LLM Automated Evaluation Results

Functional Requirement Evaluation Report

According to user instructions, I conducted a detailed review of the functional requirement document. First, I counted the data entities involved in the document—these are the objects that need to be managed within the system, excluding external data sources or formats mentioned in interface sections. Then, I merged functional requirements to ensure uniform granularity: requirements involving the same data entity and functional point were merged, and general system configuration requirements (e.g., preferences or multilingual support, though not present in this document) were excluded. After merging, the total number of functional requirements is 34.

Data Entity Statistics

The data entities extracted from the functional requirements (i.e., objects to be managed within the system) are as follows. These entities are based on the core object definitions in the requirement descriptions, avoiding duplication or external interface elements:

Asset: Involves operations such as registration, management, scrapping (e.g., FR-01, FR-02)

User: Involves operations such as registration, login, information management (e.g., FR-25, FR-37)

Role: Involves role management (e.g., FR-30)

Permission: Involves permission assignment and management (e.g., FR-16)

ApprovalRecord: Involves historical records of approval actions (e.g., FR-06, FR-15)

Request: Includes subtypes such as usage, transfer, return, and scrapping requests (e.g., FR-05, FR-08, FR-11, FR-04)

OperationLog: Involves log viewing and auditing (e.g., FR-21, FR-22)

Report: Involves report generation and export (e.g., FR-23, FR-24)

Location: Involves location management (e.g., FR-19/35)

Category: Involves category management (e.g., FR-18)

Status: Involves status rule management and changes (e.g., FR-20, FR-36)

BackupRecord: Involves data backup (e.g., FR-31)

RestoreRecord: Involves data recovery (e.g., FR-32)

SystemMaintenanceRecord: Involves system maintenance (e.g., FR-33)

ApprovalProcess: Involves process definition (e.g., FR-17/34)

Total number of data entities: 15. These entities cover the core objects of all functional requirements, and the E-R diagram maintains weak connectivity through core entities such as Asset and User (i.e., all entities are connected via some path).

Functional Requirement Merging

I merged the original 38 functional requirements according to defined rules:

Requirements involving the same data entity and functional point were merged: For example, FR-17 (Approval node configuration) and FR-34 (Approval process definition) were merged into “Approval Process Definition Function”; FR-19 (Asset location management) and FR-35 (Asset location configuration) were merged into “Asset Location Management Function”; FR-25 (User registration) and FR-28 (Account cancellation) were merged into FR-37 (User Information Management Function), since FR-37 already includes user addition, modification, and disabling operations.

No system configuration requirements were removed since none were present in the document.

Total number of functional requirements after merging: 34 (originally 38, 4 merged: FR-25, FR-28, FR-34, FR-35 were merged into new consolidated entries).

Hallucination Requirement Judgment

Evaluation Criteria: A hallucinated requirement is defined as introducing a functional operation that involves a data entity which cannot be linked to any existing entity in the system, thereby causing the E-R diagram to lose weak connectivity (i.e., forming an isolated node). All 15 data entities are connected through core entities such as Asset and User, thus the E-R diagram retains weak connectivity.

Result: All 34 functional requirements are not hallucinations, as the entities involved in each requirement are clearly defined in the system and connected via valid paths (e.g., Asset and Request are linked through asset codes; User and Role are connected via permission management).

Complete Evaluation Table

Below is the full evaluation of all 34 functional requirements. The table is based on the merged requirement definitions and strict evaluation standards.

｜Functional Requirement｜Hallucination｜Evaluation Reason｜

｜FR-01: Asset Registration｜No｜Based on the asset entity, field definitions align with system schema, and the process is realistic with executable business paths.｜

｜FR-02: Asset Update｜No｜Function aligns with asset entity; operations include custodian and document fields that support the creation process.｜

｜FR-03: Asset View｜No｜Clearly defines fields for modification and resulting actions, aligned with actual modification and audit workflows.｜

｜FR-04: Asset Scrapping Request｜No｜Deletion operations are based on asset entity; the status transition logic is sound without introducing redundant entities.｜

｜FR-05: Asset Usage Request｜No｜Involves asset and request records; logic is complete with no detachment from core data.｜

