AIGC-Assisted Data Center Inspection Management System

Baowen Liu[1], Xiuping Xia[2], Ruixi Chen[3], Bangying Wu[4]

[1]Network and Information Center, Southwest Petroleum University, Chengdu, China

[2]School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu, China

[3]School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu, China

[4]Network and Information Center, Southwest Petroleum University, Chengdu, China

Abstract: ***Existing data center inspection systems have limitations in information accuracy, anomaly traceability, and the retention of operational experience. This paper presents the design and implementation of an AIGC-assisted data center inspection management system integrating LLM-based question answering with a structured knowledge base. The system adopts a layered architecture and supports field inspectors and operations administrators through equipment-room inspection, knowledge-base management, duty scheduling, data statistics, and intelligent question answering. Retrieval-augmented generation is used to retrieve relevant operational knowledge and support traceable assistance while retaining human control over operational decisions. The system has been deployed in a campus data center and demonstrates the practical feasibility of integrating standardized inspection workflows, anomaly handling, and reusable operational knowledge within a unified management platform.***

Keywords: data center; intelligent inspection; retrieval-augmented generation; knowledge base; AIGC

1. Introduction

As university informatization advances, the equipment and business systems hosted by campus data centers continue to expand in scale, and their operational reliability directly bears on the continuity of core operations such as teaching and research. Confronted with this rapid infrastructure growth, traditional, predominantly manual inspection falls short across coverage, timeliness, and consistency, and cannot keep pace with the mounting operations and maintenance burden.

Existing research on intelligent inspection has largely focused on equipment condition monitoring and automated data collection, while paying limited attention to the human-centric inspection workflow and its knowledge retention; frontline operational experience consequently remains scattered across unstructured documents and resists systematic reuse.

This paper presents the design and implementation of an AIGC-assisted inspection management system. The main contributions of this work are summarized as follows:

1) An integrated inspection model that incorporates server-room hardware inspection, application-system inspection, duty scheduling, and knowledge-base management into a unified operational framework, enabling collaborative workflows between field inspectors and operations administrators.
2) An assisted question-answering mechanism based on retrieval-augmented generation (RAG) that leverages the knowledge base to constrain LLM-generated AIGC outputs and mitigate hallucination, while retaining human authority over decision-making.
3) The implementation of the proposed system and its deployment in a real campus data center, demonstrating its practical feasibility in production environments.

This paper is organized as follows. Section 2 reviews related work. Section 3 presents the overall system design. Section 4 describes the system implementation and deployment. Section 5 concludes the paper and outlines future work.

2. Literature Survey

2.1 Data Center Inspection Systems

Early work on data center inspection focused on watching the physical environment in real time- power, temperature, humidity, and airflow. Levy and Hallstrom built non-intrusive DCIM monitoring on top of low-power wireless sensor networks [1]. Kahil et al. later proposed an application-aware management framework for next-generation data centers [2], and Matko et al. reduced alarm noise with multi-layer node event processing [3]. Two surveys, by Feng et al. [4] and Balabanov et al.[5], observe that image-based equipment monitoring and mainstream DCIM platforms do work in practice, but mostly stay at facility-level automated monitoring.

From there, inspection moved gradually away from fixed periodic patrols toward smarter methods. He et al. built an augmented-reality (AR) operation and maintenance system for data centers that lets staff log inspections and call remote experts [6]; Zhang and Jia, for their part, built an IDC server-room management and control system from data mining and ran it in production at China Unicom [7]. On the industrial side, Królczyk [8] and Mohd Noor et al. [9] used computerized maintenance management systems (CMMS) to show why it helps to bring work orders, schedules, and documentation together and link them with predictive maintenance. Wang et al. proposed KGroot, which builds a knowledge graph from past faults to reason about root causes and turns scattered operational experience into knowledge that can actually be queried [10].

2.2 LLMs in Operations and Maintenance

LLMs have pushed AIOps past its earlier reliance on predicting from metrics and logs, and toward systems that work with meaning and generate reasoning. Zhang et al.'s survey of LLM-based AIOps [12] finds that LLMs already help with reading anomalies, tracing root causes, and suggesting fixes; Ahmed et al. studied more than 40,000 cloud incident tickets from Microsoft [13] and found that LLMs can propose plausible root-cause guesses and concrete mitigation steps both with no training and with fine-tuning, though how good the answer is depends heavily on whether enough context is given.

