项目文档

# Functional Requirement

# 1. Functional Requirements   
  
## 1.1 Email Creation Function   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: Administrators can create a new email with specified content and recipients. The email is associated with a specific email account and optionally scheduled via calendar.   
\*\*Input\*\*: Email subject, body, recipient (Contact or Distribution Group), sender (Email Account), optional schedule date/time.   
\*\*Output\*\*: A new Email entity is stored in the server storage and optionally added to a CalendarEvent.   
  
---  
  
## 1.2 Email Sending Function   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: Administrators can send an email using a specified email account. Recipients must be valid Contacts or members of a Distribution Group.   
\*\*Input\*\*: Email content, sender email account, recipient list (Contact or Distribution Group), optional schedule date/time.   
\*\*Output\*\*: The Email is sent and stored in the server storage, and optionally added to the calendar.   
  
---  
  
## 1.3 Email Receiving Function   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: The system automatically checks for new emails in configured email accounts and stores them in server storage.   
\*\*Input\*\*: Email account configuration and server connection details.   
\*\*Output\*\*: New Email entity is stored in server storage and optionally archived or logged.   
  
---  
  
## 1.4 Email Viewing Function   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: Administrators can view the details of an existing email from the server storage.   
\*\*Input\*\*: Selected EmailID and Email Account information.   
\*\*Output\*\*: Email details (Subject, Body, Sender, Recipient, Attachments, Send/Receive date) displayed in the user interface.   
  
---  
  
## 1.5 Email Deletion Function   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: Administrators can delete an email from the server storage.   
\*\*Input\*\*: Selected EmailID and confirmation to delete.   
\*\*Output\*\*: Email is removed from server storage and the email list is updated.   
  
---  
  
## 1.6 Email Archiving Function   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: Administrators can move an email from server storage to an archive based on archive policies.   
\*\*Input\*\*: Selected EmailID, confirmation to archive, and policy details.   
\*\*Output\*\*: Email is archived, and the server storage and email list are updated.   
  
---  
  
## 1.7 Archived Email Search Function   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: Administrators can search for archived emails using filters like subject, sender, date, or keywords.   
\*\*Input\*\*: Search criteria (e.g., subject, sender, date range).   
\*\*Output\*\*: List of matching archived Email entities displayed in the user interface.   
  
---  
  
## 1.8 Contact Management Function   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: Administrators can add, edit, or delete contact information and associate it with a user.   
\*\*Input\*\*: Contact details (Name, Email, Phone, UserID).   
\*\*Output\*\*: Updated Contact list is stored in the system and available for use in email sending.   
  
---  
  
## 1.9 Distribution Group Management Function   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: Administrators can create, update, or delete a distribution group and manage its members (Contacts or Email Accounts).   
\*\*Input\*\*: Group name, selected contacts or email accounts.   
\*\*Output\*\*: Updated DistributionGroup entity is stored in the system and available for use in email sending.   
  
---  
  
## 1.10 Calendar Management Function   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: Administrators can manage calendar events, including creation, updating, and deletion. Events may be associated with scheduled emails.   
\*\*Input\*\*: Event details (Title, Description, Start and End Time, associated EmailID).   
\*\*Output\*\*: Updated CalendarEvent is stored in the system and reflected in the calendar view.   
  
---  
  
## 1.11 Calendar Event Participant Management Function   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: Administrators can add, remove, or update participants of a calendar event.   
\*\*Input\*\*: Selected CalendarEventID, list of participants (UserIDs or EmailAccountIDs), and participant roles.   
\*\*Output\*\*: Updated CalendarEventParticipant list is stored in the system and reflected in the event details.   
  
---  
  
## 1.12 Email Account Management Function   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: Administrators can create, update, or delete email accounts and manage their configurations.   
\*\*Input\*\*: Email account details (Username, Password, Creation/LastModified/LastDeleted Dates).   
\*\*Output\*\*: Updated EmailAccount is stored in the system and associated with emails and server storage.   
  
---  
  
## 1.13 Server Storage Management Function   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: Administrators can configure, expand, or modify server storage settings.   
\*\*Input\*\*: Server name, capacity settings, backup and retention policies.   
\*\*Output\*\*: Updated ServerStorage entity is stored in the system, and changes are applied to the storage.   
  
---  
  
## 1.14 Archiving Policy Management Function   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: Administrators can create, update, or delete archiving policies to define how and when emails are archived.   
\*\*Input\*\*: Policy name, description, archive criteria, and expiry rules.   
\*\*Output\*\*: Updated Policy is stored in the system and applied to eligible emails.   
  
---  
  
## 1.15 Expired Email Management Function   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: Administrators can manage expired emails by either recovering or permanently deleting them.   
\*\*Input\*\*: ExpiredEmailID, action (Recover or Delete).   
\*\*Output\*\*: ExpiredEmail is either restored to server storage or removed, and the system logs the activity.   
  
---  
  
## 1.16 Expired Email Recovery Function   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: Administrators can recover an expired email from its archive or backup location using the recovery policy.   
\*\*Input\*\*: ExpiredEmailID, confirmation to recover.   
\*\*Output\*\*: ExpiredEmail is restored to server storage and associated with the original EmailAccount.   
  
---  
  
## 1.17 Email Flow Capture Function   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: The system captures and logs the flow of emails based on predefined policies.   
\*\*Input\*\*: Email Account ID, capture criteria (source, destination, conditions), and policy configuration.   
\*\*Output\*\*: EmailFlow entity is created and stored in the system.   
  
---  
  
## 1.18 Email Record Audit Function   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: Administrators can view and audit logs of email actions (e.g., send, archive, delete) for compliance purposes.   
\*\*Input\*\*: EmailRecordID, filter criteria (e.g., action type, date, user).   
\*\*Output\*\*: EmailRecord details (EmailID, Action, Timestamp, UserID) displayed in the system.   
  
---  
  
## 1.19 Email Folder Management Function   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: Administrators can create, rename, delete, or move emails between folders.   
\*\*Input\*\*: Folder name, description, and list of emails to be moved.   
\*\*Output\*\*: Updated EmailFolder and Email associations stored in the system.   
  
---  
  
## 1.20 Email Attachment Management Function   
\*\*Function ID\*\*: FR-20   
\*\*Description\*\*: Administrators can upload, view, rename, or remove attachments from emails.   
\*\*Input\*\*: EmailID, attachment file (filename, size, upload date), action (Add, Remove, Rename).   
\*\*Output\*\*: Updated EmailAttachment and Email entities stored in the system.   
  
---  
  
## 1.21 Administrator Permission Management Function   
\*\*Function ID\*\*: FR-21   
\*\*Description\*\*: Administrators can configure and manage permission levels of other administrator accounts.   
\*\*Input\*\*: AdminID, permission settings (e.g., send email, manage policies, view archive).   
\*\*Output\*\*: Updated Admin entity and associated permissions stored in the system.   
  
---  
  
## 1.22 Administrator Account Update Function   
\*\*Function ID\*\*: FR-22   
\*\*Description\*\*: Administrators can modify their own or other admin accounts' settings, including password and access level.   
\*\*Input\*\*: AdminID, updated settings (e.g., password, notification preferences).   
\*\*Output\*\*: Updated Admin entity stored in the system.   
  
---  
  
## 1.23 Administrator Account Deletion Function   
\*\*Function ID\*\*: FR-23   
\*\*Description\*\*: Administrators can delete an administrator account, and the system will process all related data (e.g., emails, policies, calendar events).   
\*\*Input\*\*: AdminID, confirmation to delete.   
\*\*Output\*\*: Admin account is removed, and associated data is either deleted or transferred.   
  
---  
  
## 1.24 System Logging and Activity Tracking Function   
\*\*Function ID\*\*: FR-24   
\*\*Description\*\*: The system logs all user actions and system events for audit and traceability.   
\*\*Input\*\*: Action type, EmailID, UserID, Timestamp.   
\*\*Output\*\*: EmailRecord or system log entry stored in the server storage.   
  
---  
  
These functional requirements are derived directly from the provided use cases and data model. Each function ensures that the system can be used to manage email communication, contact and distribution group handling, calendar scheduling, and long-term email archiving and recovery in a structured and auditable manner.

