LLM Automated Evaluation Results

**Functional Requirement Evaluation Table**

As requested, I have conducted a systematic analysis of the entire requirement document. The analysis strictly followed the specified rules: functional requirements were only merged when they concerned the same data entity and functional point (avoiding over-merging); general system settings were excluded (none were present in this document); hallucinated requirements were assessed based on data entity connectivity (E-R diagram weak connectivity). All evaluations were conducted with slightly strict criteria to ensure that no requirement was overlooked.

**Analysis Basis:**

* **Data Entity Identification**: Core data entities managed internally by the system were extracted from the functional requirements (excluding external interfaces such as APIs or data sources). A total of 11 data entities were identified:
  + Email: Covers general emails, archived emails, expired emails, etc.
  + Folder: Email folder structure.
  + Contact: Contact information.
  + Group: Distribution list/group information.
  + Event: Calendar events and reminders.
  + SystemConfiguration: Server archiving settings and mail flow capture rules.
  + UserAccount: Email accounts and shared accounts.
  + Backup: File-level and full-database backups.
  + Restore: Various recovery operations.
  + Log: Backup and restore logs.
  + Audit: Audit logs for mail archiving and backup/restore operations.
* **Functional Requirement Merging**:
  + Merging Rule: Merge only requirements that operate on the same data entity with the same functional point (e.g., searching mail and searching personal archived mail were merged into “Mail Search”). No general system setting requirements were present, so none were excluded.
  + Original requirement count: 38; after merging, reduced to 31.
  + Merge Details:
    - FR-01 and FR-02 merged as “Email Creation (Send and Receive)”
    - FR-04 and FR-21 merged as “Mail Search”
    - FR-20 and FR-24 merged as “Mail Archiving”
    - FR-30 and FR-31 merged as “Backup Operation”
    - FR-32, FR-33, and FR-34 merged as “Restore Operation”
    - FR-35 and FR-36 merged as “Log Recording”
    - All other requirements were kept independent due to differing functionality or data entities
  + Final list of 31 functional requirements:
    - Email Creation (Send and Receive)
    - Format Email
    - Mail Search
    - Move Email
    - Delete Email
    - Manage Email Folders
    - Add Contact
    - Edit Contact
    - Search Contact
    - Delete Contact
    - Create Group
    - Edit Group
    - Search Group
    - Delete Group
    - Set Event Reminder
    - Edit Event
    - View Event
    - Delete Event
    - Mail Archiving
    - Handle Expired Mail
    - Server Archiving Configuration
    - Mail Flow Capture Rule Setting
    - Admin Create Mailbox
    - Admin Delete Mailbox
    - Admin Set Shared Account
    - Admin Revoke Shared Account
    - Backup Operation
    - Restore Operation
    - Log Recording
    - Audit Mail Archiving
    - Audit Backup and Restore

**Evaluation Results:**

All 31 functional requirements were evaluated individually. Hallucinated requirements were assessed based on data entity connectivity (E-R diagram weak connectivity). Each evaluation includes a brief justification.

| Functional Requirement | Hallucination | Justification |

|------------------------|---------------|---------------|

Email Creation (Send and Receive) | No | Involves the Email entity; input/output clearly defined; sending and receiving flow aligns with system logic.  
Format Email | No | Email formatting is based on the existing entity; interactions are clear; triggering and display are reasonable.  
Mail Search | No | Email entity supports condition-based retrieval; logic aligns with interface operations.  
Move Email | No | Relationship between Email and Folder entities is clear; move operation logic is complete.  
Delete Email | No | Delete operation is based on Email entity; process is self-contained with no redundancy.  
Manage Email Folders | No | Folder entity exists; operations are practical and valid.  
Add Contact | No | Contact entity is well-defined; input and save logic is reasonable.  
Edit Contact | No | Contact information update logic is sound and aligns with user actions.  
Search Contact | No | Search behavior based on contact attributes is clearly defined; function path is valid.  
Delete Contact | No | Contact deletion is complete and consistent with the interface and entity.  
Create Group | No | Group entity exists; creation operation aligns with user interaction logic.  
Edit Group | No | Group information is editable; entity aligns with user action flow.  
Search Group | No | Search logic corresponds to group structure; query path is clear.  
Delete Group | No | Group deletion is clearly defined; functional goal is specific.  
Set Event Reminder | No | Event entity exists; setting process has a complete structure.  
Edit Event | No | Event content is updatable; function flow aligns with system interfaces.  
View Event | No | Viewing behavior is based on existing event data; display path is well-defined.  
Delete Event | No | Deletion logic is self-contained; entity path is valid.  
Mail Archiving | No | Mail archiving is based on classification; logic is valid and feasible.  
Handle Expired Mail | No | Expired mail is a state of the Email entity; handling logic is appropriate.  
Server Archiving Configuration | No | SystemConfiguration entity supports archiving strategy settings; logic is valid.  
Mail Flow Capture Rule Setting | No | Capture rules are linked with SystemConfiguration; process is clear.  
Admin Create Mailbox | No | UserAccount entity supports mailbox creation; function is valid.  
Admin Delete Mailbox | No | Mailbox account can be deleted; path and logic align with admin operations.  
Admin Set Shared Account | No | Shared account setting is clearly associated with the UserAccount entity; operation is complete.  
Admin Revoke Shared Account | No | Revoke logic matches entity structure; process is clear.  
Backup Operation | No | Backup entity exists; output location depends on undefined interface (incomplete reference), but function is valid.  
Restore Operation | No | Restore entity exists; target destination is not detailed (incomplete reference), but process is complete.  
Log Recording | No | Log entity exists; task recording logic is valid.  
Audit Mail Archiving | No | Audit entity is associated with the Email entity; operation path is clear.  
Audit Backup and Restore | No | Audit process spans multiple entities; input/output paths are consistent; function is valid.

**Summary:**

* The document involves **11 data entities** (Email, Folder, Contact, Group, Event, SystemConfiguration, UserAccount, Backup, Restore, Log, Audit).
* After merging functional requirements with identical functional points, there are **31 functional requirements**.
* **Number of Hallucinated Requirements: 0** (All requirements involve defined data entities; E-R diagram remains weakly connected with no isolated entities).
* **Number of Non-Hallucinated Requirements: 31** (No hallucinated requirements identified, so all are valid).

**Note**: Evaluation applied slightly strict criteria, but no hallucinated requirements were found. All other requirements meet the standards.