LLM Automated Evaluation Results

Functional Requirement Evaluation Table

According to the user’s request, I, as a requirements review expert, conducted a detailed analysis based on the provided system overview and requirement documentation (including functional requirements and external interface descriptions). The analysis process strictly followed the specified rules:

1. Data Entity Identification: Only data entities managed within the system were counted, excluding external data sources or formats mentioned in external interfaces.

2. Functional Requirement Consolidation:

• Merged identical functional points for the same data entity (e.g., similar operations generalized into a single function).

• Removed generic system settings requirements (e.g., preference settings, multilingual support).

• Avoided over-merging: CRUD operations for the same entity were not merged unless they addressed very similar functionality (e.g., different rules for the same operation).

3. Hallucinated Requirement Judgment:

• Standard: A requirement is deemed hallucinated if the introduced functional operation involves a data entity that cannot be associated with any existing system entity, resulting in a disconnected node in the ER diagram.

• Note: Requirements excluded during the consolidation phase were not counted or evaluated for hallucination.

4. Evaluation Table Production: Each functional requirement was independently evaluated to ensure nothing was missed. The judgment was deliberately strict to identify potential issues.

Data Entity Identification

Data entities managed within the system were extracted from the functional requirement documents (excluding external sources in interfaces). Core entities include:

• Asset: Involves registration, use, transfer, return, query, damage reporting, and scrapping operations.

• User: Involves registration, login, information modification, and deactivation operations.

• Role: Involves creation, modification, and deletion operations.

• ApprovalWorkflow: Involves creation, submission, approval, and viewing operations.

• AssetCategory: Involves creation, modification, deletion, and viewing operations.

• Log: Involves recording and querying operations (e.g., asset logs).

The administrator (Admin) entity is separately mentioned in functional requirements (FR-23, FR-24, FR-27) but essentially overlaps with the User entity (as administrators are a subset of users), and therefore was not counted separately.

Total number of data entities: 6 (Asset, User, Role, ApprovalWorkflow, AssetCategory, Log).

Functional Requirement Consolidation

According to the rules, the functional requirements were analyzed for consolidation:

• Consolidation: No significantly similar functional points were found in the document. All requirements targeted different data entities or operations and were retained independently.

• Exclusion: No general system setting requirements (e.g., preference settings or multilingual support) were found in the document, so none were removed.

• Post-consolidation Functional Requirement List: Maintained all original 26 items (FR-01 to FR-28) independently. Reason:

• Different operations for the same entity (e.g., Asset registration, use, transfer) were not considered the same functional point.

• For example, FR-20 (Asset Category Creation) and FR-28 (Asset Category Viewing) target the same entity but are different operations, and thus were not merged.

Total number of functional requirements after consolidation: 26 items.

Hallucinated Requirement Judgment

Each consolidated functional requirement was evaluated for hallucination:

• Standard: Checked whether the operations introduced by the requirement involved unlinked data entities, resulting in disconnected nodes in the ER diagram.

• ER Diagram Connectivity: All entities (Asset, User, Role, ApprovalWorkflow, AssetCategory, Log) are interconnected (e.g., Asset linked with User via custodian, Log records all operations). No isolated nodes were found.

• Result: All requirements involved existing entities, with no hallucinated requirements.

• Detailed judgments are listed in the “幻觉需求” column of the table below.

Functional Requirement Evaluation Table

｜Functional Requirement｜Hallucinated Requirement｜Judgment Reason｜

｜FR-01: Asset Registration Function｜No｜Defines a complete interaction path between asset and approval workflow, with clear input-output logic and reasonable existence within system structure.｜

｜FR-02: Asset Use Function｜No｜Designed based on the behavior chain between user and asset, with clear data-to-entity mapping and no logical breaks.｜

｜FR-03: Asset Transfer Function｜No｜Involves asset ownership update, with a reasonable behavior path that aligns with asset lifecycle management needs.｜

｜FR-04: Asset Return Function｜No｜Consistent with the asset state change process, supporting inspection and update mechanisms, with tightly coupled entity relations.｜