# External Description

# Software Requirements Specification (SRS)   
## Chapter 2: External Interfaces   
  
This chapter describes the external interfaces that the system will interact with. It includes the definitions, descriptions, and interaction methods for \*\*User Interfaces\*\*, \*\*Hardware Interfaces\*\*, \*\*Software Interfaces\*\*, and \*\*Communication Interfaces\*\*. These interfaces are identified based on the functional requirements provided and are essential for the system's integration with external components.  
  
---  
  
## 2.1 User Interface   
  
The system interacts with administrators through a \*\*graphical user interface (GUI)\*\*, which supports the full range of operations related to email, contacts, distribution groups, calendar events, and system settings. The interface is designed to be intuitive and role-based, with access to functions determined by the administrator’s permissions.   
  
### 2.1.1 Email Management UI   
- \*\*Description\*\*: A UI module for creating, viewing, deleting, archiving, and managing folders and attachments for emails.   
- \*\*Interaction Method\*\*:   
 - Input: Email subject, body, sender (Email Account), recipient (Contact or Distribution Group), optional schedule date/time, folder name, attachment file, action (e.g., send, archive, delete).   
 - Output: Display of Email details (Subject, Body, Sender, Recipient, Attachments, Send/Receive date), updated email list, folder structure, and confirmation messages.   
  
### 2.1.2 Contact and Distribution Group Management UI   
- \*\*Description\*\*: A UI module for managing Contacts and Distribution Groups, including adding, editing, and deleting entries.   
- \*\*Interaction Method\*\*:   
 - Input: Contact details (Name, Email, Phone, UserID), Distribution Group name and members (Contacts or Email Accounts).   
 - Output: Updated Contact list and Distribution Group list displayed in the UI.   
  
### 2.1.3 Calendar and Event Management UI   
- \*\*Description\*\*: A UI module for creating, updating, and viewing Calendar Events, and managing participants.   
- \*\*Interaction Method\*\*:   
 - Input: Event details (Title, Description, Start and End Time), participant list (UserIDs or EmailAccountIDs), participant roles.   
 - Output: Display of CalendarEvent details, updated event list, and participant list.   
  
### 2.1.4 Archiving and Policy Management UI   
- \*\*Description\*\*: A UI module for managing archiving policies and recovering expired emails.   
- \*\*Interaction Method\*\*:   
 - Input: Policy name, description, archive criteria, expiry rules, ExpiredEmailID, action (Recover or Delete).   
 - Output: Updated Policy list, archived email list, and confirmation of recovery or deletion.   
  
### 2.1.5 Administrator and System Settings UI   
- \*\*Description\*\*: A UI module for managing administrator accounts and permissions, system logging, and server storage settings.   
- \*\*Interaction Method\*\*:   
 - Input: AdminID, password, permission settings, server name, capacity settings, backup and retention policies.   
 - Output: Updated Admin entity, updated ServerStorage entity, and confirmation of account changes.   
  
### 2.1.6 Email Audit and Logging UI   
- \*\*Description\*\*: A UI module for viewing logs of email actions (e.g., send, archive, delete) for compliance and traceability.   
- \*\*Interaction Method\*\*:   
 - Input: EmailRecordID, filter criteria (e.g., action type, date, user).   
 - Output: Display of EmailRecord details (EmailID, Action, Timestamp, UserID) for audit purposes.   
  
---  
  
## 2.2 Hardware Interface   
  
This system does not have direct dependencies or interactions with hardware devices. It is primarily a software-based system hosted on standard server infrastructure and accessed via networked devices (e.g., desktops, laptops, or mobile devices). However, it may indirectly interact with hardware through the following ways:  
  
### 2.2.1 Server Storage Devices   
- \*\*Description\*\*: The system stores and retrieves email data, calendar events, and logs from server storage, which may include physical or virtual storage devices such as hard drives, SSDs, or cloud storage.   
- \*\*Interaction Method\*\*:   
 - Input: Server name, capacity settings, backup and retention policies.   
 - Output: Updated ServerStorage entity reflecting the new settings, and persistent storage of Email, CalendarEvent, and EmailRecord entities.   
- \*\*Notes\*\*: The hardware interface is abstracted via the software interface for ServerStorage, and developers should not need to interact directly with the hardware.   
  
---  
  
## 2.3 Software Interface   
  
The system interacts with several software components, including databases, external APIs, and internal data models. These interfaces are crucial for data persistence, data exchange, and system integration.  
  
### 2.3.1 Database Interface   
- \*\*Description\*\*: The system uses a database to persist and retrieve data such as Email, Contact, DistributionGroup, CalendarEvent, EmailAccount, ServerStorage, Policy, EmailFolder, EmailAttachment, Admin, and EmailRecord entities.   
- \*\*Interaction Method\*\*:   
 - Input: Entities (e.g., Email, CalendarEvent, EmailAccount) and their attributes (e.g., subject, body, sender, recipient, event time, policy criteria).   
 - Output: Persisted or updated entities, query results based on search criteria (e.g., archived emails, logs, contacts).   
- \*\*Notes\*\*: The database interface supports SQL queries or ORM-based operations depending on the implementation.   
  
### 2.3.2 Email Server API   
- \*\*Description\*\*: The system communicates with external email servers (e.g., SMTP, IMAP, POP3) to send, receive, and manage emails.   
- \*\*Interaction Method\*\*:   
 - Input: Email content, sender email account, recipient list (Contact or Distribution Group), optional schedule date/time.   
 - Output: Email sent to recipients, new Email entity stored in the system, or confirmation of successful receipt.   
  
### 2.3.3 Calendar System API   
- \*\*Description\*\*: The system may interface with a calendar system (e.g., Google Calendar, Microsoft Outlook, or a custom calendar) to create and manage CalendarEvents associated with scheduled emails.   
- \*\*Interaction Method\*\*:   
 - Input: Event details (Title, Description, Start and End Time, associated EmailID), participant list (UserIDs or EmailAccountIDs), and participant roles.   
 - Output: Updated CalendarEvent in the external system, including participants and event details.   
  
### 2.3.4 File Storage API   
- \*\*Description\*\*: The system uses a file storage API (e.g., local file system or cloud storage) to manage attachments associated with emails.   
- \*\*Interaction Method\*\*:   
 - Input: EmailID, attachment file (filename, size, upload date), action (Add, Remove, Rename).   
 - Output: Updated EmailAttachment entity and associated Email entity in the database.   
  
---  
  
## 2.4 Communication Interface   
  
The system communicates with external systems and users through network-based protocols and messaging systems to support sending emails, receiving emails, and logging activities.  
  
### 2.4.1 Email Communication Interface   
- \*\*Description\*\*: The system sends and receives emails via standard email protocols (e.g., SMTP for sending, IMAP/POP3 for receiving).   
- \*\*Interaction Method\*\*:   
 - Input: Email content, sender email account, recipient list (Contact or Distribution Group), optional schedule date/time.   
 - Output: Email sent to recipients or received from external email servers and stored in the system.   
  
### 2.4.2 Web Communication Interface   
- \*\*Description\*\*: The system communicates with users and administrators via a web-based interface using HTTP/HTTPS protocols.   
- \*\*Interaction Method\*\*:   
 - Input: User input via web forms or API calls (e.g., creating an email, updating a policy).   
 - Output: JSON/XML responses or HTML rendering of data for display in the UI.   
  
### 2.4.3 System Logging and Audit Communication   
- \*\*Description\*\*: The system logs all user actions and system events for audit and traceability purposes. These logs may be stored in a centralized logging system or database.   
- \*\*Interaction Method\*\*:   
 - Input: Action type, EmailID, UserID, Timestamp.   
 - Output: EmailRecord or system log entry stored in the server storage.   
  