RAG, introduced by Lewis et al. [11], pulls evidence from an external knowledge base before answering, which makes knowledge-heavy tasks more trustworthy and easier to trace. Gao et al. [14], however, note that retrieval quality, query rewriting, and context compression decide how faithful the answer really is, and that plain RAG still struggles to link across documents or reason over several hops.

Taken together, most LLM and multi-agent operations research aims at full automation in large cloud-native settings. Smaller and medium-sized data centers, by contrast, still lack a cheap and controllable way to keep expert experience on hand and use it to keep LLM outputs in line—and that gap is what led us to bring RAG and a structured inspection framework into this work.

3. System Design

3.1 Overall Architecture Design

The system uses a B/S (Browser/Server) architecture and a conventional layered structure, split from top to bottom into four tiers: the user layer, the presentation layer, the application layer, and the data layer, as shown in Figure 1.

The user layer covers the system's two kinds of end users. Field inspectors carry out on-site inspections and handle anomalies, while operations administrators take care of back-end setup: configuring inspection items, scheduling duties, producing statistics, and maintaining the knowledge base.

For the presentation layer, the browser loads static pages as the entry point and offers two interfaces. The inspection front end is meant for on-site work and brings together task overview, status monitoring, and function navigation; the management back end supports O&M control and holds the visual editors for inspection items, scheduling, statistics, and the knowledge base.

The application layer runs on Spring Boot. It is divided by business function into several modules: user and permission management, inspection management, data statistics, knowledge-base management, duty scheduling, and the AI assistant. These modules expose their functions to outside clients through REST APIs.

The data layer keeps all business data in a MySQL database, covering the main entities (users, inspection objects, inspection records, and the knowledge base). Redis stores session state and caches the data that gets requested often, which takes pressure off the database and makes list queries and statistical aggregation noticeably faster.

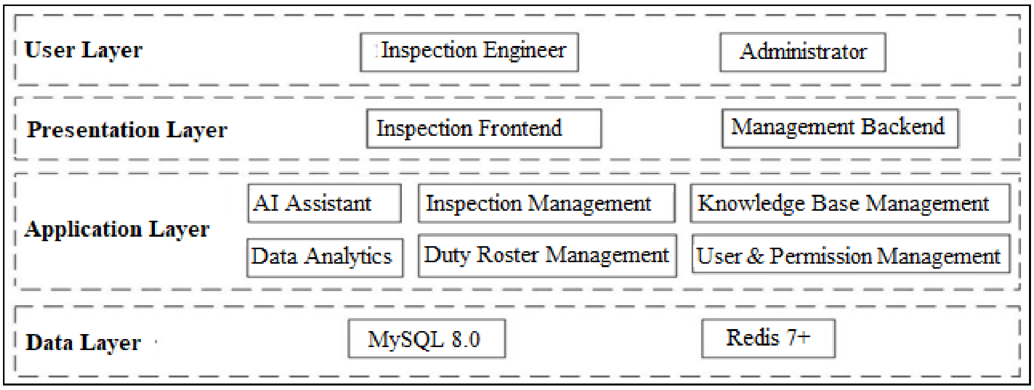

Figure 1: Overall system architecture

3.2 Overall Functional Design

As shown in Figure 2, the system comprises five functional modules: personnel management, server-room inspection, knowledge-base Q&A, data statistics, and AIGC-assisted Q&A. What characterizes the design is a combination of fine-grained personnel governance, standardized and digitalized operations, experience that is kept and reused, and multi-dimensional statistics supported by intelligent assistance.

