	Unstructured docs	Benchmarks	Open-domain QA	No/No
Wang et al. [7]	Tabular data	Automated	Structured QA	No/No
Adam et al. [13]	N/A	Experiment	Service chatbot	No/Simulated
Ravi et al. [16]	RAG responses	User study	Trust/transparency	No/Simulated
Zhang et al. [30]	ERP review	Review	ERP + AI	SME/No
Present study	Live ERP API (Odoo)	3 auto tools + Expert + SUS/Likert	ERP chatbot (RAG)	Yes/Yes

4. Methodology

4.1. Research Design

This study adopts a Research and Development (R&D) methodology [40] combined with a single-organisation case study [41]. R&D is appropriate when the research goal is to design, develop, and evaluate a prototype system in a real-world context, with iterative cycles of development, testing, and improvement [40]. Case-study methodology is appropriate when questions ask ‘how’ and ‘why’ about a contemporary phenomenon in its real-life context and when the researcher does not exercise experimental control over events [41,42].

The development cycle comprises six phases: (1) problem analysis and requirement specification; (2) system architecture design; (3) prototype development; (4) system integration and deployment; (5) iterative testing and refinement; and (6) multidimensional evaluation. This hybrid R&D approach draws on SDLC for systematic analysis and design, the Prototyping Model for rapid functional prototype construction [43,44], and Agile/Scrum principles for iterative improvement based on stakeholder feedback [45].

Figure 1 visualises the six phases of the R&D development cycle as adopted in this study. The cycle is iterative: insights from evaluation feed back into requirement refinement and architecture revision, supporting incremental improvement across successive prototype iterations.

**Figure 1.** R&D development cycle followed in this study (six phases, iterative).

4.2. Case-Study Context and Data Sources

The case-study organisation is an SME in Khon Kaen Province, Thailand, operating in the distribution sector, that uses Odoo 18 Community Edition as its primary ERP platform. Real ERP data from five business modules—products, sales, inventory, purchasing, and Contacts/Partners—served as the primary data source for the AI chatbot through API connections. The use of real ERP data is methodologically central: it allows system testing to reflect actual organisational data quality, schema complexity, and data volume—conditions that simulated data cannot replicate [41,42]. The researcher had direct familiarity with the organisation's ERP deployment, supplementing user interviews with first-hand process knowledge. Participant data were handled in accordance with ethical data protection principles; no personally identifiable financial data were reported.

4.3. As-Is Process Analysis

The existing ERP data-access process was documented through direct inquiry with ERP users and relevant personnel, including the sales manager, ERP users, and staff responsible for product, sales, inventory, purchasing, and partner data, supplemented by the researcher's own experience within the case-study organisation. The as-is process follows a consistent pattern: (1) log into Odoo, (2) navigate to the relevant module, (3) apply filter/search conditions, (4) read results, and (5) manually interpret and aggregate data. Three structural limitations were identified: menu complexity (users frequently could not identify which module contained required data), information search complexity (filter fields use system-defined terminology different from everyday business language), and interpretation burden (cross-module queries required manual data export and aggregation, taking 20–45 min per query).

4.4. To-Be Process Design

The to-be process replaces menu-driven navigation with a natural-language conversation through an Agentic RAG Chatbot that can analyse question intent, select the relevant API endpoint, retrieve structured ERP data, and ground the LLM response in that data [19,20]. The chatbot acts as an AI intermediary layer between the user and the ERP system, not replacing the ERP but making its data more accessible. For direct queries, the workflow is reduced from five steps to two (ask → read answer). For summary queries, manual aggregation is eliminated. The primary data source is the PostgreSQL database underlying Odoo ERP, accessed via Odoo's XML-RPC/JSON-RPC APIs, which provides controlled, scoped data access without exposing the database layer directly.

4.5. System Architecture and Prototype Development

User interface layer: Chainlit was used to provide a chat-style conversational interface accessible via a web browser. Chainlit is an open-source Python package for building production-ready conversational AI, and its documentation describes user sessions, authentication, streaming, and other features useful for conversational prototypes [46,47].

Orchestration layer: FastAPI handles API service management, supporting asynchronous operations via Python 3.14.3's `async/await` pattern, which is important for concurrent ERP API calls and LLM inference [48]. The LLM receives the user's question and a set of tool descriptions corresponding to Odoo API endpoints. The model selects which tool(s) to invoke based on question intent, providing the necessary parameters. This tool-calling design implements a simplified agentic RAG architecture.

Retrieval layer: Each tool calls a specific Odoo XML-RPC or JSON-RPC API endpoint using authenticated credentials. Five tool categories are implemented, covering: product information, sales orders (`sale_order`, `sale_report`), inventory stock, purchase orders, and

partner/customer records (partners/contacts). Each API call returns structured JSON data serialised into compact text before being injected into the LLM context. A guard condition handles empty retrieval results by instructing the LLM to indicate data unavailability rather than generating speculative answers. The system also uses `psycopg/asynpg` for direct PostgreSQL access in certain tool paths, and `redis` for session caching.

Generation layer: A Large Language Model (LLM) serves as the backbone for intent analysis and answer generation. The prompt template comprises: (1) a system instruction specifying role, data scope, response format, and refusal behaviour for out-of-scope questions; (2) conversation history for contextual continuity; (3) retrieved ERP data as structured context; and (4) the user question. The system prompt explicitly instructs the model to base all numerical claims on the retrieved context and to decline questions requiring data outside the defined API scope.

Multi-step orchestration mechanism. For Level 3 (complex) questions that require data from multiple Odoo modules, the orchestration layer does not assume that a single endpoint will be sufficient. Instead, it executes the tool-calling loop iteratively. The LLM is provided with the full set of available tool schemas (one per Odoo API endpoint) and, after parsing user intent, may emit one or more sequential tool calls. After each call, the FastAPI orchestrator returns the structured result back to the LLM, which then decides whether further calls are required before producing a final answer. The loop terminates when the model emits a textual response rather than a further tool call, or when a configurable maximum number of tool calls is reached. A guard condition catches empty retrievals: if any tool returns no records, the system prompt instructs the model to acknowledge data unavailability rather than to attempt to fabricate an answer.

Figure 2 illustrates this orchestration sequence using a representative complex query—‘Show low-stock products and their January sales’—which the LLM decomposes into two sequential tool calls (an inventory search followed by a sales-by-product lookup) before assembling a grounded response.

