LLM Automated Evaluation Results

Functional Requirement Evaluation Table

Based on the system overview and functional requirement document provided by the user, the following assessment was conducted. First, I followed these steps:

1. Entity Identification: Core data entities managed within the system (excluding externally referenced data sources or formats) were extracted from the functional requirements. Data entities are core data objects explicitly operated on or maintained in the requirements, such as assets and users. A total of 18 entities were identified, including Asset, User, Permission, ApprovalRecord, AssetCategory, ApprovalProcessDef, EmailNotificationConfig, ImportExportRecord, EmailNotificationRecord, BackupRecord, RestoreRecord, ReportTemplate, and operational log entities like RegistrationRecord, TransferRecord, ReturnRecord, ScrapRecord, LoginLog.

2. Merging Functional Requirements: According to the rules, functionally similar requirements referring to the same data entity were merged (e.g., user permission assignment and revocation merged into “User Permission Management”). General configuration-type requirements (e.g., preferences or multilingual support) were excluded, though none were present in the document. After merging, 24 functional requirements remain (originally 27, with 4 pairs merged: FR-09/10, FR-13/14, FR-15/16, FR-17/18; resulting in a net reduction of 3 items).

3. Hallucinated Requirement Assessment: Based on the E-R graph weak connectivity principle. If a functional requirement introduces an entity that cannot be linked to any other system entity (causing E-R graph disconnection), it is considered a hallucinated requirement. After assessment, no hallucinated requirements were found, as all entities are connected through core modules such as User, Asset, or Approval.

Merged Functional Requirement List

After merging, there are 24 functional requirements. The list is as follows (merged only where the same data entity and function type are involved to avoid over-merging):

• FR-01: Asset Registration

• FR-02: Asset Query

• FR-03: Asset Transfer

• FR-04: Asset Return

• FR-05: Asset Scrap

• FR-06: User Registration

• FR-07: User Login

• FR-08: User Information Modification

• FR-09/10 (Merged): User Permission Management (Assignment and Revocation)

• FR-11: Approval Record Viewing

• FR-12: Approval Process Management (Initiation)

• FR-13/14 (Merged): Report Generation and Export

• FR-15/16 (Merged): Data Import and Export

• FR-17/18 (Merged): Data Backup and Recovery

• FR-19: Asset Category Management

• FR-20: Approval Process Management (Definition)

• FR-21: Email Notification Configuration

• FR-22: Import/Export Record Management

• FR-23: Authentication Information Management

• FR-24: Email Notification Management

• FR-25: Backup Record Management

• FR-26: Restore Record Management

• FR-27: Report Template Management

Evaluation Table

｜Functional Requirement｜Hallucinated Requirement｜Justification｜

｜FR-01: Asset Registration｜No｜Involves Asset and RegistrationRecord entities; operation path closely tied to core modules like User and Department; structurally connected.｜

｜FR-02: Asset Query｜No｜Query is based on the Asset entity and is structurally linked to the core system; entity relationship is complete.｜

｜FR-03: Asset Transfer｜No｜Transfer involves Asset and TransferRecord; effectively connected to User or Department; no risk of isolation.｜

｜FR-04: Asset Return｜No｜Return involves Asset and ReturnRecord; function traceable and structurally coherent.｜

｜FR-05: Asset Scrap｜No｜Scrapping uses Asset and ScrapRecord; tightly integrated with system logic.｜

｜FR-06: User Registration｜No｜User is a core system entity; registration is a key lifecycle stage.｜

｜FR-07: User Login｜No｜Login links User and LoginLog; logical structure valid, clearly routed.｜

｜FR-08: User Information Modification｜No｜Involves updating core attributes of User; structure is clear and forms a closed loop.｜

｜FR-09/10: User Permission Management｜No｜Permission and User have a clearly defined management relationship; system-locatable functionality.｜

｜FR-11: Approval Record Viewing｜No｜ApprovalRecord is derived from User or Asset; data structure complete with defined linkage.｜

｜FR-12: Approval Process Management (Initiation)｜No｜Initiation behavior driven by ApprovalRecord; entity embedded in core logic.｜

｜FR-13/14: Report Generation and Export｜No｜Reports are generated from User and Asset data; structural clarity and closed linkages confirmed.｜

｜FR-15/16: Data Import and Export｜No｜Data operations rely on foundational entities like Asset; part of the system’s main process.｜

｜FR-17/18: Data Backup and Recovery｜No｜Database is the system’s data carrier; connected to core entities, structurally supported.｜

｜FR-19: Asset Category Management｜No｜AssetCategory is an attribute of Asset; tightly coupled and not isolated.｜

｜FR-20: Approval Process Management (Definition)｜No｜ApprovalProcessDef governs approval logic and links to ApprovalRecord, forming a complete path.｜

｜FR-21: Email Notification Configuration｜No｜Configuration links to event entities; structurally mapped and reasonable.｜

｜FR-22: Import/Export Record Management｜No｜Records reflect actual import/export behavior; aligned with system data flow.｜

｜FR-23: Authentication Information Management｜No｜User authentication is part of lifecycle management; logical path is valid.｜

｜FR-24: Email Notification Management｜No｜EmailNotificationRecord corresponds to system alert behavior; data entity forms a closed loop.｜

｜FR-25: Backup Record Management｜No｜BackupRecord is part of system logs; structurally connected.｜

｜FR-26: Restore Record Management｜No｜RestoreRecord aligns with recovery operations and has defined system placement.｜

｜FR-27: Report Template Management｜No｜ReportTemplate serves the report generation function; logical relationship is sound, system path is closed.｜

Summary

The document includes 18 data entities: Asset, User, Permission, ApprovalRecord, AssetCategory, ApprovalProcessDef, EmailNotificationConfig, ImportExportRecord, EmailNotificationRecord, BackupRecord, RestoreRecord, ReportTemplate, RegistrationRecord, TransferRecord, ReturnRecord, ScrapRecord, LoginLog.

After merging functionally identical items, there are 24 functional requirements (originally 27, 4 pairs merged resulting in a reduction of 3 items).

There are 0 hallucinated requirements: all functional requirements are linked to core system entities like User and Asset; the E-R graph remains weakly connected (no isolated entities).

There are 24 non-hallucinated requirements (no hallucinations identified).