---  
  
## Cross-Check Summary   
  
| Functional Requirement | External Interface(s) Involved |  
|------------------------|-------------------------------|  
| FR-01 (Email Creation) | User Interface, Database Interface, Email Server API |  
| FR-02 (Email Sending) | User Interface, Email Server API, Calendar System API, Database Interface |  
| FR-03 (Email Receiving) | Email Server API, Database Interface |  
| FR-04 (Email Viewing) | User Interface, Database Interface |  
| FR-05 (Email Deletion) | User Interface, Database Interface |  
| FR-06 (Email Archiving) | User Interface, Database Interface |  
| FR-07 (Archived Email Search) | User Interface, Database Interface |  
| FR-08 (Contact Management) | User Interface, Database Interface |  
| FR-09 (Distribution Group Management) | User Interface, Database Interface |  
| FR-10 (Calendar Management) | User Interface, Calendar System API, Database Interface |  
| FR-11 (Calendar Event Participant Management) | User Interface, Calendar System API, Database Interface |  
| FR-12 (Email Account Management) | User Interface, Email Server API, Database Interface |  
| FR-13 (Server Storage Management) | User Interface, Database Interface, Server Storage Devices |  
| FR-14 (Archiving Policy Management) | User Interface, Database Interface |  
| FR-15 (Expired Email Management) | User Interface, Database Interface |  
| FR-16 (Expired Email Recovery) | User Interface, Database Interface, File Storage API |  
| FR-17 (Email Flow Capture) | Email Server API, Database Interface |  
| FR-18 (Email Record Audit) | User Interface, Database Interface |  
| FR-19 (Email Folder Management) | User Interface, Database Interface |  
| FR-20 (Email Attachment Management) | User Interface, File Storage API, Database Interface |  
| FR-21 (Admin Permission Management) | User Interface, Database Interface |  
| FR-22 (Admin Account Update) | User Interface, Database Interface |  
| FR-23 (Admin Account Deletion) | User Interface, Database Interface |  
| FR-24 (System Logging and Activity Tracking) | User Interface, Database Interface, System Logging Interface |  
  
---  
  
## Conclusion   
  
The external interfaces defined in this chapter are critical for the system to function as intended. They ensure seamless interaction with users, external APIs, databases, and file systems. Developers should ensure that these interfaces are implemented in a consistent and secure manner, respecting the input/output formats and interaction methods described.

# Use Case

Use Case Name: Create Email   
Use Case ID: UC-01   
Actors: Administrator, Email Account   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to create an email.   
3. An email account exists in the system.   
  
Postconditions:   
1. A new email is created and stored in the system.   
2. The email is associated with the specified email account.   
3. The email is added to the relevant calendar if scheduled.   
  
Main Flow:   
1. The administrator navigates to the "Create Email" feature.   
2. The system displays a form for composing a new email.   
3. The administrator fills in the email subject, body, and recipient information.   
4. The administrator selects an email account to send the email from.   
5. If the email is to be scheduled, the administrator selects a date and time from the calendar.   
6. The administrator clicks "Send" or "Schedule".   
7. The system validates the form data.   
8. The system creates the email and stores it in the server storage.   
9. If scheduled, the email is added to the calendar with the specified time.   
10. The system confirms the email creation and provides a success message.   
  
Alternative Flow:   
1. If the form data is invalid, the system displays an error message and prompts the administrator to correct it.   
2. If the selected email account is not valid or not configured, the system displays an error message and cancels the email creation.   
3. If the administrator cancels the operation before sending, the system discards the email draft and returns to the previous screen.   
4. If there is an issue with the server storage, the system logs the error and notifies the administrator.  
  
Use Case Name: Send Email   
Use Case ID: UC-02   
Actors: Administrator, Email Account, Contact, Distribution Group   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to send an email.   
3. A valid email account exists in the system.   
4. The recipient information is valid (contact or distribution group).   
  
Postconditions:   
1. The email is sent from the selected email account.   
2. The email is stored in the server storage.   
3. If the email is scheduled, it is added to the calendar with the specified time.   
4. The system confirms the successful sending of the email.   
  
Main Flow:   
1. The administrator navigates to the "Send Email" feature.   
2. The system displays a form for composing and sending an email.   
3. The administrator fills in the email subject, body, and selects recipients (contacts or distribution groups).   
4. The administrator selects an email account to send the email from.   
5. If the email is to be scheduled, the administrator selects a date and time from the calendar.   
6. The administrator clicks "Send" or "Schedule".   
7. The system validates the form data and recipient information.   
8. The system sends the email using the selected email account.   
9. The system stores the email in the server storage.   
10. If scheduled, the email is added to the calendar with the specified time.   
11. The system confirms the successful sending of the email and provides a success message.   
  
Alternative Flow:   
1. If the form data is invalid, the system displays an error message and prompts the administrator to correct it.   
2. If the selected email account is not valid or not configured, the system displays an error message and cancels the sending.   
3. If the selected recipients are not valid or do not exist, the system displays an error message and prompts the administrator to correct the recipient information.   
4. If the administrator cancels the operation before sending, the system discards the email draft and returns to the previous screen.   
5. If there is an issue with the email sending process (e.g., network failure), the system logs the error and notifies the administrator.   
6. If there is an issue with the server storage, the system logs the error and notifies the administrator.  
  
Use Case Name: Receive Email   
Use Case ID: UC-03   
Actors: Administrator, Email Account, Server Storage   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to receive emails.   
3. An email account is configured and active in the system.   
4. The server storage is operational and accessible.   
  
Postconditions:   
1. The received email is stored in the server storage.   
2. The email is associated with the correct email account.   
3. The system notifies the administrator of the new email.   
4. If applicable, the email is archived or processed according to the recovery policy.   
  
Main Flow:   
1. The system checks for new emails in the configured email account.   
2. If new emails are found, the system downloads and stores them in the server storage.   
3. The system associates the received email with the corresponding email account.   
4. The system updates the administrator's interface to reflect the new email.   
5. The administrator views the received email in the inbox.   
6. The system optionally archives the email based on the archive policy.   
7. The system logs the successful receipt of the email.   
  
Alternative Flow:   
1. If no new emails are found, the system logs and displays a message indicating no new emails.   
2. If the email account is not configured or not active, the system displays an error and does not attempt to receive emails.   
3. If the server storage is unavailable, the system logs the error and notifies the administrator.   
4. If an email fails to download or is corrupted, the system logs the error and attempts to recover the email using the recovery policy.   
5. If the email is marked as expired, the system processes it according to the expired email policy and does not notify the administrator.  
  
Use Case Name: View Email   
Use Case ID: UC-04   
Actors: Administrator, Email Account, Server Storage   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to view emails.   
3. An email exists in the server storage for the selected email account.   
  
Postconditions:   
1. The email is displayed to the administrator in a readable format.   
2. The system logs the viewing activity.   
3. The administrator may choose to take further actions (e.g., reply, archive, delete).   
  
Main Flow:   
1. The administrator navigates to the "Inbox" or "Email List" section.   
2. The system retrieves and displays a list of emails stored in the server storage for the selected email account.   
3. The administrator selects an email from the list.   
4. The system retrieves the selected email from the server storage.   
5. The system displays the email details (subject, sender, body, attachments, if any).   
6. The system logs the viewing of the email.   
  
Alternative Flow:   
1. If no emails are found for the selected email account, the system displays a message indicating the inbox is empty.   
2. If the selected email does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the server storage is unavailable, the system logs the error and notifies the administrator.   
4. If the administrator closes the view without taking further action, the system returns to the email list.  
  
Use Case Name: Delete Email   
Use Case ID: UC-05   
Actors: Administrator, Email Account, Server Storage   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to delete emails.   
3. An email exists in the server storage for the selected email account.   
  
Postconditions:   
1. The selected email is removed from the server storage.   
2. The email list for the administrator is updated to reflect the deletion.   
3. The system logs the deletion activity.   
  
Main Flow:   
1. The administrator navigates to the "Inbox" or "Email List" section.   
2. The system retrieves and displays a list of emails stored in the server storage for the selected email account.   
3. The administrator selects the email to be deleted.   
4. The administrator clicks the "Delete" button.   
5. The system confirms the deletion request with the administrator.   
6. The administrator confirms the deletion.   
7. The system removes the selected email from the server storage.   
8. The system updates the email list to reflect the deletion.   
9. The system logs the deletion activity.   
10. The system provides a success message indicating the email was deleted.   
  