Data privacy boundaries in the prototype. The current prototype architecture relies on an external cloud-hosted LLM, and live Odoo data passes through this external API at answer-generation time. Several enterprise-compliance and data-leak risks therefore exist by construction, not merely as a peripheral limitation. To make these boundaries explicit and actionable for follow-on work, three architectural mitigations are envisaged for the next iteration: (i) a privacy-filtering proxy in front of the cloud LLM that strips or tokenises personally identifiable fields (e.g., partner names, contact details) and commercially sensitive fields before they leave the SME’s perimeter; (ii) role-based access control at the API tool layer, so that tool invocations are scoped to the calling user’s ERP role and only retrieve fields the user is authorised to see; and (iii) a locally deployable open-source LLM alternative—for example, recent Llama 3, Qwen 2.5, or Mistral families—hosted on-premise, so that ERP data never leaves the SME’s network. Recent feasibility analyses argue that the latter is increasingly tractable on commodity GPU hardware for SME-scale workloads [22,49]. In the present preliminary study, the cloud-hosted LLM is retained for engineering convenience and evaluator accessibility; the privacy mitigations above are treated as core architectural considerations for the next development cycle rather than as afterthoughts.

The prototype was developed in Python and deployed on Railway as a web application. The architecture comprises four layers, as shown in Figure 3.

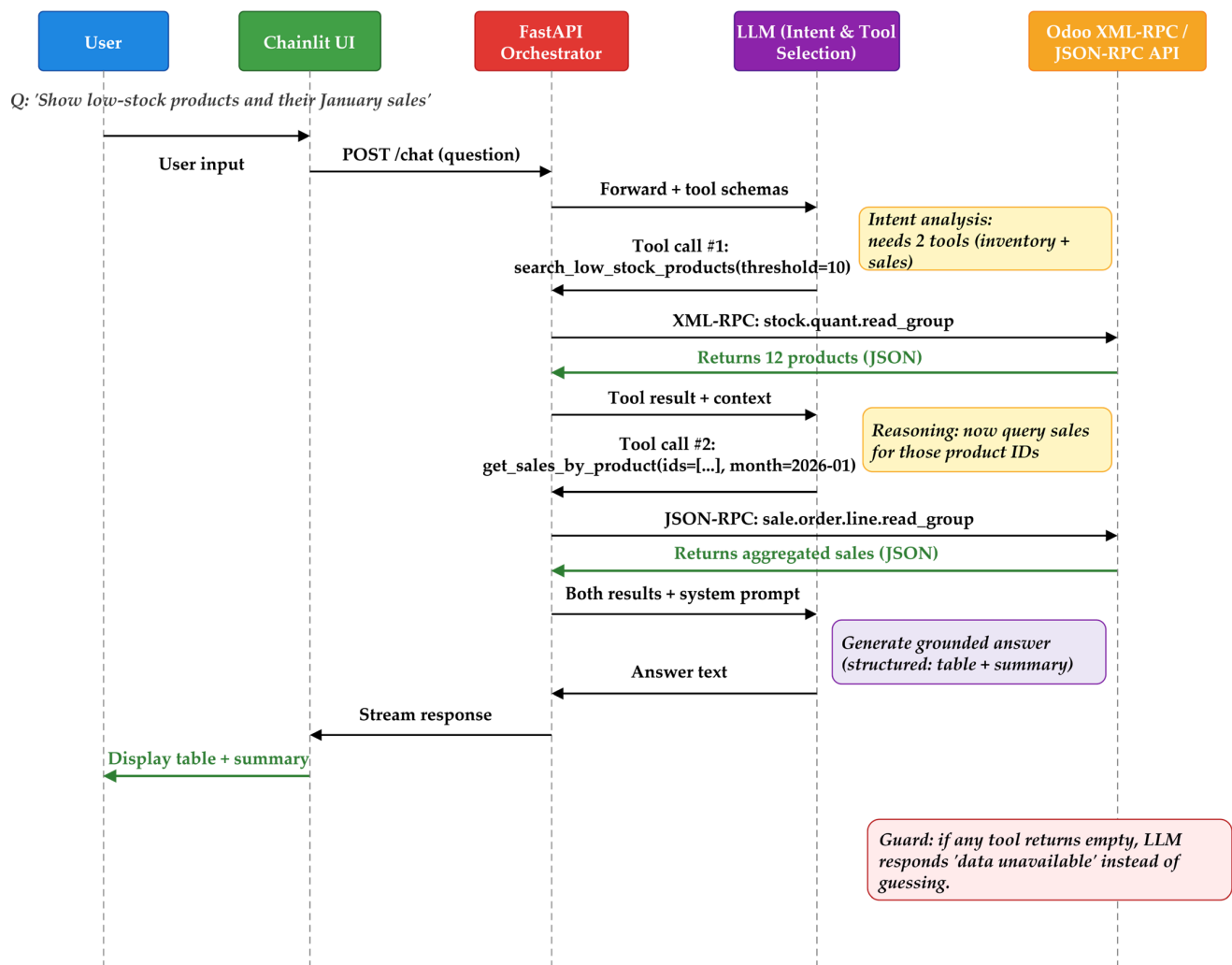


Figure 2. Sequence diagram illustrating the interaction among the user, Chainlit UI, FastAPI orchestrator, LLM, and Odoo API. Black arrows represent requests, function invocations, and internal processing steps, whereas green arrows indicate returned data or responses from the Odoo API. The symbol “#” denotes the sequential order of tool invocations (e.g., Tool call #1 is executed before Tool call #2).

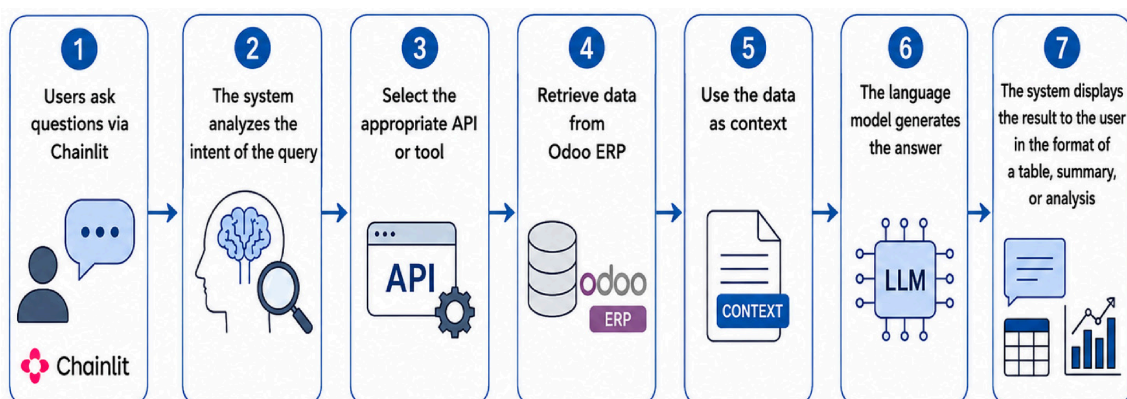


Figure 3. Overall workflow of the proposed LLM-ERP question-answering system. The numbered boxes indicate the sequential processing stages from user query submission to answer presentation. The blue arrows represent the direction of the workflow and data flow between consecutive stages. The icons (e.g., API, database, context document, LLM, and report) visually represent the primary function performed at each stage rather than specific software components.

