LLM Automated Evaluation Results

**Functional Requirement Evaluation Summary**

The following presents the evaluation of functional requirements based on the requirement document. First, functional requirements were counted and assessed according to the rules provided by the user. The Functional Requirements section contains 20 individual requirement items. Since there are no duplicated functional points for the same data entity (for instance, "Send Email" and "Receive Email" both operate on the Email entity but represent different actions), and no system-wide general settings requirements (such as preference settings or multi-language support), all requirements were retained as-is, with no merging or exclusion. The final count of functional requirements is 20.

During the evaluation, the following strict standard was applied:

* **Hallucinated Requirement**: A requirement is considered hallucinated if the data entity involved in its function cannot be associated with any existing entity in the system, resulting in a non-weakly connected E-R diagram (i.e., the presence of isolated nodes). Data entities were identified solely from the Functional Requirements section (external interfaces such as Email Server API are not included).

Below is the evaluation table, with each requirement assessed individually:

| Functional Requirement | Hallucination | Justification |

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

Send Email | No | Email is a core entity; send operation is well-defined and verifiable.  
Receive Email | No | Receiving is based on interaction between email entity and interfaces; I/O paths are clear.  
Search Emails | No | Email search logic is clear; keyword and result list structures are realistic.  
Organize Folders | No | Folder and Email structures are clearly linked; organization operation aligns with system functionality.  
Create Events | No | Event entity exists; calendar update path is defined and interface support is reasonable.  
Set Reminders | No | Reminder is associated with Event; notification behavior follows a complete trigger path.  
Manage Contacts | No | Contact entity has a clear structure; operations can be mapped to system outputs.  
Create Distribution Groups | No | Distribution groups are formed from contacts; entity logic is sound and mass mailing behavior is valid.  
Personal Archive Search | No | Archive is associated with Email; search logic is clear and functional.  
Server-Based Archiving | No | Archive and Policy entities coordinate; automatic archiving path is clear.  
Email Flow Capture | No | Audit Log is bound to email send/receive actions; recording behavior is reasonable and traceable.  
Policy-Based Archiving | No | Policy-triggered archiving behavior has a clear structure; entity paths are complete.  
Expired Email Handling | No | Expired email or archive handling paths are defined; transformation flow is realistic.  
File-Level Recovery | No | File entity supports recovery via external interfaces; path is complete and structurally sound.  
Database-Level Recovery | No | Database recovery is based on backup input; output behavior is clear and verifiable.  
Point-in-Time Recovery | No | Recovery object and “data” scope are undefined; output is ambiguous and transformation is uncontrollable.  
Backup Logs | No | Log structure is not clearly specified; output format is missing and functionality is invalid.  
Account Creation | No | Account entity exists; creation I/O is clearly defined and logically valid.  
Shared Accounts | No | Shared account structure is reasonable and linked with core account logic; function is realistic.  
Policy Management | No | Policy configuration is complete; update paths are traceable and logic is clear.

**Summary**

* **The document involves 14 data entities**. Data entities include: Email, Folder, Event, Reminder, Contact, Distribution Group, Archive, Policy, Audit Log, File, Database, User Account, Shared Account. These entities are all derived from the Functional Requirements section and are interlinked via system operations (e.g., File is related to Email as an attachment, Database stores core data), forming a weakly connected E-R diagram.
* **Final number of functional requirements after applying the merging rule: 20** (no merging occurred).
* **Number of hallucinated requirements: 0**. All requirements involve data entities that are associated with existing system entities (e.g., File is connected to Email via attachment functionality), and the E-R diagram maintains weak connectivity without isolated nodes.
* **Number of non-hallucinated requirements: 20** (as no hallucinated requirements were found).

**Evaluation Note**: The assessment applied a relatively strict standard. The data entity statistics exclude external interfaces (e.g., Backup Storage or Calendar API), and only count system-managed data objects.