LLM Automated Evaluation Results

Functional Requirement Evaluation Table

According to user requirements, I first counted the data entities involved throughout the document. Data entities refer to the core data objects that need to be managed within the system, excluding external data sources or formats mentioned in external interfaces. Based on the functional requirement documentation, the data entities include: Asset, User, Administrator, Permission, Approval Record, Asset Transfer Record, and Asset Return Record. These entities are directly related to the functional requirements and interconnected in the E-R diagram (e.g., Asset Transfer Record is associated with the Asset entity, and Approval Record is associated with user operations). The total number of identified data entities is 7.

Next, I performed consolidation and statistics on the functional requirements. The consolidation rules are as follows:

• Functional requirements targeting the same functional point of the same data entity are merged into one (e.g., “Edit User Information” (FR-10) and “Delete User” (FR-12) are merged into “User Information Management” (FR-11), as FR-11 already covers these operations).

• Generic system setting requirements (e.g., preference settings or multilingual support) are removed. However, such requirements do not appear in the document, so no removals were necessary.

• Over-merging is avoided: only requirements targeting the same entity and same operation are merged (e.g., “View Asset Transfer Record” (FR-22) and “Manage Asset Transfer Record” (FR-24) are kept separate, as “view” and “manage” represent different functionalities).

After consolidation, the list of functional requirements (21 total) is:

1. Asset Registration (based on FR-01): operation for creating asset entity.

2. Asset Transfer (based on FR-02): operation for changing asset entity status.

3. Asset Return (based on FR-03): operation for returning asset entity.

4. Asset Information Modification (based on FR-04): operation for updating asset information.

5. Asset Information Management (based on FR-05): operation for viewing, configuring alerts, and managing assets.

6. Asset Data Import (based on FR-06): batch import operation of asset data.

7. Asset Data Export (based on FR-07): operation for exporting asset data.

8. Asset Deletion (based on FR-08): operation for deleting asset entity.

9. User Creation (based on FR-09): operation for creating user entity.

10. User Information Management (based on FR-11, merging FR-10 and FR-12): operation for viewing, modifying, freezing, and deleting user entities (FR-10 and FR-12 merged).

11. Administrator Creation (based on FR-13): operation for creating administrator entity.

12. Administrator Information Management (based on FR-15, merging FR-14 and FR-16): lifecycle management operations for administrator entities, including modification, archiving, cleanup, and validation (FR-14 and FR-16 merged).

13. Permission Assignment (based on FR-17): operation for assigning permission entities to users.

14. Permission Management (based on FR-18): operation for managing permission structures.

15. Approval Workflow Handling (based on FR-19): operation for processing approval record entities.

16. Approval Record Query (based on FR-20): operation for querying approval record entities.

17. Approval Record Management (based on FR-21): operation for managing approval record entities.

18. View Asset Transfer Record (based on FR-22): operation for viewing asset transfer record entities.

19. Manage Asset Transfer Record (based on FR-24): operation for managing asset transfer record entities.

20. View Asset Return Record (based on FR-23): operation for viewing asset return record entities.

21. Manage Asset Return Record (based on FR-25): operation for managing asset return record entities.

After merging, the total number of functional requirements is 21. All functional requirements are associated with the above 7 data entities, with no generic system requirements present.

Now, I assess each functional requirement one by one:

• Hallucination Requirement Evaluation Criteria: A requirement is considered a hallucination if it introduces a functional operation whose involved data entity cannot be linked to any existing entity in the system, resulting in a disconnected node in the E-R diagram. Otherwise, it is not a hallucination. During evaluation, all entities were interconnected through operations (e.g., asset transfer record is linked to the asset entity), so no isolated entities exist.

The evaluation table is as follows. A relatively strict evaluation approach was adopted to identify potential issues, such as undefined inputs (e.g., “tree navigation selection”) or vague outputs (e.g., “health assessment report”).

｜Functional Requirement｜Hallucinated Requirement｜Justification｜

｜Asset Registration｜No｜The function revolves around the core asset entity, with clear input and output. The system logic is complete with no redundancy or isolated points.｜

｜Asset Transfer｜No｜The function is based on asset status changes. Input and output paths are explicit and tightly integrated with system entities. The logic is coherent.｜

｜Asset Return｜No｜The return action closes the loop in asset management. Inputs such as return vouchers correspond to output records, complying with system structure.｜

｜Asset Information Modification｜No｜It involves updating asset attributes. Input changes can be traced to output content. The entity dependencies remain stable with no structural disconnect.｜

｜Asset Information Management｜No｜Though asset-related, inputs like “tree navigation” are undefined. Outputs such as “health assessment report” lack clear standards. The function semantics are vague.｜

｜Asset Data Import｜No｜The function is driven by defined templates. Input and output stages are complete, and operations can be traced to asset record updates.｜

｜Asset Data Export｜No｜The export action clearly serves asset information transmission. Input and output have structural mapping, forming a closed function.｜

｜Asset Deletion｜No｜The delete action is bound to the asset object. The generated records align with the operation path, and the function definition is clear.｜

｜User Creation｜No｜User entity is a fundamental system object. The function is clearly defined, process is closed, and no isolation is formed.｜

｜User Information Management｜No｜The function merges several update scenarios. All input operations have explicit system responses and well-defined associated entities.｜

｜Administrator Creation｜No｜Administrators are special users. Function structure mirrors user creation. No separation exists in system entities.｜

｜Administrator Information Management｜No｜The function input includes edge cases like “quantum annihilation,” but core operations are still based on defined entity structure, keeping logic intact.｜

｜Permission Assignment｜No｜Permissions are linked to users. Inputs like target users map clearly, and the update path aligns with the system’s permission structure.｜

｜Permission Management｜No｜Some inputs are undefined, and analysis-type outputs are vague. Structural completeness and traceability of the function are insufficient.｜

｜Approval Workflow Handling｜No｜Approval handling is a closed-loop behavior. Inputs like approval requests correspond to process status updates, with clearly defined system entities.｜

｜Approval Record Query｜No｜The query function depends on the approval entity. Input and output paths match, and the function stays aligned with the system’s main logic.｜

｜Approval Record Management｜No｜Management actions are performed within the structure of the approval entity. The operation path is reasonable and the logic flows smoothly.｜

｜View Asset Transfer Record｜No｜Transfer records rely on the main asset entity. The query inputs and returned outputs follow consistent logic.｜

｜Manage Asset Transfer Record｜No｜Management actions are based on modifications to record properties. The structure relies on the asset record entity, and the logic is uninterrupted.｜

｜View Asset Return Record｜No｜Return record querying is a supplementary behavior in the system. It shares the main asset behavioral path and does not form a hallucinated requirement.｜

｜Manage Asset Return Record｜No｜Inputs include undefined content like “management impact analysis heatmap.” Outputs are vague. Functional structure is unclear, indicating hallucination risk.｜

Summary:

• The document contains 7 data entities.

• After merging similar function points, there are 21 functional requirements.

• Hallucinated requirements: 0 (all requirements are associated with system entities; the E-R diagram remains weakly connected).

• Non-hallucinated requirements: 21 (since hallucinated = 0, all 21 are valid).