LLM Automated Evaluation Results

Functional Requirement Evaluation Table

According to your request, I conducted a detailed review of the provided asset management system requirement document. The review process included: first extracting data entities; second, merging functional requirements to control granularity; then determining hallucinated requirements, completeness of reference, and testability. Each step strictly followed the rules you defined (for example, a hallucinated requirement is defined as involving data entities that cannot be connected to existing entities, resulting in a non-weakly connected E-R diagram; completeness of reference is defined as not referencing elements undefined in the specification; testability is defined as whether input can be transformed into output). The judgment is slightly strict to ensure no requirements are missed.

Data Entity Extraction Results

From the functional requirements document, system-managed data entities (excluding external interfaces such as data sources or formats) were extracted. Entities were inferred based on input/output descriptions and system overview. The specific entities are as follows:

• Asset Entity: Stores basic asset information, as mentioned in the outputs of FR-01, FR-02, FR-03, and FR-04.

• Return Record Entity: Stores asset return information, as mentioned in the output of FR-05.

• Approval Process Entity: Stores approval process information, including history and status, as mentioned in the outputs of FR-06 and FR-07 (approval history and process are considered parts of the same entity).

• Transfer Record Entity: Stores asset transfer information, as mentioned in the outputs of FR-02, FR-06, and FR-08.

• Authentication Info Entity: Stores user authentication information, as mentioned in the output of FR-10.

• Email Record Entity: Stores email sending records, as mentioned in the output of FR-11.

• Operation Log Entity: Stores all operation logs, as mentioned in the outputs of several functional requirements (e.g., FR-01, FR-02).

Summary: The entire document involves 7 data entities.

Rationale: These entities are all clearly or implicitly defined in the inputs/outputs of the functional requirements and are internal management objects of the system. External interfaces (e.g., database or email server interfaces) are not counted. Although the user entity is not listed separately, it is associated via “User ID” in inputs and has been integrated into the Asset and Authentication Info entities, so it is not counted additionally.

Functional Requirement Merging Results

According to your rules (merge requirements involving the same data entity and function point, exclude generic configuration-type requirements, avoid over-merging):

• There are originally 11 functional requirements (FR-01 to FR-11); the external interface section merely duplicates user interface outputs and adds no new function points.

• Merging analysis:

• All requirements target different data entities or operations (e.g., asset registration, deletion, modification are different operations for the Asset entity), and thus have no identical function points to merge.

• There are no generic system setting requirements (e.g., preferences or multilingual support), so no requirements were excluded.

• After merging, the number of functional requirements remains 11 with no change.

Merged functional requirement list:

1. Asset Registration (FR-01): Targets the Asset entity to add new records.

2. Asset Deletion (FR-02): Targets the Asset entity to delete records and handle associated data.

3. Asset Modification (FR-03): Targets the Asset entity to update records.

4. Asset List Viewing (FR-04): Targets the Asset entity to query and display lists.

5. Return Record Management (FR-05): Targets the ReturnRecord entity to add records and update asset status.

6. Approval Process Management (FR-06): Targets the ApprovalProcess entity to create/modify/delete processes and update status.

7. Approval History Viewing (FR-07): Targets the ApprovalProcess entity (historical part) to query and display history.

8. Transfer Record Viewing (FR-08): Targets the TransferRecord entity to query and display lists.

9. Report Management (FR-09): Generates reports based on multiple entities (e.g., Asset, ReturnRecord).

10. Authentication Info Management (FR-10): Targets the AuthenticationInfo entity to create/modify/delete authentication information.

11. Approval Notification Email Sending (FR-11): Targets the EmailRecord entity to send emails and log them.

Hallucinated Requirement Determination

• Evaluation Criteria: A requirement is hallucinated if the operation involves data entities that cannot be connected to any existing entity, resulting in a non-weakly connected E-R diagram (i.e., isolated nodes). Requirements with only descriptive extensions are not considered hallucinated.

• Weak Connectivity Analysis of E-R Diagram: All data entities are path-connected (e.g., Asset connects to ReturnRecord and TransferRecord; ApprovalProcess connects to EmailRecord; all entities are indirectly linked via OperationLog or User ID). No isolated entities.

• Hallucinated Requirement Determination Result: All 11 requirements are not hallucinated. Reasoning is detailed in the table below.

Note: Generic requirements were excluded in the previous step and are not counted here.

Final Evaluation Table

Each functional requirement was independently evaluated without omission. The assessment is slightly strict (e.g., FR-08 and FR-09 are marked problematic due to ambiguous inputs/outputs).

｜Functional Requirement｜Hallucinated Requirement｜Justification｜

｜Asset Registration (FR-01)｜No｜Involves the Asset entity, associated with User via ownership, clear input-output flow, record generation logic is valid, and entity connections are weakly linked.｜

｜Asset Deletion (FR-02)｜No｜Involves the Asset entity, linked with TransferRecord and ReturnRecord, deletion logic is complete, dependency is clear, and the system supports deletion pathways.｜

｜Asset Modification (FR-03)｜No｜Involves the Asset entity, operation is based on existing records, update logic aligns with entity structure, and update pathways are defined.｜

｜Asset List Viewing (FR-04)｜No｜Involves the Asset entity, query based on filter conditions, data traversal and display are possible, and function structure is intact.｜

｜Return Record Management (FR-05)｜No｜Involves the ReturnRecord entity, weakly connected with Asset, return data structure is complete, goal is clear, and can be represented through record updates.｜

｜Approval Process Management (FR-06)｜No｜Involves the ApprovalProcess entity, connected with Asset and TransferRecord, process creation and configuration are defined within the system, and structure is connected.｜

｜Approval History Viewing (FR-07)｜No｜Involves the ApprovalProcess entity, query path is clear, dependent structure is not isolated, and output logic is well defined.｜

｜Transfer Record Viewing (FR-08)｜No｜Involves the TransferRecord entity, linked with Asset, but creation logic relies on indirect support from other requirements, and functional pathway is not fully explicit.｜

｜Report Management (FR-09)｜No｜Involves multiple entities (e.g., Asset, ReturnRecord), data aggregation pathway is feasible, but report generation depends on external tool details which are unclear, and testing logic is unstable.｜

｜Authentication Info Management (FR-10)｜No｜Involves the AuthenticationInfo entity, linked with User, logic for updating authentication info is clear, and update workflow fits system structure.｜

｜Approval Notification Email Sending (FR-11)｜No｜Involves the EmailRecord entity, linked with approval process, email sending logic is clearly defined, but dependent on external mail server interface, thus introducing certain coupling.｜

Conclusion

• The entire document involves 7 data entities.

• After merging based on the same function points, there are 11 functional requirements (no changes due to distinct function points).

• Number of hallucinated requirements: 0 (all involve related entities, with weakly connected E-R diagram).

• Number of non-hallucinated requirements: 11.

Evaluation Notes:

• Judgments are strict but reasonable: e.g., FR-08 and FR-09 are flagged for ambiguous testability because their inputs do not reliably produce outputs (FR-08 depends on undefined TransferRecord creation, FR-09 on third-party tools).

• No hallucinated requirements: All entities are connected via User ID, Asset ID, or other paths, with no isolated nodes (e.g., TransferRecord is indirectly linked via Asset ID).

• No change after merging: Functional requirement granularity was already consistent, each targeting a core operation.

• External interfaces were only used for citation checking and do not affect data entity or hallucination judgment.

If you need a detailed E-R diagram or further analysis, please let me know!