Alternative Flow:   
1. If the administrator cancels the deletion confirmation, the system retains the email and returns to the email list.   
2. If the selected email does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the server storage is unavailable, the system logs the error and notifies the administrator.   
4. If the administrator does not have permission to delete the email, the system displays an error message and prevents the deletion.   
5. If the system fails to delete the email for any reason, it logs the error and notifies the administrator.  
  
Use Case Name: Archive Email   
Use Case ID: UC-06   
Actors: Administrator, Email Account, Server Storage, Archive   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to archive emails.   
3. An email exists in the server storage for the selected email account.   
4. The archive feature is configured and operational.   
  
Postconditions:   
1. The selected email is moved from the server storage to the archive.   
2. The email list for the administrator is updated to reflect the archiving.   
3. The system logs the archiving activity.   
  
Main Flow:   
1. The administrator navigates to the "Inbox" or "Email List" section.   
2. The system retrieves and displays a list of emails stored in the server storage.   
3. The administrator selects an email to be archived.   
4. The administrator clicks the "Archive" button.   
5. The system confirms the archiving request with the administrator.   
6. The administrator confirms the archiving.   
7. The system moves the selected email from the server storage to the archive.   
8. The system updates the email list to reflect the removal of the email.   
9. The system logs the archiving activity.   
10. The system provides a success message indicating the email was archived.   
  
Alternative Flow:   
1. If the administrator cancels the archiving confirmation, the system retains the email in the inbox and returns to the email list.   
2. If the selected email does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the archive is not accessible or the system fails to move the email, the system logs the error and notifies the administrator.   
4. If the administrator does not have permission to archive the email, the system displays an error message and prevents the archiving.   
5. If the system fails to update the email list, it logs the error and notifies the administrator.  
  
Use Case Name: Search Archived Email   
Use Case ID: UC-07   
Actors: Administrator, Archive, Server Storage   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to search archived emails.   
3. The archive feature is configured and accessible.   
4. There are archived emails in the system.   
  
Postconditions:   
1. The system displays the search results of archived emails.   
2. The administrator can view details of the searched emails.   
3. The system logs the search activity.   
  
Main Flow:   
1. The administrator navigates to the "Search Archived Email" feature.   
2. The system displays a search interface with options to filter by subject, sender, date, etc.   
3. The administrator enters search criteria and clicks "Search".   
4. The system queries the archive for matching emails.   
5. The system retrieves the matching emails from the archive and displays them in a list.   
6. The administrator selects an email to view its details.   
7. The system retrieves and displays the selected email's information.   
8. The system logs the search and viewing activity.   
  
Alternative Flow:   
1. If the search criteria are invalid, the system displays an error message and prompts the administrator to correct the input.   
2. If no archived emails match the search criteria, the system displays a message indicating no results found.   
3. If the archive is not accessible, the system logs the error and notifies the administrator.   
4. If the selected email cannot be retrieved, the system displays an error message and logs the issue.   
5. If the administrator cancels the search, the system returns to the previous screen without performing the search.  
  
Use Case Name: Manage Contacts   
Use Case ID: UC-08   
Actors: Administrator, Contact, Distribution Group   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to manage contacts.   
3. The contact or distribution group feature is accessible in the system.   
  
Postconditions:   
1. The contact list is updated to reflect any additions, edits, or deletions.   
2. The system logs the contact management activity.   
3. The administrator receives a confirmation message for the performed action.   
  
Main Flow:   
1. The administrator navigates to the "Manage Contacts" feature.   
2. The system displays a list of existing contacts and distribution groups.   
3. The administrator selects an action (Add Contact, Edit Contact, Delete Contact, or Manage Distribution Group).   
4. If adding or editing, the system displays a form for contact details (name, email, phone, etc.).   
5. The administrator fills in or updates the contact information and submits it.   
6. The system validates the input data.   
7. The system updates the contact list or distribution group accordingly.   
8. The system confirms the action and displays the updated list.   
9. The system logs the contact management activity.   
  
Alternative Flow:   
1. If the administrator cancels the operation, the system returns to the contact list without making changes.   
2. If the input data is invalid, the system displays an error message and prompts the administrator to correct it.   
3. If the administrator does not have permission to manage contacts, the system displays an error message and prevents the action.   
4. If the system fails to update the contact list or distribution group, it logs the error and notifies the administrator.   
5. If the selected contact or distribution group does not exist, the system displays an error message and logs the issue.  
  
Use Case Name: Create Distribution Group   
Use Case ID: UC-09   
Actors: Administrator, Contact, Email Account   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to create a distribution group.   
3. At least one contact or email account exists in the system to be added to the group.   
  
Postconditions:   
1. A new distribution group is created and stored in the system.   
2. The group includes the specified contacts or email accounts.   
3. The system logs the creation of the distribution group.   
4. The administrator receives a confirmation message for the successful creation.   
  
Main Flow:   
1. The administrator navigates to the "Manage Distribution Groups" section.   
2. The system displays an option to "Create Distribution Group".   
3. The administrator clicks on the option and enters the group name.   
4. The administrator selects contacts or email accounts to be included in the group.   
5. The administrator confirms the selections and clicks "Create".   
6. The system validates the group name and selected members.   
7. The system creates the distribution group and stores it in the system.   
8. The system updates the distribution group list and provides a success message.   
  
Alternative Flow:   
1. If the group name is invalid or already exists, the system displays an error message and prompts the administrator to correct it.   
2. If no contacts or email accounts are selected, the system displays an error and prevents the creation.   
3. If the administrator cancels the operation, the system returns to the distribution group list without changes.   
4. If the system fails to store the group for any reason, it logs the error and notifies the administrator.  
  
Use Case Name: Update Distribution Group   
Use Case ID: UC-10   
Actors: Administrator, Distribution Group, Contact, Email Account   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to update a distribution group.   
3. A distribution group exists in the system.   
4. The contact or email account feature is accessible in the system.   
  
Postconditions:   
1. The distribution group is updated with the specified changes.   
2. The updated group is stored in the system.   
3. The system logs the update activity.   
4. The administrator receives a confirmation message for the successful update.   
  
Main Flow:   
1. The administrator navigates to the "Manage Distribution Groups" section.   
2. The system displays a list of existing distribution groups.   
3. The administrator selects the distribution group to be updated.   
4. The system displays the details of the selected distribution group and allows editing.   
5. The administrator modifies the group name or adds/removes contacts and email accounts.   
6. The administrator clicks "Save Changes".   
7. The system validates the updated information.   
8. The system updates the distribution group and saves the changes in the system.   
9. The system confirms the update and displays the updated group in the list.   
  
Alternative Flow:   
1. If the updated group name is invalid or already exists, the system displays an error message and prompts the administrator to correct it.   
2. If no changes are made to the group, the system displays a message and returns to the group list.   
3. If the administrator cancels the update, the system discards the changes and returns to the group list.   
4. If the system fails to save the updated group, it logs the error and notifies the administrator.  
  
Use Case Name: Delete Distribution Group   
Use Case ID: UC-11   
Actors: Administrator, Distribution Group, Contact, Email Account   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to delete a distribution group.   
3. A distribution group exists in the system.   
4. The system is able to access and modify the distribution group and its members.   
  
Postconditions:   
1. The selected distribution group is removed from the system.   
2. All references to the group are removed from email and contact management features.   
3. The system logs the deletion activity.   
4. The administrator receives a confirmation message for the successful deletion.   
  
Main Flow:   
1. The administrator navigates to the "Manage Distribution Groups" section.   
2. The system displays a list of existing distribution groups.   
3. The administrator selects the distribution group to be deleted.   
4. The administrator clicks the "Delete" button.   
5. The system confirms the deletion request with the administrator.   
6. The administrator confirms the deletion.   
7. The system removes the selected distribution group from the system.   
8. The system updates the distribution group list to reflect the deletion.   
9. The system logs the deletion activity.   
10. The system provides a success message indicating the distribution group was deleted.   
  
