LLM Automated Evaluation Results

Functional Requirements Evaluation Table

Based on the user’s query, I, as a requirement review expert, conducted a detailed evaluation of the provided functional requirement document. The evaluation process included the following steps:

I first extracted core data entities from the document that need to be managed within the system (excluding external data sources or formats mentioned in external interfaces). A total of 14 data entities were identified: User, Email, Folder, Event, Contact, DistributionList, ReminderTask, Mailbox, SharedAccount, ArchivedEmail, BackupLog, RestoreLog, ArchiveLog, and ArchivePolicy.

Then, I merged functional requirements according to a defined rule: functional points involving the same data entity were merged when highly similar, but CRUD operations were not merged. General configuration-type requirements (such as preferences) were excluded, though none were found in this document. After merging, the total number of functional requirements was reduced from 45 to 33.

Examples of merging include: FR-01 (Create Email) and FR-02 (Send Email) were merged into “Email Creation and Sending” because they both involve the core email operation with overlapping input/output; FR-28 to FR-31, which are various types of email archiving, were merged into “Email Archiving Operations” as they serve the same purpose; FR-36, FR-37, and FR-41 (viewing different logs) were merged into “View Logs”; FR-39, FR-40, and FR-42 (deleting different logs) were merged into “Delete Logs”. Other requirements were not merged due to different operations or entities involved.

Regarding hallucinated requirements, I applied a structural standard: if a requirement involves an introduced entity that cannot connect to the existing system entities, forming a disconnected E-R diagram, it would be hallucinated. In this system, all entities are connected, especially through User, hence no hallucinated requirements exist.

The final evaluation table is presented below. Each requirement is based on the merged number (linked to original FR numbers). Evaluation reasons are brief and focus on key points.

｜Functional Requirement｜Hallucinated｜Evaluation Reason｜

｜Email Creation and Sending (FR-01,02)｜No｜Inputs and outputs are clearly defined, involving Email and User entities; functional chain is complete and system behavior is verifiable.｜

｜Receive Email (FR-03)｜No｜Entity relationships are clear, process is well-defined, and the feature can be tested by simulating email reception.｜

｜Search Email (FR-04)｜No｜Search criteria and return results match; the system can respond correctly to user input.｜

｜Manage Email Folders (FR-05)｜No｜Folder and Email relationship is clear; operations like move yield feedback and form a logical loop.｜

｜Add Event (FR-06)｜No｜Input content matches User/Event entities; system provides a response after creation.｜

｜View Event (FR-07)｜No｜Given an event ID, the system returns the corresponding event content; functional goal is clear.｜

｜Modify Event (FR-08)｜No｜Given update content, system returns a result; behavior aligns with expectations.｜

｜Delete Event (FR-09)｜No｜User provides ID for deletion; system responds accordingly, forming a closed data flow.｜

｜Add Contact (FR-10)｜No｜Contact information is complete; system feedback confirms the addition.｜

｜View Contact (FR-11)｜No｜Given a contact ID, complete information is returned; the process is clear.｜

｜Modify Contact (FR-12)｜No｜Modification includes input and response; entity relationships are well-defined.｜

｜Delete Contact (FR-13)｜No｜Given a contact ID, deletion yields a clear result; system behavior is logical.｜

｜Add Distribution List (FR-14)｜No｜Member list and distribution list create a complete operational chain; system logic is clear.｜

｜View Distribution List (FR-15)｜No｜Given a list ID, corresponding information is retrieved; entity connections are valid.｜

｜Modify Distribution List (FR-16)｜No｜Updated fields are processed by the system, and return values are clear.｜

｜Delete Distribution List (FR-17)｜No｜Deletion process is valid; system returns expected outcome; structure is complete.｜

｜Set Reminder Task (FR-18)｜No｜Reminder time and message form a full process; system responds consistently.｜

｜View Reminder Task (FR-19)｜No｜Task ID is linked to ReminderTask entity; system output matches the input.｜

｜Delete Reminder Task (FR-20)｜No｜System processes ID deletion correctly; the flow is complete.｜

｜Manage Admin Permissions (FR-21)｜No｜Permission model exists but lacks detail; effectiveness of permission operations is unclear.｜

｜Create Mailbox (FR-22)｜No｜Input-output flow is clear; system can create a mailbox object.｜

｜Manage Mailbox Permissions (FR-23)｜No｜Permission model is incomplete; actual effect of configuration cannot be determined.｜

｜Delete Mailbox (FR-24)｜No｜Deletion process is complete; inputs are clearly defined; behavior is observable.｜

｜Create Shared Account (FR-25)｜No｜Account structure is valid; system feedback is definable and logic is consistent.｜

｜Manage Shared Account Permissions (FR-26)｜No｜Permission types are undefined; system response lacks verifiable evidence.｜

｜Delete Shared Account (FR-27)｜No｜Deletion process is logical; system provides expected results and structure is sound.｜

｜Email Archiving Operations (FR-28–31)｜No｜Involves archive policy definition and application; entity chain is complete; feedback is clear.｜

｜View Archived Emails (FR-32)｜No｜Archived email ID corresponds to entity; system response follows logical flow.｜

｜Manage Archived Emails (FR-33)｜No｜Operational behavior matches entities; system responds reasonably.｜

｜Backup Data (FR-34)｜No｜Inputs and outputs are defined; system functions logically and coherently.｜

｜Restore Data (FR-35)｜No｜Path input aligns with data recovery action; entity structure is complete.｜

｜View Logs (FR-36,37,41)｜No｜Log filtering and display functions are clear; system data flow is verifiable.｜

｜Manage Archive Policies (FR-38)｜No｜Policy content and management behavior are clearly mapped; process is complete.｜

｜Delete Logs (FR-39,40,42)｜No｜Deletion operations are logically consistent; system behavior matches expectations.｜

｜Account Registration (FR-43)｜No｜User input and system response form a closed loop; functionality is complete.｜

｜Set Admin Permissions (FR-44)｜No｜Permission definitions are unclear; system handling lacks concrete standards.｜

｜Account Login (FR-45)｜No｜Credential input and session output form a valid flow; system behavior is stable.｜

Summary:

Number of data entities involved: 14 (User, Email, Folder, Event, Contact, DistributionList, ReminderTask, Mailbox, SharedAccount, ArchivedEmail, BackupLog, RestoreLog, ArchiveLog, ArchivePolicy).

Functional requirements count: Original 45; after merging similar functions, 33 remain.

Hallucinated requirements: 0 (All entities are connected to core system logic and form a weakly connected E-R diagram).

Non-hallucinated requirements: 33 (All verified as non-hallucinated).

Review Notes:

When merging functional requirements, strict adherence to the rules was maintained: only highly similar functional points were merged (e.g., multiple archiving operations), while CRUD operations remained distinct.

No hallucinated requirements exist: all entities are connected via User or operational links (e.g., User interacts with Email, Event, etc.); the E-R diagram remains weakly connected throughout.

All requirements have been evaluated with no omissions.