LLM Automated Evaluation Results

Functional Requirement Evaluation Table

Below is a detailed evaluation based on your provided system overview and the external interface documentation. I first extracted functional requirements and data entities from the system overview and then applied your rules for merging and exclusion. The system overview describes the core functionality, while the external interface document provides interface definitions used to assess full referencing and testability.

Data Entity Extraction

From the system overview, I extracted data entities that are internally managed by the system (external sources and formats mentioned in the interface documentation are not counted). A data entity is defined as a core object being managed in the system:

• Email: involved in sending, receiving, formatting, searching, and organizing

• Calendar Event: involved in scheduling

• Contact: involved in management

• Distribution Group: involved in management (possibly associated with contacts)

• Task: involved in setting reminders

• Account: involved in managing mailbox and shared accounts

• Archived Email: involved in personal archive search, server archiving, email stream capture, policy-based archiving, and managing expired emails

• Policy: involved in policy-based archiving (defined as a standalone entity used for archive rules)

• Backup Record: involved in backup and recovery operations

• Log: involved in backup and restore log recording

Total number of data entities: 10 (Email, Calendar Event, Contact, Distribution Group, Task, Account, Archived Email, Policy, Backup Record, Log)

Functional Requirement Merging

Based on the system overview, I extracted the initial functional points and merged them according to your rules:

• Rule 1: Merge requirements targeting the same operation on the same data entity (e.g., admin creation of mailbox and shared account merged as “Account Management”)

• Rule 2: Eliminate general system settings (e.g., preferences, multilingual support); none were found in the overview

• Merge Note: No over-merging. For example, different operations on the Email entity (send, receive, search, etc.) are not merged since they are not the same functional point. CRUD operations are not merged unless operations and rules are close (e.g., account management)

Merged functional requirement list (18 items):

1. Send Email

2. Receive Email

3. Format Email

4. Search Email

5. Organize Email Folders

6. Calendar Scheduling

7. Manage Contacts

8. Manage Distribution Groups

9. Set Reminder Tasks

10. Manage Accounts (merged mailbox and shared account creation)

11. Personal Archive Search

12. Server Archiving

13. Email Stream Capture

14. Policy-Based Archiving

15. Manage Expired Emails

16. Backup Data

17. Restore Data

18. Record Backup Logs

The external interface document only describes interfaces (e.g., user interface, hardware interface) and does not introduce new functional requirements. Hardware interfaces (e.g., card reader, sensors) are considered external data sources and are not counted as internal entities or requirements.

Hallucination Identification

Evaluation criterion: a requirement is considered hallucinated if the data entity it introduces is not connected to any existing entity, breaking weak connectivity in the E-R diagram. General system settings are excluded from this assessment.

Result: All functional requirements involve the defined 10 entities and no isolated entity exists. The E-R diagram remains connected. Therefore, no hallucinated requirements exist.

Reason: Each requirement is connected to a core entity (e.g., Email, Archived Email). For example, Policy-Based Archiving involves the Policy entity, which links to Archived Email and does not break connectivity.

Complete Evaluation Table

All 18 functional requirements are evaluated individually. Evaluation is slightly strict to allow imperfections (such as incomplete referencing or untestability).

| Functional Requirement | Hallucination | Evaluation Justification |

| 1. Send Email | No | Involves Email entity, E-R connected; input/output clear, supported by external SMTP interface, functionality valid. |

| 2. Receive Email | No | Involves Email entity, E-R connected; definition clear, relies on external SMTP interface, functionally feasible. |

| 3. Format Email | No | Involves Email entity, E-R connected; formatting handled via UI, logical input/output. |

| 4. Search Email | No | Involves Email entity, E-R connected; query condition defined, output result clear and complete. |

| 5. Organize Email Folders | No | Involves Email entity and sub-properties, E-R connected; behavior definition valid, UI supports the action. |

| 6. Calendar Scheduling | No | Involves Calendar Event entity, E-R connected; scheduling supported structurally, input/output well defined. |

| 7. Manage Contacts | No | Involves Contact entity, E-R connected; management logic complete, UI interactions reasonable. |

| 8. Manage Distribution Groups | No | Involves Distribution Group entity, E-R connected; input/output logic clearly defined, system supports the action. |

| 9. Set Reminder Tasks | No | Involves Task entity, E-R connected; reminders supported via mail notifications, logic valid. |

| 10. Manage Accounts | No | Involves Account entity, E-R connected; account creation/modification logic clear, UI support complete. |

| 11. Personal Archive Search | No | Involves Archived Email entity, E-R connected; input/output path complete, function conventional. |

| 12. Server Archiving | No | Involves Archived Email entity, E-R connected; archiving process is conventional, functional design reasonable. |

| 13. Email Stream Capture | No | Involves Archived Email entity, E-R connected; lacks defined external interface and I/O structure, logic incomplete. |

| 14. Policy-Based Archiving | No | Involves Archived Email and Policy entities, E-R connected; policy input undefined, behavior transformation unclear. |

| 15. Manage Expired Emails | No | Involves Archived Email entity, E-R connected; conditions and actions not explicitly defined, input-output conversion missing. |

| 16. Backup Data | No | Involves Backup Record entity, E-R connected; I/O defined, structural logic valid. |

| 17. Restore Data | No | Involves Backup Record entity, E-R connected; recovery flow clearly defined, system supports execution. |

| 18. Record Backup Logs | No | Involves Log entity, E-R connected; log I/O defined, functionality structurally complete. |

Summary

Total number of data entities involved in the document: 10 (Email, Calendar Event, Contact, Distribution Group, Task, Account, Archived Email, Policy, Backup Record, Log)

Total functional requirements after merging: 18

Hallucinated requirements: 0 (all requirements involve existing entities, E-R diagram is connected)

Non-hallucinated requirements: 18

Note: Evaluation is based on functional requirements from the system overview and the external interface document. External sources mentioned in hardware interfaces are not counted to ensure the focus remains on internal system logic. The assessment is slightly strict, but overall requirement quality is high, with only a few requirements marked incomplete or untestable due to ambiguity in definitions.