Paper ID: SR26917220843 DOI: https://dx.doi.org/10.21275/SR26917220843

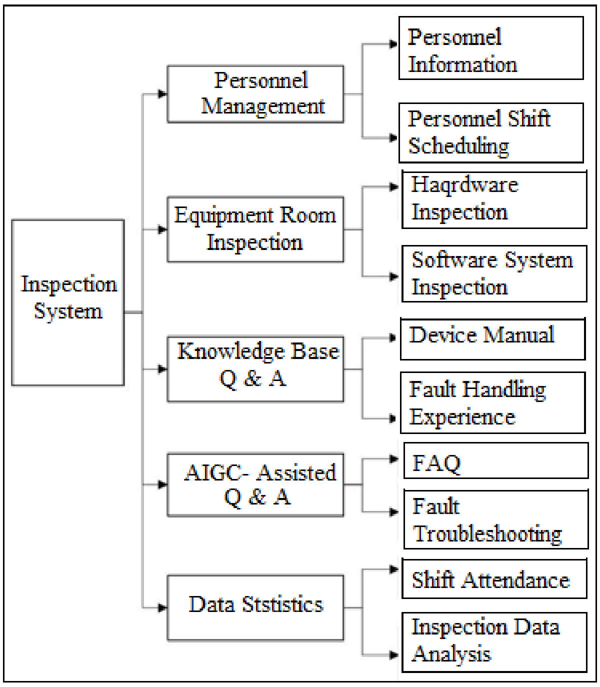

Figure 2: Overall functional diagram

3.3 Core Functional Design

3.3.1 Equipment Room Inspection Module Design

The Equipment Room Inspection is the operational core of the system. It handles two kinds of inspection objects, server-room hardware and application systems, and stores every inspection record. Hardware objects cover power and environment metrics such as server-room temperature and humidity, plus the core devices: servers, storage, UPS, and switches. Application-system objects range over storage clusters, database clusters, virtualization management platforms, and security audit platforms.

When an inspector picks an object, its detail page opens with the object's attributes, inspection guidelines, and past records. After finishing the on-site check, the inspector fills in the status, remarks, and a rich-text description, attaches photos taken on site as evidence, and submits. If something is wrong, the inspector marks the fault and performs immediate remediation, sending the remediation steps together with the entry; the system then ties the record to both the object and the inspector and saves it to the database.

Each inspection record holds the same set of fields: the inspection object, the inspector, status, time, remarks, a detailed description, and attachments. When an anomaly shows up, it also keeps the fault symptoms and the remediation measures. Operations administrators can filter records by server room, status, and time span, which helps trace anomalies and tally completion; the matching remediation plans are then stored in the knowledge base so they can be reused when the same problem comes back, as shown in Figure 3.

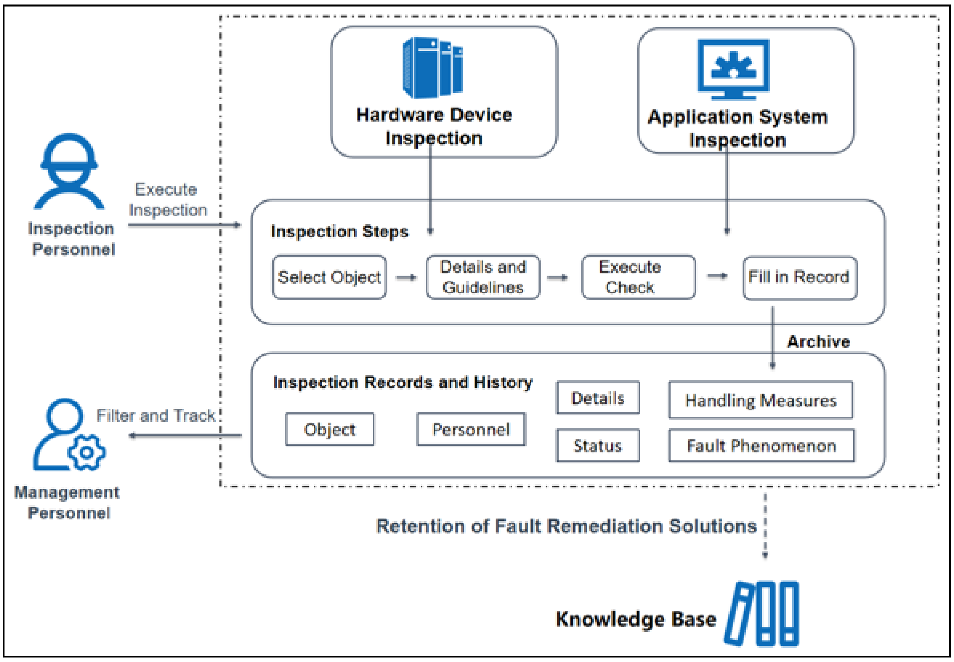

Figure 3: Personnel inspection workflow

3.3.2 AI Assistant and Knowledge-Base-Assisted Design

The AI assistant pairs the knowledge base with a large language model (LLM) to give both kinds of users interactive, natural-language help with their work, which shortens the on-site remediation and decision chain. When an inspector hits a device anomaly or needs to look up past faults and how they were handled, they can just ask the assistant in plain language.

