LLM Automated Evaluation Results

Functional Requirements Evaluation Table

Based on the system overview, functional requirement document, and external interface document provided by the user, I, as a requirements review expert, have conducted a detailed evaluation. The review process strictly follows the rules specified by the user:

• Data entity identification: Only internal data entities managed by the system (such as assets, users, etc.) are counted, excluding external data sources or formats mentioned in external interfaces.

• Functional requirement consolidation: Merge identical functional points for the same data entity (e.g., multiple requirements for the same asset operation are merged into one). Generic system configuration requirements (e.g., preferences or multilingual support) are excluded to avoid excessive merging.

• Hallucinated requirement judgment: A requirement is considered hallucinated if the data entities it involves cannot be associated with any existing entities in the system, resulting in an entity-relationship (E-R) diagram that is not weakly connected (i.e., isolated nodes exist). This judgment is based solely on structural criteria.

Data Entity Identification

Data entities managed internally by the system are extracted from the functional requirements, focusing on core data objects rather than operations or interfaces. The extracted entities include:

• Asset: involves operations such as registration, transfer, return, viewing, modification.

• User: involves registration, information modification, deletion, permission assignment.

• Role: involves permission configuration.

• Permission: involves assignment, modification, viewing.

• Approval: involves request submission, progress tracking, processing, and record viewing.

• Log: includes operational and audit logs.

• Report: includes asset reports and analytical reports.

• Category: involves asset classification management.

A total of 8 data entities. External data sources mentioned in interface documents (such as SMS verification service, database interface) are not counted.

Functional Requirement Consolidation and Statistics

There are 38 original functional requirements (FR-01 to FR-38). Based on the consolidation rules:

• Merge identical functional points: Same operations on the same data entity are merged (e.g., FR-15 and FR-33 both deal with asset report generation/export, merged into “Manage Asset Reports”; FR-23 and FR-38 both deal with asset classification, merged into “Manage Asset Categories”).

• Exclude general configuration: No generic configuration (e.g., preferences, multilingual) is present, so no requirement is removed.

• Avoid over-merging: Only highly similar requirements with the same entity and operation are merged (e.g., FR-24 and FR-36 both involve viewing operation logs, merged into “View Operation Logs”; FR-13 and FR-35 are not merged since they target different operations: progress and history record viewing).

After merging, the total number of functional requirements is 34. Merged items include:

• FR-15 (generate asset report) and FR-33 (export asset report) are merged as FR-15/33 (Manage Asset Reports).

• FR-23 (Manage Asset Categories) and FR-38 (Asset Classification Management) are merged as FR-23/38 (Manage Asset Categories).

• FR-24 (Audit Operation Log) and FR-36 (View Operation Log) are merged as FR-24/36 (View Operation Logs).

• FR-26 (Export Operation Log) and FR-37 (Export Operation Log) are merged as FR-26/37 (Export Operation Logs).

• All other requirements remain standalone with no further merging.

Hallucinated Requirement Judgment

• Criteria: If a functional requirement involves operations on data entities that cannot be linked with existing entities (Asset, User, Role, Permission, Approval, Log, Report, Category), leading to a disconnected E-R graph (i.e., isolated nodes), it is marked as hallucinated.

• Result: All 34 functional requirements are linked to existing data entities. The E-R graph is weakly connected: Asset and User are linked via transfer/return; Approval is linked with Asset/User; Permission is linked with Role/User; Log tracks all operations; Report and Category are linked with Asset. No isolated node exists, hence no hallucinated requirement.

Complete Evaluation Table

Each merged functional requirement is evaluated individually. Column explanations:

• Functional Requirement: Merged requirement ID and description.

• Hallucinated Requirement: Yes/No (based on E-R graph connectivity).

• Justification: Brief explanation.