4.6. Test-Question Design

Twenty test questions were designed based on real business use cases representative of the case-study organisation’s actual decision-making information needs. Questions were reviewed by the researcher to ensure coverage across four complexity levels (Table 2).

Table 2. Test-question design taxonomy for ERP chatbot evaluation.

Level	Characteristics	Key Metrics	<i>n</i>	Example
Easy	Single-fact direct retrieval	Answer relevancy, faithfulness, Tool Call Accuracy	6	‘How many products are currently in stock?’
Moderate	Aggregation, filtering, or time-scoped retrieval	Answer relevancy, faithfulness, context recall/precision	6	‘What is the total sales value for January 2026?’
Complex	Multi-condition synthesis or multi-tool reasoning	Faithfulness, context recall/precision, Multi-tool Accuracy, Agent Goal Accuracy	5	‘Which products have high sales but no pending purchase orders?’
Out-of-scope	Questions outside defined ERP data domains	Domain Adherence, Hallucination Avoidance	5	‘What is the weather forecast for Khon Kaen tomorrow?’

Note: The 3-question out-of-scope count in the automated evaluation table (Table 3) reflects the subset included in the 20-question automated test set; the full question taxonomy includes five out-of-scope test scenarios. All 20 questions were fixed prior to evaluation and used identically across all three automated evaluation tools.

Table 3. Indicative comparison of as-is menu-driven workflow versus to-be conversational workflow for a representative cross-module query (Khon Kaen SME context, timed walk-throughs).

Aspect	As-Is (Menu-Driven)	To-Be (Conversational)
User interaction steps	5 steps (login → module → filter → results → manual aggregation)	2 steps (ask in natural language → read grounded answer)
Approximate completion time *	20–45 min (cross-module aggregation)	Under 30 s (single-turn response)
Required prior knowledge	Module structure, field/filter terminology, manual aggregation logic	Plain business question phrasing
Success rate in walk-throughs **	Highly variable: 2 of 5 walk-throughs completed without researcher assistance	5 of 5 walk-throughs returned an answer; 4 of 5 were verified correct
User-perceived effort (qualitative)	High—repeated context switching across screens	Low—single conversational turn

4.7. Multidimensional Evaluation Protocol

The evaluation protocol captures system quality from four complementary perspectives [36,50].

Dimension 1—Automated technical evaluation: Three tools were applied to the 20-question test set. (a) OpenAI Evals assessed answer correctness against predefined expected outputs using a rule-based approach that reports pass/fail per test case [51,52]. (b) DeepEval assessed semantic answer quality using the LLM-as-a-judge approach, evaluating coherence, relevance, and completeness [25,53]. (c) Ragas evaluated four RAG-specific quality dimensions: faithfulness, answer relevancy, context precision, and context recall [23,54]. Using three tools simultaneously reveals failure modes invisible to any single metric [36].

Dimension 2—Real test-case verification: Each system response was manually verified against actual Odoo ERP data by the research team, providing human-validated ground truth for correctness and identifying specific failure cases.

Dimension 3—Expert assessment: Five domain experts evaluated the prototype using a structured questionnaire with 5-point Likert-scale items (1 = Strongly disagree; 5 = Strongly agree). The expert panel comprised specialists in information systems, AI/chatbot development, ERP, and organisational IS, with experience ranging from approximately 4 to 20 years, reflecting both academic and practitioner perspectives. The questionnaire covered five evaluation dimensions (see Table 4). Data were analysed using descriptive statistics (mean, SD, Median, IQR), with Likert-type scales treated as structured perception measures suitable for descriptive interpretation [39].

Table 4. Automated evaluation results by tool.

Evaluation Tool	Total Test Cases	Passed	Failed	Pass Rate (%)	Primary Evaluation Dimension
OpenAI Evals	20	19	1	95.00	Answer correctness against predefined conditions
Ragas	20	18	2	90.00	Q-context-answer relationship (faithfulness, relevancy, precision, recall)
DeepEval	20	17	3	85.00	Semantic answer quality (LLM-as-judge)

Dimension 4—End-user assessment: Three SME staff members from the case-study organisation (store manager and sales staff) participated in end-user evaluation after a brief orientation on the chatbot’s scope. This sample size is consistent with formative usability evaluation and case-based prototype research, where small purposively selected samples can surface actionable usability issues, while being clearly inadequate for statistical generalisation [55–57]. All end-user findings are therefore reported as descriptive, exploratory signals consistent with a preliminary R&D study, not as confirmatory measurements. Participants completed two instruments: (a) a 5-point Likert satisfaction questionnaire covering five dimensions (Table 5) and (b) the 10-item System Usability Scale (SUS), providing a standardised usability score on a 0–100 scale [37,38]. We acknowledge that 20 test cases provide an indicative rather than exhaustive probe of system behaviour; expanding the test set to several hundred questions, spanning additional industries, ERP modules, and natural-language phrasings, is identified as a priority in Section 7.

Table 5. Expert evaluation results ($n = 5$; 5-point Likert scale).

Evaluation Dimension	Mean	SD	Median	IQR	Level
D1: Answer correctness and reliability	3.92	1.26	4.00	2.00	High
D2: Appropriateness of system operation	3.72	1.49	4.00	2.00	High
D3: Usefulness for business operations	3.72	1.37	4.00	2.00	High
D4: Feasibility of practical implementation	3.88	1.51	5.00	2.00	High
D5: Quality of system response in RAG dimensions	3.85	1.23	4.00	2.00	High
Overall	3.82	1.36	4.00	2.00	High

Note: Scale anchors: 4.21–5.00 = highest; 3.41–4.20 = high; 2.61–3.40 = moderate; 1.81–2.60 = low; 1.00–1.80 = lowest [39]. The relatively high SD values (1.23–1.51) reflect genuine expert opinion variability across the 5-point scale for a 5-person panel, a common characteristic of small-sample formative evaluation [56].