To keep the model's answers in check and limit hallucination, the system draws on the knowledge base and improves how traceable its suggestions are. Through retrieval-augmented generation (RAG), it first pulls the relevant passages from the knowledge base, builds them into a prompt, and passes them to the LLM to produce device information and remediation steps, so faults can be located and resolved quickly. Operations administrators can similarly ask the assistant for statistics on inspection completion, anomaly distribution, and personnel workload; the system then reads the aggregated results from the operational database and returns them, as shown in Figure 4.

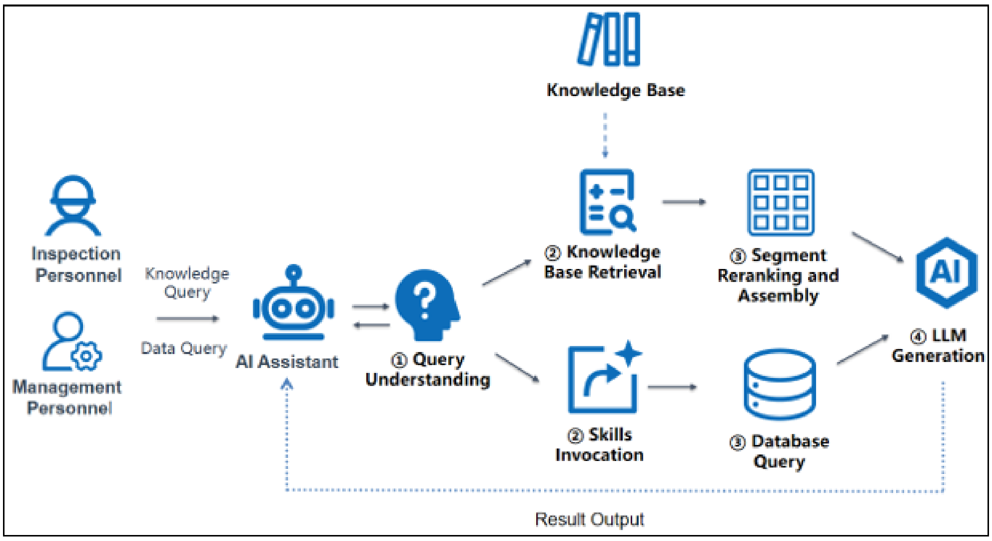

Figure 4: AI assistant and knowledge-base-assisted design workflow

3.4 Core Data Model Design

The system uses a relational data model, organizing its data around five core entities: users, inspection objects, inspection records, the knowledge base, and duty scheduling. Its main data tables are the user table, the inspection-object table, the inspection record table (inspection_record), the knowledge-base table (knowledge), and the duty-scheduling table (shift_schedule). The inspection record table and the knowledge-base table, the two core business tables, are then used as examples to show how their schemas are defined, as shown in Tables 1 and 2.

Table 1: Inspection record table

Data Item	Field	Type	Length	Not Null	Primary Key
Record ID	id	BIGINT	20	Yes	Yes
Equipment Room ID	room_id	BIGINT	20	No	No
Equipment Room Name	room_name	VARCHAR	120	No	No
User ID	user_id	BIGINT	20	No	No
User Name	user_name	VARCHAR	64	No	No
Status	status	VARCHAR	32	No	No
Remarks	notes	TEXT	-	No	No
Rich-text Content	rich_content	MEDIUMTEXT	-	No	No
Images	images	MEDIUMTEXT	-	No	No
Inspection Time	inspection_time	DATETIME	19	No	No
Inspector ID	inspector_id	BIGINT	20	No	No
Inspector Name	inspector_name	VARCHAR	64	No	No
Check Time	inspect_time	DATETIME	19	No	No
Creation Time	create_time	DATETIME	19	No	No
Update Time	update_time	DATETIME	19	No	No
Is Deleted	is_deleted	TINYINT	4	No	No