Alternative Flow:   
1. If the administrator cancels the deletion confirmation, the system retains the distribution group and returns to the group list.   
2. If the selected distribution group does not exist, the system displays an error message and logs the issue.   
3. If the system fails to delete the group (e.g., due to dependencies or access issues), it logs the error and notifies the administrator.   
4. If the administrator does not have permission to delete the distribution group, the system displays an error message and prevents the deletion.  
  
Use Case Name: Manage Calendar   
Use Case ID: UC-12   
Actors: Administrator, Calendar, Email   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to manage the calendar.   
3. A calendar feature is available and properly configured in the system.   
4. At least one email exists that can be scheduled or is already scheduled.   
  
Postconditions:   
1. The calendar is updated to reflect any additions, edits, or deletions of scheduled events.   
2. The system logs the calendar management activity.   
3. The administrator receives confirmation for the performed action.   
  
Main Flow:   
1. The administrator navigates to the "Calendar" feature.   
2. The system displays a calendar interface showing scheduled emails.   
3. The administrator selects an action (Add Event, Edit Event, or Delete Event).   
4. If adding an event, the administrator creates a new scheduled email by selecting a date and time.   
5. If editing or deleting, the administrator selects a specific event to modify or remove.   
6. The system validates the selected action and updates the calendar accordingly.   
7. The system confirms the action and displays the updated calendar view.   
8. The system logs the calendar management activity.   
  
Alternative Flow:   
1. If the calendar feature is not accessible, the system logs the error and notifies the administrator.   
2. If the selected event does not exist, the system displays an error message and logs the issue.   
3. If the administrator cancels the operation, the system returns to the calendar view without making changes.   
4. If the system fails to update the calendar, it logs the error and notifies the administrator.  
  
Use Case Name: Create Calendar Event   
Use Case ID: UC-13   
Actors: Administrator, Calendar, Email   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to create a calendar event.   
3. The calendar feature is accessible and properly configured.   
4. The email feature is available and operational.   
  
Postconditions:   
1. A new calendar event is created and associated with a scheduled email.   
2. The calendar interface is updated to reflect the new event.   
3. The system logs the creation of the calendar event.   
4. The administrator receives a confirmation message for the successful creation.   
  
Main Flow:   
1. The administrator navigates to the "Calendar" feature.   
2. The system displays the calendar interface.   
3. The administrator selects the "Create Event" option.   
4. The administrator specifies the date, time, and details for the event.   
5. The administrator links the event to an email by selecting an email or composing a new one.   
6. The system validates the event details and email association.   
7. The system creates the calendar event and stores it.   
8. If an email was composed, the system schedules it using the specified time.   
9. The system updates the calendar view and provides a success message.   
  
Alternative Flow:   
1. If the event details are invalid, the system displays an error message and prompts the administrator to correct the input.   
2. If the selected email is not valid or cannot be scheduled, the system displays an error and cancels the event creation.   
3. If the administrator cancels the operation before submission, the system discards the event and returns to the calendar view.   
4. If the system fails to store the event or schedule the email, it logs the error and notifies the administrator.  
  
Use Case Name: Update Calendar Event   
Use Case ID: UC-14   
Actors: Administrator, Calendar, Email   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to update a calendar event.   
3. A calendar event exists in the system and is associated with a scheduled email.   
4. The calendar and email features are accessible and properly configured.   
  
Postconditions:   
1. The calendar event is updated with the new details.   
2. The associated email is rescheduled or modified accordingly.   
3. The system logs the update activity.   
4. The administrator receives a confirmation message for the successful update.   
  
Main Flow:   
1. The administrator navigates to the "Calendar" feature.   
2. The system displays the calendar interface with existing scheduled events.   
3. The administrator selects an event to update.   
4. The system displays the event details and allows editing of the date, time, or associated email.   
5. The administrator modifies the event details or updates the email content.   
6. The administrator clicks "Save Changes".   
7. The system validates the updated event and email information.   
8. The system updates the calendar event and reschedules or modifies the associated email.   
9. The system confirms the update and displays the revised calendar view.   
  
Alternative Flow:   
1. If the updated event details are invalid, the system displays an error message and prompts the administrator to correct the input.   
2. If the associated email is invalid or cannot be rescheduled, the system displays an error and cancels the update.   
3. If the administrator cancels the operation, the system discards the changes and returns to the calendar view.   
4. If the system fails to update the event or reschedule the email, it logs the error and notifies the administrator.  
  
Use Case Name: Delete Calendar Event   
Use Case ID: UC-15   
Actors: Administrator, Calendar, Email   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to delete calendar events.   
3. A calendar event exists in the system and is associated with a scheduled email.   
4. The calendar and email features are accessible and properly configured.   
  
Postconditions:   
1. The selected calendar event is removed from the calendar.   
2. The associated scheduled email is either deleted or updated accordingly.   
3. The system logs the deletion activity.   
4. The administrator receives a confirmation message for the successful deletion.   
  
Main Flow:   
1. The administrator navigates to the "Calendar" feature.   
2. The system displays the calendar interface with existing events.   
3. The administrator selects the calendar event to be deleted.   
4. The administrator clicks the "Delete" button.   
5. The system confirms the deletion request with the administrator.   
6. The administrator confirms the deletion.   
7. The system removes the selected calendar event from the calendar.   
8. If the event is linked to a scheduled email, the system either deletes the email or updates its status.   
9. The system updates the calendar view to reflect the deletion.   
10. The system logs the deletion activity and provides a success message.   
  
Alternative Flow:   
1. If the administrator cancels the deletion confirmation, the system retains the calendar event and returns to the calendar view.   
2. If the selected calendar event does not exist, the system displays an error message and logs the issue.   
3. If the associated email cannot be deleted or modified, the system displays an error and logs the issue.   
4. If the system fails to update the calendar or email, it logs the error and notifies the administrator.  
  
Use Case Name: Recover Expired Email   
Use Case ID: UC-16   
Actors: Administrator, Expired Email, Server Storage, Recovery Policy   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to recover expired emails.   
3. An expired email exists in the system that can be recovered.   
4. The recovery policy is configured and accessible.   
  
Postconditions:   
1. The expired email is restored from its storage location (e.g., archive or backup).   
2. The email is re-associated with the correct email account.   
3. The system logs the recovery activity.   
4. The administrator receives a confirmation message for the successful recovery.   
  
Main Flow:   
1. The administrator navigates to the "Expired Emails" section or recovery feature.   
2. The system displays a list of expired emails that are eligible for recovery.   
3. The administrator selects the expired email to recover.   
4. The administrator clicks the "Recover" button.   
5. The system confirms the recovery request with the administrator.   
6. The administrator confirms the recovery action.   
7. The system retrieves the expired email based on the recovery policy.   
8. The system restores the email to the server storage and associates it with the relevant email account.   
9. The system logs the recovery activity.   
10. The system provides a success message indicating the email was successfully recovered.   
  
Alternative Flow:   
1. If the administrator cancels the recovery confirmation, the system returns to the expired email list without changes.   
2. If the selected expired email does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the recovery policy is not configured or accessible, the system displays an error and prevents the recovery.   
4. If the system fails to restore the email to the server storage, it logs the error and notifies the administrator.   
5. If the administrator does not have permission to recover expired emails, the system displays an error message and prevents the action.  
  
Use Case Name: Modify Email Account   
Use Case ID: UC-17   
Actors: Administrator, Email Account   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to modify email accounts.   
3. An email account exists in the system and is selected for modification.   
4. The email account configuration feature is accessible.   
  
Postconditions:   
1. The selected email account is updated with the new configuration.   
2. The system logs the modification activity.   
3. The administrator receives a confirmation message for the successful modification.   
  
Main Flow:   
1. The administrator navigates to the "Manage Email Accounts" section.   
2. The system displays a list of existing email accounts.   
3. The administrator selects the email account to be modified.   
4. The system displays the current configuration details of the email account and allows editing.   
5. The administrator updates the email account information (e.g., password, server settings).   
6. The administrator clicks "Save Changes".   
7. The system validates the updated information.   
8. The system updates the email account configuration and saves the changes.   
9. The system confirms the modification and displays the updated account in the list.   
  