5. Results

5.1. As-Is Process Analysis Findings

The as-is process analysis confirmed three structural barriers to ERP data access. Menu complexity required users to navigate five to seven modules depending on query type; users frequently could not identify which module contained the required data. Information search complexity arose because Odoo's filter interface uses system-field terminology (e.g., `sale.order.partner_id.name`) rather than the business language familiar to users. Interpretation burden was most pronounced for cross-module queries, with users reporting 20–45 min to assemble data for a typical management question. These findings confirm that ERP centralises data without democratising data access—the motivating gap for the chatbot intervention.

5.2. To-Be Process Design Findings

The to-be process model reduces the data-access workflow from five steps to two steps (ask → read answer) for direct retrieval queries. For summary queries requiring multi-module data, the chatbot's agentic tool-calling architecture eliminates the manual aggregation step entirely. Table 3 (as-is vs to-be comparison) in the thesis confirmed key transitions: access method changed from ERP menu navigation to natural-language question; required user knowledge changed from menu/module familiarity to no prior system knowledge; number of steps reduced from multi-step to ask-and-read; result format changed from raw ERP list data to natural-language text, table, or analytical summary.

To make this workflow reduction more concrete, Table 3 contrasts the approximate effort required for a representative cross-module query ('Which products are low in stock and what were last month's sales for them?') under the as-is menu-driven workflow and the to-be conversational workflow, drawn from the as-is process analysis and from researcher-led timing trials with the prototype. The figures are indicative and based on a small number of timed walk-throughs; they are intended to characterise order-of-magnitude differences rather than to support inferential comparisons.

5.3. Prototype Development

The developed prototype is a web application deployed on Railway (<https://gentle-caring-production.up.railway.app/> (accessed on 15 March 2026)), and is accessible through a login screen followed by a Chainlit chat interface. Users authenticate and then submit natural-language questions about products, sales, inventory, purchase orders, and partner data. The system handles out-of-scope questions through polite refusal messages explaining its scope boundaries. Figures 4–6 illustrate representative system outputs.

Which month had the highest sales and which month had the lowest sales from January to March 2026?



Based on the sales data from January 1, 2026, to March 31, 2026, the summary is as follows:

- **The month with the highest sales was:** January 2026.
- **The month with the lowest sales was:** February 2026.

Please let me know if you would like a detailed calculation of the total sales for each month.

Figure 4. Example of an analytical answer generated by the prototype. **Note:** The pink four-petal symbol is the Chainlit logo and denotes responses generated by the AI assistant.

Please provide detailed information.

 The monthly sales summary for Wine O'Clock Khon Kaen from January to March 2026 is as follows:

Month	Total (Baht)
January 2026	
February 2026	
March 2026	

- **January 2026** had the highest sales with a total of **THB.**
- **February 2026** had the lowest sales, with a total of **THB.**

Note: These sales figures were calculated from the sales transaction records in the store's system. Please feel free to ask if you would like more specific details about products or customers.

Figure 5. Example of text and tabular summary output. **Note:** The pink four-petal symbol is the Chainlit logo and denotes responses generated by the AI assistant.

Will it rain tomorrow ?

 Sorry, this question is outside the scope of the store's supported data. The system can only provide information related to wine products, inventory, sales, and reports for Wine O'Clock Khon Kaen.



Figure 6. Example of out-of-scope question handling. **Note:** The pink four-petal symbol is the Chainlit logo and denotes responses generated by the AI assistant.

5.4. Automated Technical Evaluation Results

Table 4 presents the automated evaluation results. Overall pass rates were 95.00% (OpenAI Evals, 19/20), 90.00% (Ragas, 18/20), and 85.00% (DeepEval, 17/20).

The sub-category pattern confirmed that easy and moderate questions were handled well across all three tools. The single OpenAI Evals failure and two Ragas failures occurred in complex-level questions. The three DeepEval failures were distributed across moderate ($n = 1$) and complex ($n = 2$) questions, reflecting its more demanding semantic completeness criterion. Out-of-scope questions were handled correctly in all cases across all three tools. The single OpenAI Evals failure was attributed to incomplete multi-step synthesis for a complex cross-module question. Ragas failures reflected incomplete context recall rather than factual errors. DeepEval failures indicated adequate factual content but insufficient semantic completeness. Real test-case verification confirmed that no answer failure was attributable to LLM hallucination; all failures arose from retrieval limitations (API scope constraints) or multi-step reasoning gaps.

These pass rates compare favourably with published RAG systems benchmarks. Chen et al. [20] reported factual correctness rates of 78–91% for LLMs in structured data RAG settings; the present study's 95.00% OpenAI Evals rate exceeds this range, consistent with

API-mediated retrieval providing higher factual precision than embedding-based retrieval over serialised structured text [7].

Deepening the multi-tool comparison. The three automated tools deliberately probe different facets of system quality, which explains the spread between the 95.00%, 90.00%, and 85.00% pass rates. OpenAI Evals applies rule-based pass/fail comparisons against curated expected outputs and is comparatively forgiving when the model's wording differs from the reference, provided the key facts are present—this tolerance favours higher pass rates and biases the metric toward factual coverage. Ragas, by contrast, decomposes quality into faithfulness, answer relevancy, context precision, and context recall, and penalises answers where the retrieved context only partially covers the information used in the final answer; the two Ragas failures in this study are both context-recall related rather than factual errors. DeepEval's G-Eval-style LLM-as-a-judge enforces a more demanding semantic completeness criterion and is the most sensitive to omissions or under-explanation; its three failures (one moderate, two complex) accordingly highlight cases where the system answered correctly but tersely. Taken together, the three tools therefore expose three distinct failure modes—rule-based factual gaps, retrieval-recall gaps, and semantic completeness gaps—that no single tool would have surfaced individually. This complementarity is the practical argument for treating multi-tool evaluation as a default for RAG systems rather than as redundant tooling [36].