｜Functional Requirement｜Hallucinated Requirement｜Justification｜

｜FR-01: Asset Registration Function｜No｜Involves the Asset entity; input/output clearly defined; file upload supported; registration result can be verified.｜

｜FR-02: Asset Transfer Function｜No｜Involves Asset and User entities; input/output logic is complete; database relationships validate transfer operations.｜

｜FR-03: Asset Return Function｜No｜Input/output process is clear; state updates are trackable; return operations are supported by records.｜

｜FR-04: View Asset Details Function｜No｜Query interfaces exist; data presentation supports detail view; output integrity is verifiable.｜

｜FR-05: Modify Asset Information Function｜No｜Modification operation is bound to form input; version change can be validated post-update; logic conforms.｜

｜FR-06: Export Asset Data Function｜No｜Export is performed based on filter conditions; output format is defined; conforms to asset export norms.｜

｜FR-07: User Registration Function｜No｜Involves the User entity; registration process is complete; user state and verification code flow are clearly linked.｜

｜FR-08: Modify User Information Function｜No｜Input for modification is clear; version records can validate updates; functional closure is ensured.｜

｜FR-09: Assign User Permissions Function｜No｜Permission configuration is clearly linked with the User entity; dynamic adjustments are supported; output is observable.｜

｜FR-10: View User List Function｜No｜Query mechanism supports customizable filters; user display structure supports browsing needs.｜

｜FR-11: Delete User Account Function｜No｜Deletion logic is straightforward; entity relationships are clear; state change is observable.｜

｜FR-12: Submit Approval Request Function｜No｜Approval process entry is standardized; approval form submission triggers request creation; logic is closed-loop.｜

｜FR-13: View Approval Progress Function｜No｜Approval ID maps to progress view; supports visualization of approval path.｜

｜FR-14: Process Approval Requests Function｜No｜Decision input updates approval status; process path is clear; response logic is predictable.｜

｜FR-15/33: Manage Asset Reports Function｜No｜Report management and generation process is well-structured; supports parameter configuration and export validation.｜

｜FR-16: Export Approval Records Function｜No｜Approval logs are structured for export; meets compliance or archival needs.｜

｜FR-17: Assign System Permissions Function｜No｜Permission granularity and configuration items are clearly defined; assignments can be logged.｜

｜FR-18: Modify Permission Settings Function｜No｜Permission updates are traceable; logic path is closed; operation results are clear.｜

｜FR-19: View Permission List Function｜No｜Permission view supports mode switching; structure is hierarchical and readable.｜

｜FR-20: Search Asset Information Function｜No｜Search logic is based on keywords and field matching; multi-condition search is supported.｜

｜FR-21: Filter Asset Records Function｜No｜Filter conditions are structured; logic combinations are supported; output is evidence-based.｜

｜FR-22: Bulk Update Asset Status Function｜No｜Batch processing is clearly defined; supports batch input and state changes.｜

｜FR-23/38: Manage Asset Categories Function｜No｜Category structure supports add/delete/edit; category info changes can be verified.｜

｜FR-24/36: View Operation Logs Function｜No｜Log records align with conditional queries; supports auditing of user actions.｜

｜FR-25: Configure Role Permissions Function｜No｜Role IDs are clearly linked to permission configuration; assignment actions are traceable.｜

｜FR-26/37: Export Operation Logs Function｜No｜Log export meets data backup or analysis needs; output format is standardized.｜

｜FR-27: User Login Function｜No｜Login flow and identity authentication mechanism are complete; token mechanism supports access control.｜

｜FR-28: User Logout Function｜No｜Logout process is closed-loop; user state changes verify logout success.｜

｜FR-29: Reset Password Function｜No｜Password reset logic integrates verification code and update flow; input/output is complete.｜

｜FR-30: Data Backup Function｜No｜Backup process is parameterized and well-defined; backup file generation is verifiable.｜

｜FR-31: Data Recovery Function｜No｜Recovery path is bound to backup data; process logic is reasonable; results are traceable.｜

｜FR-32: Configure Approval Processes Function｜No｜Approval process structure supports custom configuration; state changes are recorded clearly.｜

｜FR-34: Data Analysis Function｜No｜Analysis tasks support multi-dimensional configuration and report output; conforms to reporting logic.｜

｜FR-35: View Approval Records Function｜No｜Approval IDs map to historical records; record content is supported by visual presentation.｜

Summary

• A total of 8 data entities are involved: Asset, User, Role, Permission, Approval, Log, Report, Category.

• After merging identical functions, 34 functional requirements remain: Originally 38, 4 merged.

• Number of hallucinated requirements: 0: All are linked to existing data entities; E-R graph is weakly connected.

• Non-hallucinated requirements: 34: Since no hallucinations, all are valid.

Review Notes: The review process strictly follows user-defined rules. Hallucination judgment is based on structural criteria (E-R diagram). Although the user remarked that “normally not every requirement should be perfect,” this requirement document is well-designed. All requirements meet the specified standards, no omissions are found, and each requirement has been evaluated independently.