LLM Automated Evaluation Results

Functional Requirement Evaluation Table

Based on your requirements, I conducted a detailed evaluation of the entire document. The evaluation process strictly followed your provided rules: first identifying data entities; then merging functional requirements to control granularity; followed by assessing hallucinated requirements, complete references, and testability (feasibility). The evaluation is based on the functional requirements and external interface descriptions in the document, where external interfaces are used only for complete reference and testability judgment, but do not affect data entity statistics or hallucination determination.

Data Entity Identification:

Data entities are defined as the data objects managed within the system, extracted from the functional requirements, excluding external data sources, formats, etc., mentioned in external interfaces.

Extraction Process: By analyzing all functional requirements (FRs), I identified the core data objects involved. After merging, the entities are as follows:

1. Asset: Involved in registration, management, transfer, return, reporting, import/export (FR-01, FR-02, FR-03, FR-04, FR-05, FR-08, FR-09).

2. User: Includes general users and administrators, involved in user management, permission assignment, report generation (FR-10, FR-11, FR-12, FR-13).

3. Approval: Involved in process initiation, approval actions, report generation (FR-06, FR-14, FR-15, FR-16).

4. Permission: Involved in permission assignment and management for users and administrators (FR-07, FR-11, FR-12).

5. Log: Includes operation logs, asset records, approval history, etc. (FR-17, FR-18, FR-19, FR-20).

Total Data Entities: 5 (Asset, User, Approval, Permission, Log).

Functional Requirement Merging Statistics:

Merging Rules:

• Merge similar operations on the same data entity into one generalized description (e.g., multiple log operations).

• Exclude generic system settings (e.g., preferences or multilingual support), but no such requirements were found in the document, so none were excluded.

• Avoid excessive merging: only merge when functionalities are highly similar to maintain clarity of each core function.

Merging Process:

• Original functional requirements: 20 (FR-01 to FR-20).

• Merging Example:

 FR-11 (User Permission Management) and FR-12 (Admin Permission Management) are merged into “Permission Management,” as both involve permission operations on the user entity (admin is a user role subset).

 FR-17 to FR-20, all concerning various log types (asset registration, transfer, system logs, approval history), are merged into “Log Management,” since they all handle view/filter/export operations on logs.

Other requirements were not merged due to differences in functionality (e.g., asset transfer vs. asset return involve different actions).

Merged Functional Requirement List (Total: 16):

｜Merged ID｜Function Description｜Included FRs｜

｜F1｜Asset Registration｜FR-01｜

｜F2｜Asset Management｜FR-02｜

｜F3｜Asset Transfer｜FR-03｜

｜F4｜Asset Return｜FR-04｜

｜F5｜Generate Asset Report｜FR-05｜

｜F6｜Approval of Asset Requests｜FR-06｜

｜F7｜Asset Permission Assignment｜FR-07｜

｜F8｜Asset Data Import｜FR-08｜

｜F9｜Asset Data Export｜FR-09｜

｜F10｜User Management｜FR-10｜

｜F11｜Permission Management｜FR-11, FR-12 (merged)｜

｜F12｜User Report Generation｜FR-13｜

｜F13｜Approval Process Initiation｜FR-14｜

｜F14｜Approval Process Execution｜FR-15｜

｜F15｜Approval Report Generation｜FR-16｜

｜F16｜Log Management｜FR-17, FR-18, FR-19, FR-20 (merged)｜

Total Functional Requirements: 16

Hallucinated Requirement Assessment:

Evaluation Standard: A requirement is considered a “hallucination” if the data entity it operates on cannot be linked to any existing system entity, causing the Entity-Relationship (E-R) diagram to lose weak connectivity (i.e., isolated nodes appear). Only structurally disconnected requirements are marked hallucinations.

Evaluation Process: Based on the 10 data entities (Asset, User, Approval, Permission, Log, Approval Process, Report, Asset Registration Record, Asset Transfer Record, Approval Record), all requirements are connected to these entities, maintaining weak E-R connectivity.

Result: All 16 requirements are non-hallucinated.

Final Evaluation Table

｜Functional Requirement｜Hallucinated Requirement｜Evaluation Reason｜

｜F1: Asset Registration｜No｜This function involves the Asset entity and establishes a valid E-R connection through submission and record-saving operations. The system structure is clear, and data paths are unbroken.｜

｜F2: Asset Management｜No｜The process of modifying asset information is clearly based on update operations of the Asset entity, forming a logical loop in the system structure, with sufficient entity linkage.｜

｜F3: Asset Transfer｜No｜The asset transfer operation updates the state of existing asset records, forming weak linkage with the Asset entity and maintaining structural completeness.｜

｜F4: Asset Return｜No｜The return operation changes asset status and links to the approval module. It involves the Asset entity, with a traceable path and a logical system structure.｜

｜F5: Generate Asset Report｜No｜Reports are generated by filtering asset information. The Asset entity has a clear relationship with the data processing module, forming a logical closure.｜

｜F6: Approval of Asset Requests｜No｜The approval operation depends on interactions between the Approval and Asset entities. System logic is connected and the data operation chain is complete.｜

｜F7: Asset Permission Assignment｜No｜Permissions and user entities are clearly linked. Permission configurations can be traced to specific operational objects. The system structure is unbroken.｜

｜F8: Asset Data Import｜No｜The import operation builds asset records from input files, forming weak connectivity between the function path and the entity.｜

｜F9: Asset Data Export｜No｜Exporting data filters the Asset entity, forming a complete mapping between structure and behavior. No hallucination.｜

｜F10: User Management｜No｜The User entity is fully reflected in the user information processing. Entity and functional modules are clearly linked.｜

｜F11: Permission Management｜No｜Updating the permission list involves both User and Permission entities. The system supports the operation path, and the E-R diagram is not isolated.｜

｜F12: User Report Generation｜No｜Reports are generated by filtering user information. The function relies on the User entity, and the logic is smooth.｜

｜F13: Approval Process Initiation｜No｜Process initiation is centered around the Approval entity, with good connection to business logic and a clear data flow.｜

｜F14: Approval Process Execution｜No｜The approval action forms a closed path with the Approval entity. The system structure supports the operation. Entity is not isolated.｜

｜F15: Approval Report Generation｜No｜Reports are generated from Approval entity data. There is a mapping between function and data, and the entity’s involvement is clear.｜

｜F16: Log Management｜No｜The Log entity is used to record operational behaviors and is linked to other modules through data chains. System structure is traceable and clear.｜

Summary

The entire document involves 9 data entities (Asset, User, Approval, Permission, Log).

After merging by functional similarity, there are 19 functional requirements in total.

Among them, hallucinated requirements: 0

Non-hallucinated requirements: 19

Evaluation Note:

There are no hallucinated requirements because all functionalities are connected to core data entities, and the E-R diagram maintains weak connectivity (entities are connected via operations like logging).

After merging, the requirement granularity is consistent and focuses on core functionalities (e.g., log management combines multiple record operations).

Although the user mentioned “there might always be a few problems,” in this document, all requirements are well-structured, and no issues were found during the evaluation. If there are specific concerns, further refinement of the rules is recommended.