5.5. Expert Evaluation Results

Table 5 presents the expert evaluation results across five assessment dimensions ($n = 5$ experts, 5-point Likert scale). The overall mean of 3.82 (SD = 1.36; Median = 4.00; IQR = 2.00; high level) indicated that experts regarded the prototype positively. Answer correctness and reliability received the highest mean (3.92), consistent with strong automated correctness results. Feasibility of practical implementation scored second highest (3.88), indicating experts considered the system viable for further development. The RAGAS-dimension quality score (3.85) reflects expert judgement that the system's retrieval-grounding approach is sound. Appropriateness of system operation and usefulness for business operations both scored 3.72, suggesting room for improvement in handling complex queries.

Qualitative expert feedback highlighted three recurring themes: (1) the prototype's greatest strength was its ability to connect ERP data and present it through natural-language dialogue, making business data accessible without requiring ERP menu knowledge; (2) experts recommended adding data validation mechanisms, source attribution for answers, and role-based access control before production deployment; and (3) several experts recommended extending the API scope to cover Accounting and CRM modules as high-priority additions.

5.6. End-User Evaluation Results

Table 6 presents the end-user satisfaction evaluation results ($n = 3$; 5-point Likert scale). The overall mean of 4.33 (SD = 0.77; Median = 5.00; IQR = 1.00; highest level) indicated strongly positive user reception. No participant rated any dimension at levels 1 or 2. Convenience and clarity (D1) and appropriateness and overall satisfaction (D3) received the highest means (both 4.44), reflecting that users found the natural-language interface format intuitive and suitable for their daily data-access tasks.

Table 6. End-user satisfaction evaluation results ($n = 3$; 5-point Likert scale).

Evaluation Dimension	Mean	SD	Median	IQR	Level
D1: Convenience and clarity	4.44	0.53	4.00	1.00	Highest
D2: Benefits for work	4.22	0.83	4.00	1.50	Highest
D3: Appropriateness and overall satisfaction	4.44	0.73	5.00	1.00	Highest
D4: Intention to continue using the system	4.33	1.03	5.00	2.00	Highest
D5: Quality of answers and data	4.25	0.87	4.50	1.50	Highest
Overall	4.33	0.77	5.00	1.00	Highest

Note: Score distribution: Level 5 scored 23 times, Level 4 scored 14 times, Level 3 scored 8 times; no Level 1 or 2 responses. The higher SD for D4 (intention to continue, 1.03) and IQR of 2.00 reflects variation in adoption intent linked to participants' primary task focus: staff whose work centres on daily inventory and sales queries expressed stronger adoption intent than those performing less frequent, more complex financial queries.

5.7. System Usability Scale Results

The SUS evaluation ($n = 3$) produced a score of 66.67 out of 100. Recent SUS benchmarking work commonly treats a mean score near 68 as an approximate usability benchmark, indicating that the system is acceptable as a prototype but has identifiable usability gaps requiring attention before production deployment [37,38]. Item-level analysis revealed that confidence in system use (item 9, mean raw score = 2.33) and the need for expert assistance (item 4, mean raw score = 2.67) were the weakest dimensions, while system consistency (item 6, mean raw score = 1.33 on the negative item, corresponding to a favourable 3.67 after reversal) and perceived ease of use (item 3, mean raw score = 4.00) were the strongest. The SUS score provides important complementary information not captured by the Likert satisfaction scores: users were satisfied overall but less confident in their independent command of the system.

6. Discussion

6.1. Principal Findings

The central finding of this study is that a RAG-based ERP chatbot, developed following R&D and case-study principles and evaluated using a four-dimensional assessment protocol, can demonstrably reduce ERP data-access complexity for SME users while achieving strong technical performance on structured data retrieval tasks. Three automated evaluation tools converged on the conclusion that the system performs well (85–95% pass rates), and this technical performance was validated and contextualised by expert evaluation (mean 3.82) and end-user satisfaction (mean 4.33) using real organisational participants. The SUS score of 66.67 provides important nuance: users were satisfied with the chatbot experience but identified confidence and stability gaps that must be addressed before production deployment.

6.2. Theoretical Implications

The findings extend contemporary ERP success research based on the DeLone and McLean IS Success Model in an important direction: they demonstrate that an AI intermediary layer can directly improve the information-quality and system-quality dimensions of ERP success—accessibility and timeliness—without requiring changes to the underlying ERP database or business logic [10,11]. Consistent with the model's predictions, improvements in perceived information quality translated into higher use intention and user satisfaction in the end-user evaluation.

The divergence between expert (mean 3.82) and end-user (mean 4.33) evaluations is theoretically informative from a Task-Technology Fit perspective [12]. Experts, assessing against technical and professional standards, identified limitations in complex query handling and output consistency. End users, whose primary tasks involve routine data

retrieval, experienced a stronger fit between their task requirements and the chatbot's capabilities. This divergence suggests that TTF assessments for enterprise AI systems should distinguish between expert-evaluated fitness and user-experienced fitness, as these may not coincide at the prototype stage.

The SUS score (66.67) reinforces this interpretation: the score indicates that while the system meets a minimum usability threshold, user confidence in independent operation remains limited—consistent with Adam et al.'s [13] finding that user adoption of AI systems depends not only on initial satisfaction but on sustained confidence in the system's reliability.

Reconciling the SUS–satisfaction divergence. A central interpretive challenge raised by the reviewers is the apparent tension between strongly positive overall satisfaction (mean 4.33/5.00) and a marginal SUS score (66.67/100, on the boundary between 'OK' and 'Good'). Item-level inspection sharpens this picture: users were enthusiastic about the concept and convenience of the conversational interface, yet rated their independent confidence in system use low (item 9 raw mean = 2.33) and expressed a comparatively high perceived need for expert assistance (item 4 raw mean = 2.67). A plausible reading is that the satisfaction score reflects a 'novelty effect': participants encountered the natural-language interface for the first time, found the absence of menu navigation liberating, and responded affectively to a dramatic reduction in surface-level task friction. The SUS items, however, probe a different construct—sustained operational confidence and perceived independence—and these probes revealed underlying anxieties about data correctness, system stability under load, and what the user should do when the chatbot's answer differs from the user's prior expectation. This pattern is consistent with prior chatbot-adoption findings that novelty-driven enthusiasm fades rapidly when confidence in factual correctness is not actively reinforced [13,14]. The implication for the next development cycle is concrete: usability gains must be accompanied by explicit confidence-building mechanisms (source attribution, traceable links back to ERP records, lightweight uncertainty signals, and graceful failure messaging), not by interface polish alone.