Alternative Flow:   
1. If the updated information is invalid, the system displays an error message and prompts the administrator to correct it.   
2. If the selected email account does not exist, the system displays an error message and logs the issue.   
3. If the administrator cancels the modification, the system returns to the email account list without changes.   
4. If the system fails to save the updated configuration, it logs the error and notifies the administrator.  
  
Use Case Name: Delete Email Account   
Use Case ID: UC-18   
Actors: Administrator, Email Account, Server Storage   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to delete email accounts.   
3. An email account exists in the system and is selected for deletion.   
4. The system can access and modify the email account and related data.   
  
Postconditions:   
1. The selected email account is removed from the system.   
2. All emails associated with the account are either deleted or moved according to the configured policy.   
3. The system logs the deletion activity.   
4. The administrator receives a confirmation message for the successful deletion.   
  
Main Flow:   
1. The administrator navigates to the "Manage Email Accounts" section.   
2. The system displays a list of existing email accounts.   
3. The administrator selects the email account to be deleted.   
4. The administrator clicks the "Delete" button.   
5. The system confirms the deletion request with the administrator.   
6. The administrator confirms the deletion.   
7. The system deletes the email account configuration from the system.   
8. The system processes all associated emails according to the configured policy (e.g., delete, archive).   
9. The system updates the email account list to reflect the deletion.   
10. The system logs the deletion activity and provides a success message.   
  
Alternative Flow:   
1. If the administrator cancels the deletion confirmation, the system retains the email account and returns to the account list.   
2. If the selected email account does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the system fails to process associated emails (e.g., due to storage issues), it logs the error and notifies the administrator.   
4. If the administrator does not have permission to delete the email account, the system displays an error message and prevents the deletion.   
5. If the system fails to delete the email account configuration, it logs the error and notifies the administrator.  
  
Use Case Name: Manage Server Storage   
Use Case ID: UC-19   
Actors: Administrator, Server Storage   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to manage server storage.   
3. The server storage is accessible and properly configured.   
4. Emails or other data may exist in the server storage.   
  
Postconditions:   
1. The server storage configuration is updated according to the administrator's request.   
2. The system logs the server storage management activity.   
3. The administrator receives confirmation for the performed action.   
  
Main Flow:   
1. The administrator navigates to the "Server Storage Management" section.   
2. The system displays the current server storage settings and options (e.g., capacity, retention policy, backup settings).   
3. The administrator selects an action (e.g., expand storage, modify retention policy, enable backup).   
4. The administrator updates the storage settings as required.   
5. The administrator clicks "Save Changes".   
6. The system validates the updated storage configuration.   
7. The system applies the changes to the server storage.   
8. The system confirms the successful modification and displays the updated storage settings.   
  
Alternative Flow:   
1. If the administrator cancels the operation, the system returns to the server storage settings without making changes.   
2. If the updated configuration is invalid, the system displays an error message and prompts the administrator to correct it.   
3. If the system fails to apply the changes (e.g., due to storage unavailability or system error), it logs the error and notifies the administrator.   
4. If the administrator does not have permission to modify server storage settings, the system displays an error message and prevents the action.  
  
Use Case Name: Set Archiving Policy   
Use Case ID: UC-20   
Actors: Administrator, Server Storage, Archive   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to configure archiving policies.   
3. The server storage and archive features are configured and accessible.   
4. No active archiving process is currently running for the selected email account.   
  
Postconditions:   
1. The archiving policy is updated with the specified settings.   
2. The system logs the policy change.   
3. The administrator receives a confirmation message for the successful update.   
  
Main Flow:   
1. The administrator navigates to the "Manage Archiving Policies" section.   
2. The system displays a list of existing archiving policies and an option to create or edit a policy.   
3. The administrator selects the "Set Archiving Policy" option for a specific email account or globally.   
4. The system displays a form for configuring the archiving policy (e.g., retention period, archive location, conditions for archiving).   
5. The administrator fills in the policy details and clicks "Save Policy".   
6. The system validates the input and checks for conflicts with existing policies.   
7. The system updates the archiving policy in the system configuration.   
8. The system confirms the policy update and displays the new settings.   
9. The system logs the archiving policy change.   
  
Alternative Flow:   
1. If the administrator cancels the operation, the system returns to the policy list without making changes.   
2. If the input is invalid or incomplete, the system displays an error message and prompts the administrator to correct it.   
3. If the selected email account is not valid or not configured, the system displays an error and prevents the policy update.   
4. If there is a conflict with an existing policy, the system warns the administrator and suggests resolution.   
5. If the system fails to save the policy, it logs the error and notifies the administrator.  
  
Use Case Name: Update Archiving Policy   
Use Case ID: UC-21   
Actors: Administrator, Server Storage, Archive   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to update archiving policies.   
3. An existing archiving policy is configured in the system.   
4. The server storage and archive features are accessible.   
  
Postconditions:   
1. The archiving policy is updated with the new settings.   
2. The system logs the policy update activity.   
3. The administrator receives a confirmation message for the successful update.   
  
Main Flow:   
1. The administrator navigates to the "Manage Archiving Policies" section.   
2. The system displays the current archiving policy details.   
3. The administrator selects the "Edit Archiving Policy" option.   
4. The system provides a form with editable fields (e.g., retention period, archive location, conditions for archiving).   
5. The administrator modifies the policy settings as needed.   
6. The administrator clicks "Save Changes".   
7. The system validates the updated policy settings.   
8. The system updates the archiving policy configuration.   
9. The system confirms the policy update and displays the revised settings.   
10. The system logs the update activity.   
  
Alternative Flow:   
1. If the administrator cancels the edit, the system returns to the policy list without making changes.   
2. If the input settings are invalid or incomplete, the system displays an error and prompts the administrator to correct them.   
3. If the archive or server storage is not accessible, the system displays an error and prevents the update.   
4. If the system fails to apply the updated policy, it logs the error and notifies the administrator.  
  
Use Case Name: Delete Archiving Policy   
Use Case ID: UC-22   
Actors: Administrator, Server Storage, Archive   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to delete archiving policies.   
3. An existing archiving policy is configured in the system.   
4. The server storage and archive features are accessible.   
  
Postconditions:   
1. The selected archiving policy is removed from the system configuration.   
2. The system logs the deletion activity.   
3. The administrator receives a confirmation message for the successful deletion.   
  
Main Flow:   
1. The administrator navigates to the "Manage Archiving Policies" section.   
2. The system displays a list of existing archiving policies.   
3. The administrator selects the archiving policy to be deleted.   
4. The administrator clicks the "Delete" button.   
5. The system confirms the deletion request with the administrator.   
6. The administrator confirms the deletion.   
7. The system removes the selected archiving policy from the system configuration.   
8. The system updates the policy list to reflect the deletion.   
9. The system logs the deletion activity and provides a success message.   
  
Alternative Flow:   
1. If the administrator cancels the deletion confirmation, the system retains the policy and returns to the policy list.   
2. If the selected archiving policy does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the system fails to delete the policy (e.g., due to dependencies or access issues), it logs the error and notifies the administrator.   
4. If the administrator does not have permission to delete the archiving policy, the system displays an error message and prevents the deletion.  
  
Use Case Name: Capture Email Flow   
Use Case ID: UC-23   
Actors: Administrator, Email, Email Account, Server Storage, Policy   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to capture email flows.   
3. An email account exists in the system and is active.   
4. The system has a configured policy for email flow capture.   
5. The server storage is accessible and ready for data storage.   
  
Postconditions:   
1. The email flow is captured and stored in the server storage.   
2. The system logs the capture activity.   
3. The administrator receives a confirmation message for the successful capture.   
4. The captured flow is associated with the appropriate email account and policy.   
  
Main Flow:   
1. The administrator navigates to the "Capture Email Flow" feature.   
2. The system displays a list of email accounts and policies available for capturing flows.   
3. The administrator selects an email account and the corresponding policy for the flow.   
4. The administrator defines the flow criteria (e.g., source, destination, conditions for triggering).   
5. The administrator clicks "Capture" to initiate the flow monitoring process.   
6. The system validates the selected account, policy, and flow criteria.   
7. The system begins capturing the email flow according to the defined policy.   
8. The system stores the captured flow data in the server storage.   
9. The system provides a success message and updates the administrator's interface.   
  
