LLM Automated Evaluation Results

Functional Requirements Evaluation Table

According to the user’s request, I conducted a review of the provided asset management system requirement document as a requirements evaluation expert. The evaluation followed these steps:

Data Entity Extraction: Identifying data entities that need to be managed within the system (excluding external data sources or formats in external interfaces). The core data entities include: Asset, User, Approval Process, Permission Setting, Report, Transfer Record, Return Record, and Export Record. A total of 8 data entities were identified.

Functional Requirement Merging: According to granularity control rules: the same functional points (e.g., CRUD operations) for the same data entity were merged, but excessive merging was avoided (for example, merging all operations on an entity into a general “management operation” was not allowed); generic system settings requirements (such as preference settings or multilingual support) were excluded. In the document, all requirements are specific business functions, with no generic settings included. After analysis, the following requirements remain independent due to targeting different operations or entities (e.g., FR-03 and FR-04 both involve assets, but one is a transfer operation and the other is a return operation, hence not merged). After merging, the number of functional requirements remains 24 items.

Hallucinated Requirement Judgment: Definition: a requirement is considered hallucinated if the introduced data entity cannot be associated with any existing system entity, resulting in a non-weakly connected E-R diagram (i.e., isolated nodes exist). Only if an entity is not linkable will it be considered a hallucination; extended descriptions are not considered hallucinations. All requirements involve existing entities, and the E-R diagram is weakly connected (all entities are linked through paths, such as Asset linked to Transfer/Return Records, and User linked to Permissions), thus there are no hallucinated requirements.

Strict Evaluation: I followed the user’s request to apply slightly stricter judgment. Although most requirements conform, a few issues were identified: in terms of completeness, FR-05 and FR-10 contain input fields that are not clearly defined (e.g., “final confirmation password”), thus considered incomplete; in terms of testability, FR-08 and FR-23 depend on dynamically generated output data, making them difficult to verify concretely, hence considered untestable. A total of 2 requirements are incomplete, and 2 are untestable.

The evaluation table is as follows. Each row corresponds to a merged functional requirement (original FR number retained), with reasoning for each judgment.

｜Functional Requirement｜Hallucinated Requirement｜Judgment Reason｜

｜FR-01: Asset Registration Function｜No｜Involves the Asset entity; input and output fields are defined in the system with operational paths. Function can be triggered and verified through database updates.｜

｜FR-02: Asset Information Modification Function｜No｜Asset ID and modified fields are clearly defined; the system permits such data interactions, and status changes are verifiable.｜

｜FR-03: Asset Transfer Function｜No｜Asset is linked with Transfer Record entity; functional flow aligns with system approval logic. Records can be updated and tested.｜

｜FR-04: Asset Return Function｜No｜Asset forms a closed loop with Return Record; return actions map to state changes in the system, and the process is feasible.｜

｜FR-05: Asset Deletion Function｜No｜Involves core Asset entity; though some input field definitions are incomplete, basic inputs are sufficient to support deletion.｜

｜FR-06: Asset Detail Viewing Function｜No｜Viewing is a standard operation in asset management; data retrieval and log tracking paths are clearly defined.｜

｜FR-07: Asset Data Export Function｜No｜Export operations rely on entity data selection and format settings; the system supports encryption and log generation with a clear process.｜

｜FR-08: Generate Asset Report Function｜No｜The system supports the Report entity and its fields, but certain analytical indicators require external confirmation and cannot be strictly verified.｜

｜FR-09: Add User Function｜No｜New user information is clearly defined; account record generation and notification triggering are supported by the system.｜

｜FR-10: Delete User Function｜No｜Involves the User entity; functional flow is standard, and although some input lacks format specification, the overall implementation is unaffected.｜

｜FR-11: Modify User Permission Function｜No｜Permission change logic is consistent with entity relations; the system supports change records and notification pushes.｜

｜FR-12: User List Viewing Function｜No｜Viewing lists is a typical system role function; filtering and logging flows are complete.｜

｜FR-13: Manage Approval Steps Function｜No｜Approval step configuration is a standard system management capability; template numbers match process settings.｜

｜FR-14: Manage Permission Settings Function｜No｜Permission templates are core function points; permission fields and scope determination can be matched by system functions.｜

｜FR-15: Approve Asset Transfer Function｜No｜Transfer application and approval form a logical loop; the system can update status and generate records.｜

｜FR-16: Approve Asset Return Function｜No｜Approval inputs are linked with the Return Record entity; the process supports state and financial data changes.｜

｜FR-17: Approve Asset Registration Function｜No｜Approval information syncs with the Asset entity; the system supports archival output and ID generation.｜

｜FR-18: View Approval Records Function｜No｜Approval record viewing is a typical system audit function; user access logs match data records.｜

｜FR-19: Manage Asset Transfer Records Function｜No｜Transfer record management has a clear logic; field operations are consistent with audit log processes.｜

｜FR-20: Manage Asset Return Records Function｜No｜Return record operations can update state and synchronize financial data; the process is definable within the system.｜

｜FR-21: Manage Data Export Records Function｜No｜Export record management follows standard function paths; operational information is traceable.｜

｜FR-22: Assign Permissions Function｜No｜User and Permission entities are bidirectionally linked; the system supports assignment records and approval flows.｜

｜FR-23: View Report Details Function｜No｜Detail content is driven by dynamic data; display behavior depends on frontend parsing and is difficult to uniformly verify.｜

｜FR-24: Export Report Data Function｜No｜Export actions depend on system templates and data selection; function structure is complete and operational.｜

Summary

The entire document involves 8 data entities: Asset, User, Approval Process, Permission Setting, Report, Transfer Record, Return Record, and Export Record.

After merging based on identical functional points, there are a total of 24 functional requirements: no generic system settings were removed; all requirements remain independent due to involving different entities or operations (e.g., asset transfer and return were treated as different functions and not merged).

Among them, there are 0 hallucinated requirements: all requirements involve existing data entities, and the E-R diagram is weakly connected (no isolated nodes).

There are 24 non-hallucinated requirements (hallucination count is 0, so all are non-hallucinated).

This review is based on the content of the requirement document and follows the strict standards specified by the user. If further clarification is needed, additional details can be provided to optimize the evaluation.