6.3. Practical Implications for SME Digital Transformation

For SME managers and IS practitioners, this study offers three concrete guidance points. First, RAG-based ERP chatbots are deployable at SME scale using Odoo's standard APIs and cloud-hosted LLMs—no sophisticated ML infrastructure is required. Second, the multi-tool automated evaluation framework (OpenAI Evals + DeepEval + Ragas) provides a practical quality assurance template: organisations can apply these tools to monitor factual correctness, semantic quality, and retrieval quality continuously. Third, the transparent domain-scoping mechanism (appropriate refusal of out-of-scope questions) was well-received by both experts and users, suggesting that clear boundary communication is a prerequisite for trust—a finding consistent with Følstad and Brandtzæg's [15] work on chatbot adoption.

6.4. Methodological Implications

The multi-tool evaluation design reveals important complementarity: three tools operating on the same test set produced meaningfully different pass rates (85%, 90%, 95%). OpenAI Evals' higher rate reflects tolerance for semantically correct answers with varied phrasing; DeepEval's lower rate reflects a more demanding semantic completeness criterion; Ragas' intermediate rate is sensitive to retrieval-specific failures (incomplete context recall). The SUS score adds a further orthogonal dimension: users were satisfied (4.33/5.00), but their confidence in independent system use was limited (SUS = 66.67). These four evaluation dimensions, automated correctness, semantic quality, retrieval quality, and user confidence,

are genuinely non-redundant. Any single metric would have provided an incomplete and potentially misleading quality picture, supporting the recommendation of Gan et al. [36] that RAG evaluation should routinely combine multiple complementary tools.

6.5. Comparison with Prior Work

Consistent with recent RAG surveys [4,5], the study confirms that retrieval grounding substantially reduces hallucination risk when the model is required to answer from retrieved evidence—all 20 test-case failures were attributable to retrieval limitations rather than model hallucination. The 95.00% OpenAI Evals correctness rate exceeds the 78–91% range reported by Chen et al. [20] for structured data RAG, consistent with API-mediated retrieval providing higher factual precision. The end-user mean of 4.33 compares favourably with comparable RAG chatbot studies [16], likely reflecting the trust advantage of retrieval-grounded accuracy. The SUS score of 66.67 sits close to common SUS benchmark values used for interpreting prototype usability [37,38], appropriate for a functional prototype. This study directly addresses the gap identified by Zhang et al. [30], providing the first fully documented R&D cycle for an RAG-based ERP chatbot evaluated with real SME data.

7. Limitations and Future Work

This study has several limitations that should be considered when interpreting findings.

Small evaluation sample: Expert evaluation used $n = 5$, and end-user evaluation used $n = 3$. While these sample sizes are justified for formative usability evaluation and R&D case-study research [55–57], statistical generalisation from these data is not warranted. The SUS score in particular should be treated as an indicative estimate rather than a statistically robust performance measure until replicated with larger samples. Future research should conduct a confirmatory evaluation with a larger user sample—ideally $n \geq 15$ to support meaningful descriptive comparison—and test across multiple organisations.

External validity: The study uses a single case-study organisation in one industry sector (distribution) and one ERP platform (Odoo 18 Community). Findings cannot be automatically generalised to SMEs in other sectors, with different ERP platforms (SAP Business One, ERPNext, Microsoft Dynamics), or in other cultural contexts. Future research should replicate the study across multiple organisations and ERP platforms.

Transferability conditions. Although the prototype’s logical pattern—natural-language front end, FastAPI orchestration, structured-API retrieval layer, LLM-grounded response—is platform-neutral in principle, several practical conditions determine whether the design transfers cleanly to other ERPs. Specifically, transfer is most straightforward for ERP systems that (i) expose documented external APIs at module level (e.g., SAP Business One Service Layer, ERPNext REST API, Microsoft Dynamics OData), (ii) provide consistent authentication and rate-limit guarantees suitable for synchronous LLM tool calls, and (iii) support row-level or field-level permission scoping that can be mapped onto the LLM-tool layer. ERPs without one or more of these properties (e.g., legacy in-house systems or some heavily customised on-premise SAP deployments) will require additional adapter engineering, and the prototype’s evaluation pattern would have to be re-validated against each new platform’s API semantics. A planned multi-platform follow-up study is identified in the future-work agenda below to test these conditions empirically.

API coverage: The prototype connects to five Odoo modules. Questions involving Accounting, CRM, HR, Project Management, or Manufacturing cannot be answered. Expert evaluators specifically recommended extending the API scope to Accounting and CRM as high-priority additions. Future development should expand API integration to these modules.

LLM dependency and data privacy: The prototype relies on an external cloud-hosted LLM for both intent classification and answer generation. This creates vendor dependency and raises data privacy concerns when real business data are passed to an external API. Future work should evaluate locally deployable open-source LLMs (e.g., gemini-3.1-flash-lite-preview, gemini-3.5-flash) as privacy-preserving alternatives and benchmark their performance relative to the cloud model used in this study.

Longitudinal evaluation: End-user evaluation was a single session. Longitudinal evaluation tracking daily usage would provide richer evidence of adoption persistence, emergent use cases, and habituation effects on the SUS score.

Cost and latency analysis: The study did not systematically measure API call costs or response latency under production-scale loads. Future work should provide cost-per-query and latency benchmarks to inform SME deployment decisions. System stability improvements—identified as a concern by end users—should also be addressed through fault tolerance and connection management enhancements.

8. Conclusions

This study designed, developed, and evaluated a Retrieval-Augmented Generation ERP chatbot for supporting natural-language access to SME business data, using a real Odoo ERP deployment in Khon Kaen Province, Thailand, as the empirical context. The research was conducted within an R&D + case-study framework, ensuring systematic documentation of both the design artefact and its multidimensional evaluation.

The study's four principal contributions are: (1) a replicable R&D-grounded architecture for ERP-integrated RAG chatbots, specifying design decisions from API endpoint selection through agentic tool-calling and prompt structure; (2) formal as-is and to-be process models documenting the transition from a five-step menu-driven ERP access process to a two-step natural-language conversation; (3) a four-level test-question taxonomy (easy, moderate, complex, out-of-scope) validated as a practical evaluation tool for conversational ERP systems; and (4) an empirical demonstration of the diagnostic complementarity of three automated RAG evaluation tools (OpenAI Evals, DeepEval, Ragas) combined with Likert-scale expert assessment and SUS-based end-user evaluation.