Alternative Flow:   
1. If the selected email account is not valid or not configured, the system displays an error and cancels the capture.   
2. If the policy is not configured or does not match the selected account, the system displays an error and prevents the capture.   
3. If the flow criteria are invalid or incomplete, the system displays an error message and prompts the administrator to correct them.   
4. If the server storage is unavailable, the system logs the error and notifies the administrator.   
5. If the administrator cancels the operation before initiating the capture, the system returns to the previous screen without changes.   
6. If the system fails to start capturing the flow, it logs the error and notifies the administrator.  
  
Use Case Name: Manage Expired Email   
Use Case ID: UC-16   
Actors: Administrator, Expired Email, Server Storage, Recovery Policy   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to manage expired emails.   
3. Expired emails exist in the system (e.g., in an archive or backup location).   
4. The recovery policy is configured and accessible.   
  
Postconditions:   
1. The expired email is either restored or permanently removed based on the administrator's action.   
2. The system updates the relevant storage or archive to reflect the change.   
3. The system logs the management activity of the expired email.   
4. The administrator receives a confirmation message for the performed action.   
  
Main Flow:   
1. The administrator navigates to the "Manage Expired Emails" section.   
2. The system displays a list of expired emails available for recovery or deletion.   
3. The administrator selects an expired email to manage.   
4. The administrator chooses to either "Recover" or "Delete" the email.   
5. If the action is "Recover", the system confirms the recovery request with the administrator.   
6. If the action is "Delete", the system confirms the deletion request with the administrator.   
7. The administrator confirms the selected action.   
8. The system retrieves the expired email using the recovery policy or removes it from the system.   
9. The system updates the server storage or archive accordingly.   
10. The system logs the activity and provides a success message.   
  
Alternative Flow:   
1. If the administrator cancels the action confirmation, the system retains the expired email in the list and returns to the previous screen.   
2. If the selected expired email cannot be retrieved or processed, the system displays an error message and logs the issue.   
3. If the recovery policy is not configured or accessible, the system displays an error and prevents the recovery.   
4. If the system fails to update the storage or archive, it logs the error and notifies the administrator.   
5. If the administrator does not have permission to manage expired emails, the system displays an error message and prevents the action.  
  
Use Case Name: Recover Expired Email   
Use Case ID: UC-16   
Actors: Administrator, Expired Email, Server Storage, Recovery Policy   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to recover expired emails.   
3. An expired email exists in the system (e.g., in an archive or backup location).   
4. The recovery policy is configured and accessible.   
  
Postconditions:   
1. The expired email is either restored or permanently removed based on the administrator's action.   
2. The system updates the relevant storage or archive to reflect the change.   
3. The system logs the management activity of the expired email.   
4. The administrator receives a confirmation message for the performed action.   
  
Main Flow:   
1. The administrator navigates to the "Manage Expired Emails" section.   
2. The system displays a list of expired emails available for recovery or deletion.   
3. The administrator selects an expired email to manage.   
4. The administrator chooses to either "Recover" or "Delete" the email.   
5. The system confirms the selected action with the administrator.   
6. The administrator confirms the action.   
7. The system retrieves the expired email using the recovery policy or removes it from the system.   
8. The system updates the server storage or archive accordingly.   
9. The system logs the activity and provides a success message.   
  
Alternative Flow:   
1. If the administrator cancels the action confirmation, the system retains the expired email in the list and returns to the previous screen.   
2. If the selected expired email cannot be retrieved or processed, the system displays an error message and logs the issue.   
3. If the recovery policy is not configured or accessible, the system displays an error and prevents the recovery.   
4. If the system fails to update the storage or archive, it logs the error and notifies the administrator.   
5. If the administrator does not have permission to manage expired emails, the system displays an error message and prevents the action.  
  
Use Case Name: Audit Email Records   
Use Case ID: UC-24   
Actors: Administrator, Email, Server Storage   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to audit email records.   
3. The server storage is accessible and contains email records.   
4. The audit feature is enabled and configured in the system.   
  
Postconditions:   
1. The system displays the audit report or logs of email records.   
2. The system logs the audit activity.   
3. The administrator receives a confirmation message for the audit process.   
  
Main Flow:   
1. The administrator navigates to the "Audit Email Records" section.   
2. The system displays options to filter the audit by date, email account, or type of activity.   
3. The administrator selects the desired audit criteria and clicks "Generate Report".   
4. The system queries the server storage for relevant email records and audit logs.   
5. The system compiles the audit data into a report or list.   
6. The system displays the audit results to the administrator.   
7. The system logs the audit activity performed.   
8. The administrator confirms the audit completion.   
  
Alternative Flow:   
1. If the audit criteria are invalid, the system displays an error message and prompts the administrator to correct the input.   
2. If the server storage is unavailable, the system logs the error and notifies the administrator.   
3. If no audit records match the criteria, the system displays a message indicating no results were found.   
4. If the administrator cancels the audit process before completion, the system returns to the audit interface without generating a report.   
5. If the system fails to compile or display the audit report, it logs the error and notifies the administrator.  
  
Use Case Name: Manage Administrator Permissions   
Use Case ID: UC-25   
Actors: Administrator, Email Account, Server Storage, Policy   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to manage other administrators or user roles.   
3. The system has at least one administrator account or user role configured.   
4. The permission management feature is accessible and properly configured.   
  
Postconditions:   
1. Administrator permissions are updated or assigned according to the specified changes.   
2. The system logs the permission management activity.   
3. The administrator receives a confirmation message for the successful action.   
4. The updated permissions are enforced for the affected user(s).   
  
Main Flow:   
1. The administrator navigates to the "Manage Administrator Permissions" section.   
2. The system displays a list of existing administrator accounts or user roles.   
3. The administrator selects an account or role to modify.   
4. The system displays the current permission settings and allows editing.   
5. The administrator updates or assigns permissions (e.g., send email, view archive, manage policies).   
6. The administrator clicks "Save Changes".   
7. The system validates the updated permission settings.   
8. The system updates the permission configuration in the system.   
9. The system confirms the successful update and displays the revised permissions.   
10. The system logs the permission management activity.   
  
Alternative Flow:   
1. If the administrator cancels the operation, the system returns to the permission list without making changes.   
2. If the updated permissions are invalid or conflicting, the system displays an error message and prompts the administrator to correct them.   
3. If the system fails to save the updated permissions, it logs the error and notifies the administrator.   
4. If the administrator does not have permission to manage other administrators, the system displays an error message and prevents the action.   
5. If the selected administrator account or role does not exist, the system displays an error and logs the issue.  
  
Use Case Name: Update Administrator Settings   
Use Case ID: UC-26   
Actors: Administrator, Email Account, Server Storage, Policy   
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to update administrator settings.   
3. The system has an existing administrator account to modify.   
4. The administrator settings feature is accessible and properly configured.   
  
Postconditions:   
1. The administrator settings are updated with the specified changes.   
2. The system logs the update activity.   
3. The updated settings are enforced for the administrator account.   
4. The administrator receives a confirmation message for the successful update.   
  
Main Flow:   
1. The administrator navigates to the "Administrator Settings" section.   
2. The system displays the current settings for the administrator account (e.g., password, access level, notification preferences).   
3. The administrator selects the settings to be modified.   
4. The system provides an interface to update the selected settings.   
5. The administrator inputs new values for the settings and clicks "Save Changes".   
6. The system validates the updated settings (e.g., password strength, access level permissions).   
7. The system updates the administrator account with the new settings.   
8. The system confirms the update and displays the revised settings.   
9. The system logs the activity.   
  
Alternative Flow:   
1. If the administrator cancels the operation, the system returns to the settings interface without making changes.   
2. If the updated settings are invalid (e.g., password does not meet criteria), the system displays an error message and prompts the administrator to correct them.   
3. If the system fails to update the administrator account, it logs the error and notifies the administrator.   
4. If the administrator does not have permission to update settings, the system displays an error message and prevents the action.   
5. If the server storage is unavailable, the system logs the error and notifies the administrator.  
  