Table 2: Knowledge base table

Data Item	Field	Type	Length	Not Null	Primary Key
Knowledge ID	id	BIGINT	20	Yes	Yes
Title	title	VARCHAR	200	Yes	No
Tags	tags	VARCHAR	255	No	No
Content	content	MEDIUMTEXT	-	No	No
Device Type	device_type	VARCHAR	100	No	No
Creating User ID	create_user_id	BIGINT	20	No	No
Creating User Name	create_user_name	VARCHAR	64	No	No
Type	type	VARCHAR	64	No	No
Attachment Path	attachment_path	VARCHAR	500	No	No
Creator ID	creator_id	BIGINT	20	No	No
Creator Name	creator_name	VARCHAR	64	No	No
Creation Time	create_time	DATETIME	19	No	No
Update Time	update_time	DATETIME	19	No	No
Is Deleted	is_deleted	TINYINT	4	No	No

4. System Implementation

4.1 Development and Deployment Environment

The system's development and deployment environment is listed in Table 3. It uses containerized deployment, with the application and its dependencies packed into Docker images. The system integrates the AI assistant through a configurable external LLM interface, and keeps the model type and API key in the system configuration.

Table 3: Environment configuration

Category	Configuration
Operating system	Windows 10
Processor	Intel(R) Core (TM) i5-10500 CPU
Memory	32 GB
Deployment	Docker Desktop
Front end	HTML/CSS/JS
Backend framework	Spring Boot (JDK 17, Maven 3.8+)

4.2 Key Module Implementation

4.2.1 Equipment Room Inspection Module

The inspection homepage brings together task overview, status monitoring, and function navigation. It counts the number of outstanding inspection items for each equipment room in real time and dynamically flags anomalies, and gives quick access to hardware-device status, the knowledge base, and relevant documents. Inspection personnel click an equipment-room card to open its detail page, where they run inspection tasks following the standardized checklists and operating procedures, with the key metrics and precautions highlighted, as shown in Figure 5.

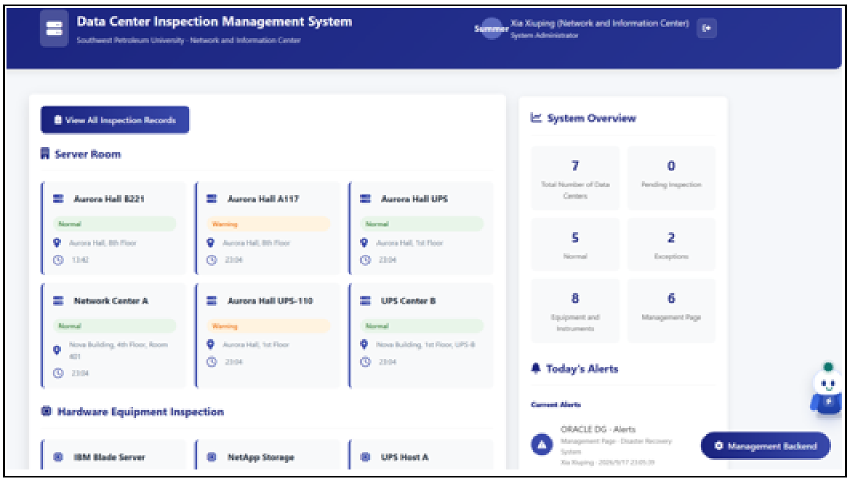

Figure 5: Inspection homepage

On the backend, administrators manage the full lifecycle of inspection items for equipment-room hardware devices and application systems, and can filter all records across multiple dimensions. They assign follow-up handling for anomalous data, closing the loop on anomaly handling, as shown in Figure 6.