The prototype achieved automated pass rates of 95.00% (OpenAI Evals), 90.00% (Ragas), and 85.00% (DeepEval). Expert evaluation produced an overall mean of 3.82 (high level; $n = 5$), end-user satisfaction produced an overall mean of 4.33 (highest level; $n = 3$), and the SUS score was 66.67/100 (acceptable prototype-level usability, with identified gaps in user confidence and system stability). These results collectively confirm that RAG-based ERP chatbots are technically feasible, practically acceptable, and sufficiently useful for SME deployment—while highlighting clear development priorities (larger API coverage, UX/UI improvement, role-based access control, and stability enhancements) required before production release.

As ERP systems continue to centralise ever-larger volumes of SME operational data, the design of natural-language access interfaces grounded in reliable retrieval represents an increasingly important frontier for IS research and practice. This study provides an empirical foundation and a replicable evaluation framework to support that agenda.

Author Contributions: Conceptualisation, P.P. and W.C.; methodology, P.P. and W.C.; software, P.P. and W.C.; validation, P.P. and W.C.; formal analysis, P.P. and W.C.; investigation, P.P. and W.C.; resources, P.P.; data curation, P.P.; writing—original draft preparation, P.P. and W.C.; writing—review and editing, P.P. and W.C.; visualisation, P.P. and W.C.; supervision, W.C.; project administration, W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request, subject to participant privacy agreements.

Acknowledgments: During the preparation of this work, the authors utilised ChatGPT (OpenAI, San Francisco, CA, USA; GPT-5.3) and Grammarly (Grammarly Inc., San Francisco, CA, USA; Premium version, web-based application) to enhance their English proficiency. All authors have reviewed and edited the manuscript after using these tools and take full responsibility for the final content. No individuals are acknowledged in this section without their prior consent.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Gartner. *ERP Market Forecast and User Adoption Study*; Gartner Research: Stamford, CT, USA, 2023.
2. Fathoni, M.Z.; Asih, A.M.S.; Wibisono, M.A. Adoption of open-source enterprise resource planning in small and medium industries: A literature review. In *2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*; IEEE: New York, NY, USA, 2024; pp. 272–276. [CrossRef]
3. Setiawan, D.; Fahrezha, M.; Prakoso, N.A.B.; Qurtubi, Q. A proposed framework for ERP system implementation in SMEs. *Int. J. Artif. Intell. Res.* **2023**, *7*, 181. [CrossRef]
4. Sharma, C. Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers. *arXiv* **2025**. [CrossRef]
5. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv* **2023**. [CrossRef]
6. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria, 7–11 May 2024; pp. 1–30.
7. Wang, Y.; Ma, X.; Wu, W. Retrieval-augmented generation over tabular data: Challenges and opportunities. *Find. Assoc. Comput. Linguist. ACL* **2024**, *2024*, 5621–5635.
8. Odoo, S.A. External API—Odoo 18.0 Documentation. 2026. Available online: https://www.odoo.com/documentation/18.0/developer/reference/external_api.html (accessed on 25 June 2026).
9. Arvianto, A.; Rosyada, Z.F.; Saptadi, S.; Budiawan, W.; Demilda, Y.E. ERP Odoo implementation in small retailers. *Int. J. Appl. Sci. Eng. Rev.* **2022**, *3*, 66–85. [CrossRef]
10. Barus, N.A.; Muda, I.; Kesuma, S.A. A systematic review of the DeLone & McLean model in enterprise resource planning (ERP) systems success. *J. Mod. Account. Audit.* **2025**, *21*, 90–107. [CrossRef]
11. Jo, H.; Bang, Y. Understanding continuance intention of enterprise resource planning (ERP): TOE, TAM, and IS success model. *Heliyon* **2023**, *9*, e21019. [CrossRef] [PubMed]
12. Sonntag, M.; Mehmman, J.; Teuteberg, F. AI-Based Chatbots in Customer Service: A Task-Technology Fit (TTF) Model. *Int. J. Serv. Sci. Manag. Eng. Technol.* **2025**, *16*, 1–20. [CrossRef]
13. Adam, M.; Wessel, M.; Benlian, A. AI-based chatbots in customer service and their effects on user compliance. *Electron. Mark.* **2021**, *31*, 427–445. [CrossRef]
14. Ding, Y.; Najaf, M. Interactivity, humanness, and trust: A psychological approach to AI chatbot adoption in e-commerce. *BMC Psychol.* **2024**, *12*, 595. [CrossRef] [PubMed]
15. Følstad, A.; Brandtzæg, P.B. Users’ experiences with chatbots: Findings from a questionnaire study. *Qual. User Exp.* **2020**, *5*, 3. [CrossRef]
16. Ravi, D.; Sindhgatta, R. Exploring trust and transparency in retrieval-augmented generation for Domain Experts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*; ACM: New York, NY, USA, 2025. [CrossRef]
17. Izacard, G.; Grave, E. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 874–880. [CrossRef]
18. Ram, O.; Levine, Y.; Dalmedigos, I.; Muhlgay, D.; Shashua, A.; Leyton-Brown, K.; Shoham, Y. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguist.* **2023**, *11*, 1316–1331. [CrossRef]
19. Liang, J.; Sugang, Lin, H.; Wu, Y.; Zhao, R.; Li, Z. Reasoning RAG via System 1 or System 2: A survey on reasoning agentic retrieval-augmented generation for industry challenges. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 1954–1966.