Use Case Name: Delete Administrator Account   
Use Case ID: UC-27   
Actors: Administrator, Email Account, Server Storage   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to delete administrator accounts.   
3. The administrator account to be deleted exists in the system.   
4. The system can access and modify the administrator account and related data.   
  
Postconditions:   
1. The selected administrator account is removed from the system.   
2. All associated data (e.g., emails, calendar events, policies) is either deleted or transferred according to the configured policy.   
3. The system logs the deletion activity.   
4. The administrator receives a confirmation message for the successful deletion.   
  
Main Flow:   
1. The administrator navigates to the "Manage Administrator Accounts" section.   
2. The system displays a list of existing administrator accounts.   
3. The administrator selects the account to be deleted.   
4. The administrator clicks the "Delete" button.   
5. The system confirms the deletion request with the administrator.   
6. The administrator confirms the deletion.   
7. The system removes the administrator account from the system.   
8. The system processes all associated data (e.g., delete, archive, or transfer).   
9. The system updates the administrator account list to reflect the deletion.   
10. The system logs the deletion activity and provides a success message.   
  
Alternative Flow:   
1. If the administrator cancels the deletion confirmation, the system retains the account and returns to the account list.   
2. If the selected administrator account does not exist, the system displays an error message and logs the issue.   
3. If the system fails to process associated data (e.g., due to storage issues or policy conflicts), it logs the error and notifies the administrator.   
4. If the administrator does not have permission to delete the account, the system displays an error message and prevents the action.   
5. If the system fails to delete the administrator account, it logs the error and notifies the administrator.  
  
Use Case Name: View Email Record   
Use Case ID: UC-28   
Actors: Administrator, Email Record, Server Storage   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to view email records.   
3. An email record exists in the server storage for the selected email account.   
4. The system is able to access the email record.   
  
Postconditions:   
1. The email record is displayed to the administrator in a detailed and structured format.   
2. The system logs the viewing activity of the email record.   
3. The administrator may choose to take further actions (e.g., export, audit, or delete the record).   
  
Main Flow:   
1. The administrator navigates to the "Email Records" section or selects the option to view a specific email record.   
2. The system retrieves and displays a list of available email records in the server storage.   
3. The administrator selects an email record to view.   
4. The system fetches the selected email record from the server storage.   
5. The system displays the email record with detailed information, including sender, recipient, subject, content, date, and associated metadata.   
6. The system logs the viewing activity for the selected email record.   
7. The administrator can optionally export the record or navigate back to the email record list.   
  
Alternative Flow:   
1. If the administrator does not have the required permission to view the email record, the system displays an error message and prevents access.   
2. If the selected email record does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the server storage is unavailable, the system logs the error and notifies the administrator.   
4. If the administrator cancels the operation before viewing, the system returns to the email record list or the previous screen.   
5. If the system fails to display the email record (e.g., due to data corruption or retrieval issues), it logs the error and notifies the administrator.  
  
Use Case Name: Manage Email Folder   
Use Case ID: UC-29   
Actors: Administrator, Email Folder, Email, Server Storage   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to manage email folders.   
3. The email folder feature is accessible and properly configured.   
4. The server storage is operational and accessible.   
  
Postconditions:   
1. The email folder is updated, created, or deleted based on the administrator's action.   
2. Emails associated with the folder are reorganized accordingly.   
3. The system logs the folder management activity.   
4. The administrator receives a confirmation message for the performed action.   
  
Main Flow:   
1. The administrator navigates to the "Manage Email Folders" section.   
2. The system displays a list of existing email folders and an option to create a new folder.   
3. The administrator selects an action (Create Folder, Rename Folder, Move Emails to Folder, Delete Folder).   
4. If creating or renaming a folder, the system provides a form for the folder name and description.   
5. The administrator fills in the required details and submits the action.   
6. If moving emails, the administrator selects one or more emails to be moved to the specified folder.   
7. The system validates the folder name and the availability of the folder.   
8. The system updates the folder structure and modifies the association of emails with the folder.   
9. The system confirms the action and displays the updated folder structure.   
10. The system logs the folder management activity.   
  
Alternative Flow:   
1. If the administrator cancels the operation, the system returns to the folder list without making changes.   
2. If the folder name is invalid or already exists, the system displays an error message and prompts the administrator to correct it.   
3. If the system fails to update the folder structure or modify email associations, it logs the error and notifies the administrator.   
4. If the selected emails cannot be moved (e.g., due to access restrictions or invalid state), the system displays an error and prevents the operation.   
5. If the server storage is unavailable, the system logs the error and notifies the administrator.   
6. If the administrator does not have permission to manage email folders, the system displays an error message and prevents the action.  
  
Use Case Name: Manage Email Attachment   
Use Case ID: UC-30   
Actors: Administrator, Email, Email Account, Server Storage   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to manage email attachments.   
3. An email with an attachment exists in the system.   
4. The server storage is accessible and properly configured.   
  
Postconditions:   
1. The email attachment is either added, viewed, modified, or removed based on the administrator's action.   
2. The system logs the attachment management activity.   
3. The email record is updated to reflect changes in the attachments.   
4. The administrator receives a confirmation message for the performed action.   
  
Main Flow:   
1. The administrator navigates to the "Manage Email Attachments" section or selects an email with attachments for management.   
2. The system retrieves the selected email and displays a list of its associated attachments.   
3. The administrator selects an action (Add Attachment, View Attachment, Remove Attachment, or Rename Attachment).   
4. If adding an attachment, the administrator uploads a file and associates it with the email.   
5. If viewing an attachment, the system displays or opens the attachment for inspection.   
6. If removing an attachment, the system confirms the deletion with the administrator.   
7. The administrator confirms the action.   
8. The system updates the email record in the server storage with the new attachment information.   
9. The system logs the attachment management activity.   
10. The system provides a success message indicating the attachment was successfully managed.   
  
Alternative Flow:   
1. If the administrator cancels the operation, the system returns to the email details or attachment list without making changes.   
2. If the selected email does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the file uploaded as an attachment is invalid or exceeds the system's size limit, the system displays an error and prompts the administrator to select a valid file.   
4. If the system fails to update the email record or modify the attachments, it logs the error and notifies the administrator.   
5. If the administrator does not have permission to manage attachments, the system displays an error message and prevents the action.   
6. If the server storage is unavailable, the system logs the error and notifies the administrator.  
  
Use Case Name: Manage Calendar Event Participant   
Use Case ID: UC-31   
Actors: Administrator, Calendar, Calendar Event, Calendar Event Participant   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The administrator has the necessary permissions to manage calendar event participants.   
3. A calendar event exists in the system and is selected for modification.   
4. The calendar feature is accessible and properly configured.   
  
Postconditions:   
1. The calendar event is updated to reflect the addition, removal, or modification of participants.   
2. The system logs the calendar event participant management activity.   
3. The administrator receives a confirmation message for the performed action.   
4. The participant list associated with the calendar event is updated accordingly.   
  
Main Flow:   
1. The administrator navigates to the "Calendar" feature and selects a specific event.   
2. The system displays the event details, including the list of participants.   
3. The administrator selects the "Manage Participants" option.   
4. The system provides an interface to add, remove, or modify participants (e.g., search for contacts or distribution groups).   
5. The administrator performs the desired action (e.g., adds a new participant, removes an existing one, or edits participant details).   
6. The system validates the participant data and checks for conflicts (e.g., duplicate entries).   
7. The system updates the calendar event with the new participant information.   
8. The system saves the changes to the event and updates the participant list.   
9. The system logs the participant management activity.   
10. The system provides a success message confirming the changes to the calendar event participants.   
  
Alternative Flow:   
1. If the administrator cancels the operation, the system returns to the event details without making changes.   
2. If the selected calendar event does not exist or cannot be retrieved, the system displays an error message and logs the issue.   
3. If the participant data is invalid or incomplete, the system displays an error message and prompts the administrator to correct it.   
4. If the system fails to update the event or participant list, it logs the error and notifies the administrator.   
5. If the administrator does not have permission to manage calendar event participants, the system displays an error message and prevents the action.