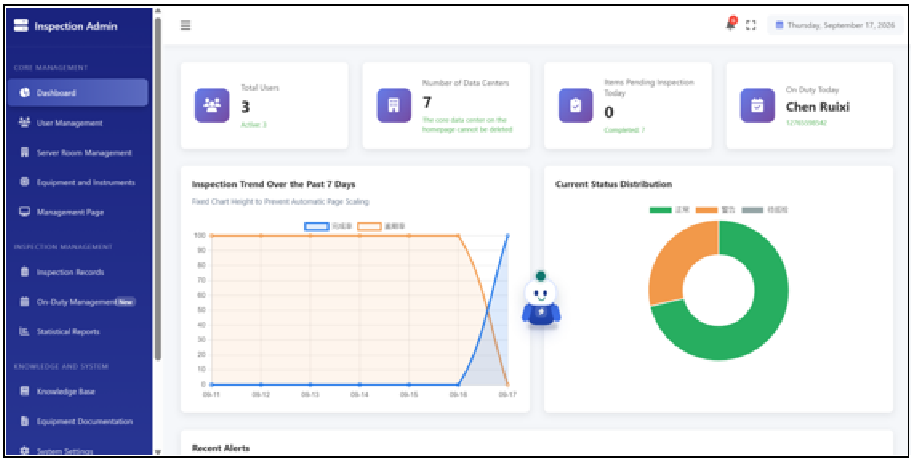

Figure 6: Administrator backend

4.2.2 Knowledge Base and AI Assistant

As shown in Figure 7, the knowledge base and AI assistant module lets the system retain knowledge and provide intelligent support. The AI assistant handles all user-facing assistance. When users run into a device anomaly or an operational question, they type it into a chat interface; the assistant then identifies the user's request, pulls from the fault knowledge base or the device documentation, and returns targeted remediation plans along with the relevant technical documents. O&M administrators can similarly ask the assistant for operational data such as shift status and inspection completion rates. The system answers these data questions by querying the business database for statistics, which helps with management decisions.

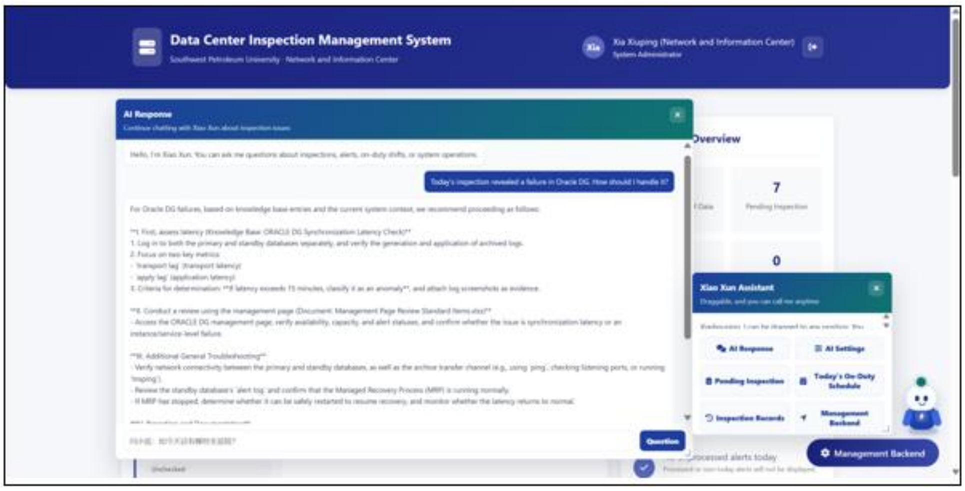

Figure 7: AI assistant

As shown in Figure 8, administrators maintain the knowledge base across its full lifecycle and can manage device documentation and fault articles across dimensions such as tags, categories, body content, and attachments. After maintenance personnel fix a fault, they can feed the resulting solution back into the system, continuously enriching the knowledge base. That way, when a similar issue comes up again, inspection personnel can pull up the matching solution quickly, either through the AI assistant or with a direct search, which speeds up both remediation and maintenance.