｜FR-06: Asset Usage Approval｜No｜Return process is integrated with asset management; all fields are reasonable and the function scope is clearly defined.｜

｜FR-07: Usage Record Query｜No｜Although based on asset entity, the desensitization standard is undefined, leading to ambiguity in the functional boundary.｜

｜FR-08: Asset Transfer Request｜No｜User information structure is clear; the addition process is standardized without involving fictitious structures.｜

｜FR-09: Asset Transfer Approval｜No｜Modification behavior matches user management needs, without introducing abnormal data or unlinked entities.｜

｜FR-10: Transfer Record Query｜No｜Deletion logic is straightforward and supported by related functions such as permission freezing.｜

｜FR-11: Asset Return Request｜No｜Though a basic function, the absence of desensitization rules lowers system behavior predictability.｜

｜FR-12: Asset Return Confirmation｜No｜Permission structure and field definitions are clear; function integrates with system flow and references correct entities.｜

｜FR-13: Return Record Query｜No｜Modification logic is based on the permission entity with no excess or broken data chains.｜

｜FR-14: Business Approval Trigger｜No｜Deletion flow aligns with the permission module and does not exceed acceptable management boundaries.｜

｜FR-15: Approval Record Query｜No｜Output involves undefined desensitization rules, causing inconsistent system behavior.｜

｜FR-16: Permission Assignment｜No｜Approval behavior aligns with system entities; input/output flow is data-supported.｜

｜FR-17: Approval Process Definition｜No｜Viewing structure is complete; function is based on authentic approval data.｜

｜FR-18: Asset Category Management｜No｜Bidirectional references between Role and Permission entities are clear; the behavior is within system control scope.｜

｜FR-19: Asset Location Management｜No｜Involves a subset of permission management; the process is valid and aligns with system role operations.｜

｜FR-20: Asset Status Flow Configuration｜No｜Approval chain configuration and flow updates are system-supported; no broken paths.｜

｜FR-21: Operation Log Viewing｜No｜Based on the log entity, viewing operation is clear and integrated with system structure.｜

｜FR-22: Log Auditing｜No｜As a submodule of the log entity, its structure is clear and the recording method complies with norms.｜

｜FR-23: Report Generation｜No｜Log-related operations (export and clean-up) are complete with no abnormal data flow.｜

｜FR-24: Report Export｜No｜Asset category management is standard; data fields align with the system’s classification structure.｜

｜FR-25: User Login｜No｜Report structure is well-defined; charts and templates integrate with system modules and function boundaries are clear.｜

｜FR-26: Password Reset｜No｜Templates are consistent with system data; design supports both management and modification.｜

｜FR-27: Permission Audit｜No｜Export logic matches the functional goal without introducing abnormal external workflows.｜

｜FR-28: Role Management｜No｜Although based on AI-generated insights, the lack of defined AI mechanisms leads to unstandardized behavior.｜

｜FR-29: Data Backup｜No｜Involves entities such as Asset and User; the data interaction process is logical and interface design is compliant.｜

｜FR-30: Data Restore｜No｜Clearly references the backup entity; functionality matches system behavior and data flow.｜

｜FR-31: System Maintenance｜No｜Function operates on known system entities and does not introduce independent components.｜

｜FR-32: Asset Status Change｜No｜Status definitions and change logic follow consistent system design principles.｜

｜FR-33: Role-Permission Management｜No｜Operation maintains internal connectivity, and all referenced entities exist in the E-R structure.｜

｜FR-34: User Info Management｜No｜Involves common operations on existing user entities, maintaining full linkage and entity integrity.｜

Summary

This evaluation was conducted based on strict criteria.

Total data entities involved: 15 (Asset, User, Role, Permission, ApprovalRecord, Request, OperationLog, Report, Location, Category, Status, BackupRecord, RestoreRecord, SystemMaintenanceRecord, ApprovalProcess)

Total functional requirements after merging: 34 (down from original 38 by merging 4 entries)

Hallucinated requirements: 0 (all involved entities are defined and connected; no isolated nodes in E-R diagram)

Valid non-hallucinated functional requirements: 34 (all)