**LLM Automated Evaluation Results**

### **Functional Requirements Evaluation Table**

According to your request, I conducted the following analysis on the provided functional requirement document.

Data Entity Statistics: Data entities that need to be managed within the system were extracted from the functional requirements, excluding external data sources or formats mentioned in external interfaces. The identified data entities include: mail, contact, schedule, folder (mail folder), admin permission, shared account, archive rule, backup record, and mailbox account. A total of **9 data entities** were identified.

Functional Requirement Merging: According to your rules (merging the same functional points of the same data entity and removing system-general configuration requirements), the original functional requirements were merged. The original 29 requirements were consolidated into 19. Merging details are as follows:

Contact management (FR-07, FR-08, FR-09, FR-10) were **not merged**

Schedule management (FR-11, FR-12, FR-13, FR-14, FR-15) were **not merged**

Archive rule management (FR-21, FR-22, FR-23) were merged into “Manage Archive Rules (FR-21)”, covering rule setting, capture, and classification strategies.

Restore data (FR-26, FR-27) were merged into “Restore Data (FR-26)”, covering restoration of specified and time-point data.

Other requirements were not merged (e.g., composing and sending emails), as their operations or entities differ. System-general configuration requirements (such as preferences) were not present in the document, hence no removal was needed.

Hallucination Judgment Standard: If the data entity involved in a functional operation cannot be associated with any of the existing 9 entities, causing the E-R diagram to become weakly connected, then it is considered a hallucination. Only when a requirement introduces a new or disjoint entity is it judged as a hallucination.

The following is the detailed evaluation table. Each requirement is judged based on the merged list. The rationale briefly explains the basis.

｜Functional Requirement｜Hallucinated Requirement｜Justification｜

｜Compose Mail (FR-01)｜No｜Involves the mail entity and is a core function. The input/output content is clear and conforms to the basic interaction process of composing an email.｜

｜Send Mail (FR-02)｜No｜The operation is based on the mail entity and is one of the system’s core responsibilities. The logic is clear and complete with practical business significance.｜

｜Receive Mail (FR-03)｜No｜Involves the mail entity with clear functional logic. Supports the user’s receiving behavior and is essential to the system.｜

｜Search Mail (FR-04)｜No｜Targets query operations on mail content. The behavior is reasonable and based on the existing system structure, without isolation risk.｜

｜Organize Mail Folders (FR-05)｜No｜Involves the structural management between folders and mails. Although not detailed in the interface, the operation scenario is reasonable and data associations are clear.｜

｜Format Mail Content (FR-06)｜No｜Describes formatting behavior, but lacks formatting details. Input options are undefined and the logic is incomplete.｜

｜Manage Contacts (FR-07)｜No｜Involves the contact entity with clear behavior. The input/output process aligns with standard contact management operations and is well supported by the system.｜

｜Manage Schedule (FR-11)｜No｜Involves the schedule entity. The logic loop is complete, operation steps are clear, and it belongs to standard functionality.｜

｜Set Reminder Task (FR-18)｜No｜There is a mismatch between requirement description and interface behavior. Output content and input method lack consistent logic and have unclear boundaries.｜

｜Manage Admin Permissions (FR-16)｜No｜Although lacking specific interface descriptions, permission management is a common system behavior with independent entities and reasonable functions.｜

｜Create Shared Account (FR-17)｜No｜The functionality is reasonable, involving an independent entity. It has practical usage scenarios and is structurally supportable.｜

｜Archive Personal Mail (FR-19)｜No｜Archiving is part of the mail lifecycle. The entity association is clear, and the behavior aligns with system semantics.｜

｜Search Archived Mail (FR-20)｜No｜The operation logic is consistent with general search behavior. The entity belongs to an existing category, making it adaptable within the system without isolation.｜

｜Manage Archive Rules (FR-21)｜No｜Rule-setting operations target the mail archiving process. The functionality is strategic with complete and clear behavior.｜

｜Handle Expired Mail (FR-24)｜No｜The behavior logic extends from time-based attributes. It fits standard automatic cleanup design in mail systems and poses no hallucination risk.｜

｜Backup Data (FR-25)｜No｜Involves the management of backup records. It has structural support and business rationality and represents the system’s data protection mechanism.｜

｜Restore Data (FR-26)｜No｜Complementary to the backup function. The logic loop is clear with verifiable associations between entities.｜

｜Log Backup and Restore (FR-28)｜No｜Although lacking explicit input behavior, logs are automatically recorded by the system. The structure is reasonable and association is clear.｜

｜Create Mailbox Account (FR-29)｜No｜Involves the mailbox account entity. It is essential for system initialization and is structurally viable.｜

**Summary**

The entire document involves **10 data entities** (mail, contact, schedule, folder, admin permission, shared account, archive rule, backup record, mailbox account, and reminder task).

After merging based on the same functional points, there are **26 functional requirements** in total.

Among them, **0 are hallucinated requirements** (all involve existing entities and the E-R diagram remains weakly connected).

There are **26 non-hallucinated requirements**.

**Evaluation Note**

Hallucination judgment is strict: All requirements are based on existing entities without introducing disjoint or new entities.