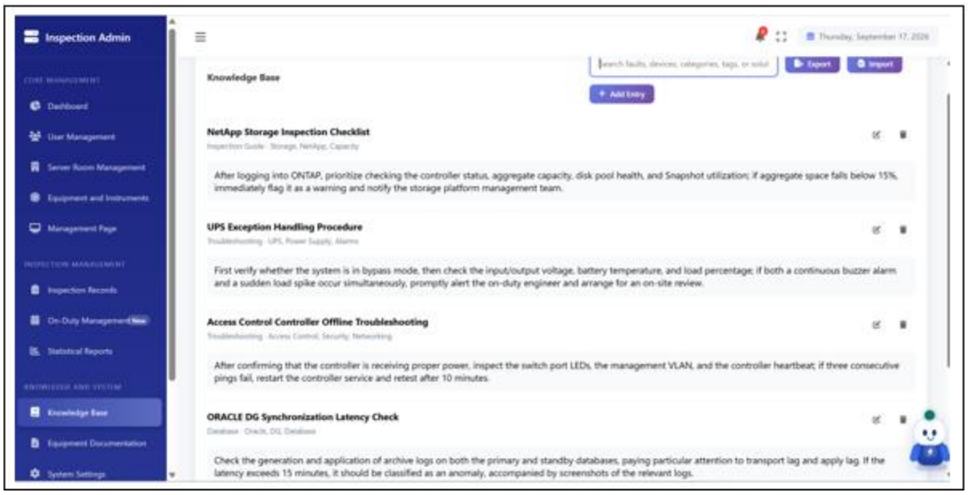

Figure 8: Knowledge base

5. Conclusion

This paper presents and implements an AIGC-assisted inspection management system for university data centers by integrating standardized inspection workflows, knowledge-base retrieval, and intelligent question answering. Deployment in a campus data center demonstrates the practical feasibility of combining inspection records, anomaly handling, operational knowledge retention, and retrieval-augmented assistance within a unified platform. The current AI assistant mainly supports retrieval-augmented question answering, while root-cause localization and remediation decisions continue to require human intervention. Future work will investigate multi-agent collaboration and integration with monitoring and alerting systems to further improve the automation of inspection, diagnosis, and anomaly-handling workflows.

References

[1] Levy M, Hallstrom J O. A new approach to data center infrastructure monitoring and management (DCIMM)[C]// 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas, NV, USA: IEEE, 2017: 1–6.

[2] Kahil H, Saleh T, Välisuo P. Toward a Next-Generation Data Center Management System: Enabling Application-Aware Integration[C]// 2025 International Conference on Networking, Sensing and Control (ICNSC). Oulu, Finland: IEEE, 2025: 307–314.

[3] Matko V, Brezovec B, Milanovič M. Intelligent Monitoring of Data Center Physical Infrastructure[J]. Applied Sciences, 2019, 9(23): 4998.

[4] Feng T, Sun H Z, Lu S. Image-based methods for data centre equipment status monitoring: a review[C]// 12th International Conference on Renewable Power

Generation (RPG 2023). Shanghai, China: IET, 2023: 1195–1201.

[5] Balabanov G, Apostolov N S. Data Centre Infrastructure Monitoring[C]// 2025 60th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST). Ohrid, North Macedonia: IEEE, 2025.

[6] He C, Song X, Rao Y, et al. An AR Operation and Maintenance System for Data Center[C]// 2018 International Conference on Computer Science and Software Engineering (CSSE 2018). Lancaster, PA, USA: DEStech Publications, 2018: 165–177.

[7] Zhang T, Jia K. Design and Implementation of IDC Engine Room Intelligent Management and Control System Based on Data Mining[C]// 2012 International Conference on Computer Science and Service System (CSSS). Nanjing, China: IEEE, 2012: 1366–1369.

[8] Królczyk J B, Legutko S, Wojtecki D. Implementation and Benefits of Introducing a Computerised Maintenance Management System into a Manufacturing Company[J]. Applied Mechanics and Materials, 2015, 809–810: 1354–1359.

[9] Mohd Noor H, Mazlan S A, Amrin A. Computerized Maintenance Management System in IR4.0 Adaptation - A State of Implementation Review and Perspective[J]. IOP Conference Series: Materials Science and Engineering, 2021, 1051(1): 012019.

[10] Wang T, Qi G, Wu T. KGroot: Enhancing Root Cause Analysis through Knowledge Graphs and Graph Convolutional Neural Networks[J]. arXiv preprint arXiv:2402.13264, 2024.

[11] Lewis P, Perez E, Piktus A, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks[C]// Advances in Neural Information Processing Systems 33 (NeurIPS 2020). 2020: 9459–9474.

[12] Zhang L, Jia T, Jia M, et al. A Survey of AIOps in the Era of Large Language Models[J]. ACM Computing Surveys, 2025, 58(2): Article 44, 1–35.

[13] Ahmed T, Ghosh S, Bansal C, et al. Recommending Root-Cause and Mitigation Steps for Cloud Incidents using Large Language Models[C]// Proceedings of the 45th International Conference on Software Engineering (ICSE 2023). Melbourne, Australia: IEEE, 2023: 1737–1749.

[14] Gao Y, Xiong Y, Gao X, et al. Retrieval-Augmented Generation for Large Language Models: A Survey[J]. arXiv preprint arXiv:2312.10997, 2023.