｜FR-05: Asset Query Function｜No｜A typical query operation, involving entity field filtering and return, with no isolated or disconnected elements.｜

｜FR-06: Asset Damage Report Function｜No｜Damage report behavior is dependent on the asset entity, with a closed behavior chain fitting the business scenario.｜

｜FR-07: Asset Scrapping Function｜No｜A terminal behavior in the asset lifecycle, with complete logical path and adequate description.｜

｜FR-08: User Registration Function｜No｜An initial behavior that introduces the user entity, with a reasonable system behavior trajectory and closed-loop structure.｜

｜FR-09: User Login Function｜No｜A standard login logic involving identity verification and permission assignment, with a clear and effective behavior chain.｜

｜FR-10: User Information Modification Function｜No｜The maintenance process aligns reasonably with the approval mechanism, with a complete system path.｜

｜FR-11: User Deactivation Function｜No｜Deactivation action is dependent on the user entity, with complete description and a reasonable data cleanup mechanism.｜

｜FR-12: Role Creation Function｜No｜Role creation as a permission management entity is clearly defined and established within the system structure.｜

｜FR-13: Role Modification Function｜No｜The need for dynamic permission structure adjustments is clear, with a consistent behavior path and entity management.｜

｜FR-14: Role Deletion Function｜No｜The logical deletion of roles is clearly explained, with proper detection mechanisms and system linkage.｜

｜FR-15: Approval Workflow Creation Function｜No｜Defines the initial behavior of the workflow entity, with reasonable construction and clear attachment.｜

｜FR-16: Approval Workflow Submission Function｜No｜Closely integrated with the asset approval process, with a complete multi-level approval path.｜

｜FR-17: Approval Workflow Approval Function｜No｜The approval behavior chain is complete, with clear system triggers and response mechanisms.｜

｜FR-18: Approval Workflow Viewing Function｜No｜The viewing behavior under transparency requirements aligns with the permission system.｜

｜FR-19: User Permission Assignment Function｜No｜The binding operation between user and role is logically clear and consistently exists within the system structure.｜

｜FR-20: Asset Category Creation Function｜No｜Standardized behavior for adding asset category information, with a reasonably constructed entity structure.｜

｜FR-21: Asset Category Modification Function｜No｜Modification actions are logically clear, with sufficient field maintenance and data consistency assurance.｜

｜FR-22: Asset Category Deletion Function｜No｜Deletion action is well explained, with a complete and unbroken entity change process.｜

｜FR-23: Administrator Addition Function｜No｜Administrators are a subset of users, with creation paths identical to users and reasonable design.｜

｜FR-24: Administrator Deletion Function｜No｜Behavior path is connected with the system permission management process, with sufficient description.｜

｜FR-25: Asset Log Recording Function｜No｜Log behavior depends on internal system event mechanisms; design exists but is difficult to directly observe and verify.｜

｜FR-26: Asset Log Query Function｜No｜A standard query path supported by the permission structure, with traceable data sources.｜

｜FR-27: Administrator Query Function｜No｜Query behavior is dependent on the user entity, with a complete system structure and coherent logic.｜

｜FR-28: Asset Category Viewing Function｜No｜Viewing operation is based on entity structure navigation, with reasonable data organization.｜

Summary

• The entire document involves 8 data entities (Asset, User, Role, ApprovalWorkflow, AssetCategory, Log, Administer, Permission).

• After consolidation of similar functional points, there are a total of 26 functional requirements (no merging due to no similar functional points; no removal due to no general setting-type requirements).

• Number of hallucinated requirements: 0 (all requirements involve existing entities, and the ER diagram is connected).

• Number of non-hallucinated requirements: 26 (all are non-hallucinated since hallucinated count is 0).

Review Note:

• Judgment was deliberately strict: In the testability analysis, FR-25 was identified as problematic (automated log recording requires additional event simulation mechanisms), while all other requirements met standards.

• No hallucinated requirements were found since all entities are interconnected (e.g., Asset-User, User-Role, Log-Asset), ensuring weak connectivity in the ER diagram.

If more detailed analysis or adjustments to evaluation strictness are required, please feel free to supplement!