20. Chen, J.; Lin, H.; Han, X.; Sun, L. Benchmarking large language models in retrieval-augmented generation. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 17754–17762. [\[CrossRef\]](#)
21. Singh, A.; Ehtesham, A.; Kumar, S.; Khoei, T.T.; Vasilakos, A.V. Agentic retrieval-augmented generation: A survey on agentic RAG (Version 4). *arXiv* **2026**. [\[CrossRef\]](#)
22. Zhao, P.; Zhang, H.; Yu, Q.; Wang, Z.; Geng, Y.; Fu, F.; Yang, L.; Zhang, W.; Jiang, J.; Cui, B. Retrieval-augmented generation for AI-generated content: A survey (Version 4). *arXiv* **2026**. [\[CrossRef\]](#)
23. Es, S.; James, J.; Espinosa-Anke, L.; Schockaert, S. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*; Aletras, N., De Clercq, O., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 150–158. [\[CrossRef\]](#)
24. Saad-Falcon, J.; Khattab, O.; Potts, C.; Zaharia, M. ARES: An automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 338–354. [\[CrossRef\]](#)
25. Confident AI. G-Eval. DeepEval Documentation. 2026. Available online: <https://deepeval.com/docs/metrics-llm-evals> (accessed on 25 June 2026).
26. Wu, J.Y.; Chen, L.T. Odoo ERP with business intelligence tool for a small-medium enterprise. In *Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning*; ACM: New York, NY, USA, 2020; pp. 323–327. [\[CrossRef\]](#)
27. Chanphet, P.; Tianpasakorn, K.; Wuttipanyarattanakul, S. Critical success factors of ERP implementation: Literature review. *J. Manag. Sci. Ubon Ratchathani Univ.* **2024**, *13*, 94–112.
28. Sudarmo, S.; Rusdiana, A.; Munir, M. The effect of enterprise resource planning (ERP) system implementation, user training, and management support on user satisfaction in manufacturing companies. *West Sci. Inf. Syst. Technol.* **2024**, *2*, 233–243. [\[CrossRef\]](#)
29. Rutz, P.; Stevens, G.; Wulf, V. Supporting the appropriation of ERP systems in SMEs: A practice-centred approach. In *Proceedings of the 22nd European Conference on Computer-Supported Cooperative Work*; European Society for Socially Embedded Technologies: Bonn, Germany, 2024.
30. Zhang, L.; Wang, S.; Liu, B. Artificial intelligence integration in ERP systems: A review and research agenda for Industry 4.0. *Comput. Ind.* **2024**, *155*, 104052.
31. Adamopoulou, E.; Moussiades, L. An overview of chatbot technology. In *Artificial Intelligence Applications and Innovations*; Maglogiannis, I., Iliadis, L., Pimenidis, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; pp. 373–383. [\[CrossRef\]](#)
32. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Zoph, B. GPT-4 technical report. *arXiv* **2023**. [\[CrossRef\]](#)
33. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Amodei, D. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
34. Shi, L.; Tang, Z.; Zhang, N.; Zhang, X.; Yang, Z. A survey on employing large language models for text-to-SQL tasks. *arXiv* **2024**. [\[CrossRef\]](#)
35. Rajkumar, N.; Li, R.; Bahdanau, D. Evaluating the text-to-SQL capabilities of large language models. *arXiv* **2022**. [\[CrossRef\]](#)
36. Gan, A.; Yu, H.; Zhang, K.; Liu, Q.; Yan, W.; Huang, Z.; Tong, S.; Hu, G. Retrieval augmented generation evaluation in the era of large language models: A comprehensive survey. *arXiv* **2025**. [\[CrossRef\]](#)
37. Lewis, J.R.; Sauro, J. Item benchmarks for the System Usability Scale. *J. Usability Stud.* **2018**, *13*, 158–167.
38. Hyzy, M.; Bond, R.; Mulvenna, M.; Bai, L.; Dix, A.; Leigh, S.; Hunt, S. System Usability Scale benchmarking for digital health apps: Meta-analysis. *JMIR mHealth uHealth* **2022**, *10*, e37290. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Koo, M. Likert-type scale. *Encyclopedia* **2025**, *5*, 18. [\[CrossRef\]](#)
40. Kroop, S. Artifact validity in design science research (DSR): A comparative analysis of three influential frameworks. *arXiv* **2025**. [\[CrossRef\]](#)
41. Annamalah, S. Exploring the relevance and rigour of case study research in business and management. *J. Sustain. Res.* **2025**, *7*, e250004.
42. Käss, S.; Schermann, M.; Krcmar, H. Short and sweet: Multiple mini case studies as a form of rigorous case study research in information systems. *Inf. Syst. E-Bus. Manag.* **2024**, *22*, 351–384. [\[CrossRef\]](#)
43. Yas, Q.M.; Ali, Z.H.; Hussein, M.K. A comprehensive review of software development life cycle methodologies: Traditional and agile approaches. *Int. J. Comput. Sci. Mob. Comput.* **2023**, *4*, 14.
44. Sanmocte, E.M.T.; Costales, J.A. Exploring effectiveness in software development: A comparative review of system analysis and design methodologies. *Int. J. Comput. Theory Eng.* **2025**, *17*, 36–43. [\[CrossRef\]](#)
45. Schwaber, K.; Sutherland, J. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. November 2020. Available online: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf> (accessed on 2 March 2026).
46. Chainlit. Overview. Chainlit Documentation. 2026. Available online: <https://docs.chainlit.io/get-started/overview> (accessed on 25 June 2026).

47. Chainlit. User Session. Chainlit Documentation. 2026. Available online: <https://docs.chainlit.io/concepts/user-session> (accessed on 25 June 2026).
48. FastAPI. (n.d.). FastAPI. Available online: <https://fastapi.tiangolo.com> (accessed on 10 April 2026).
49. Paulsson, V.; Johansson, B. Cloud ERP systems architectural challenges on cloud adoption in large international organizations: A sociomaterial perspective. *Procedia Comput. Sci.* **2023**, *219*, 797–806. [CrossRef]
50. Hevner, A.R.; Parsons, J.; Brendel, A.B.; Lukyanenko, R.; Tiefenbeck, V.; Tremblay, M.C.; Vom Brocke, J. Transparency in design science research. *Decis. Support Syst.* **2024**, *182*, 114236. [CrossRef]
51. Perez, E.; Ringer, S.; Lukošūtė, K.; Nguyen, K.; Chen, E.; Heiner, S.; Kaplan, J. Discovering language model behaviors with model-written evaluations. *Find. Assoc. Comput. Linguist. ACL* **2023**, *2023*, 13387–13434. [CrossRef]
52. OpenAI. Working with Evals. OpenAI API Documentation. 2026. Available online: <https://developers.openai.com/api/docs/guides/evals/> (accessed on 25 June 2026).
53. Confident AI. Introduction to LLM metrics. DeepEval Documentation. 2026. Available online: <https://deepeval.com/docs/metrics-introduction> (accessed on 25 June 2026).
54. Ragas. List of Available Metrics. Ragas Documentation. 2026. Available online: https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/ (accessed on 25 June 2026).
55. Nielsen, J. Why You Only Need to Test with 5 Users. Nielsen Norman Group. 19 March 2000. Available online: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (accessed on 12 May 2026).
56. Clark, N.; Dabkowski, M.; Driscoll, P.; Kennedy, D.; Kloo, I.; Shi, H. Empirical decision rules for improving the uncertainty reporting of small sample System Usability Scale scores. *arXiv* **2021**. [CrossRef]
57. Al Qur'an, M.N. Conducting Case Study Research in International Entrepreneurship: A Protocol for Qualitative Case Study. *Int. J. Qual. Methods* **2025**, *24*, 1–10. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.