项目文档

# Functional Requirement

# Chapter 1: Functional Requirements  
  
## 1.1 Email Sending Function   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: Users can compose and send emails to individuals, shared accounts, or distribution groups. The system validates recipient information and sends the email via the configured email server.   
\*\*Input\*\*: Email content (subject, body, attachments), recipient list (contacts, email addresses, distribution groups).   
\*\*Output\*\*: Sent email stored in the system and recipient inboxes; confirmation message displayed to the user.  
  
## 1.2 Email Receiving Function   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: The system checks the email server for new messages and processes incoming emails to the user's inbox or shared account inbox.   
\*\*Input\*\*: Incoming email data from the email server.   
\*\*Output\*\*: Email displayed in the user’s inbox or shared account inbox; email logged in the archive for compliance.  
  
## 1.3 Email Formatting Function   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: Users can format email content with font style, size, color, and other text formatting options before sending.   
\*\*Input\*\*: Draft or composed email content.   
\*\*Output\*\*: Formatted email content; visual confirmation of formatting applied.  
  
## 1.4 Contact Management Function   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: Users can add, edit, or delete contacts in their personal or shared contact list. The system validates contact data and updates related email or distribution group configurations.   
\*\*Input\*\*: Contact details (name, email, phone, etc.).   
\*\*Output\*\*: Updated contact list; confirmation message displayed to the user.  
  
## 1.5 Distribution Group Creation Function   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: Users or administrators can create a new distribution group by selecting existing contacts or manually adding email addresses.   
\*\*Input\*\*: Group name, description, and member list (contacts or email addresses).   
\*\*Output\*\*: New distribution group stored in the system; confirmation message displayed.  
  
## 1.6 Distribution Group Update Function   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: Users or administrators can modify the name, description, or member list of an existing distribution group.   
\*\*Input\*\*: Updated group name, description, and/or member list.   
\*\*Output\*\*: Updated distribution group stored in the database; confirmation message displayed.  
  
## 1.7 Distribution Group Deletion Function   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: Users or administrators can delete a distribution group after confirming the action.   
\*\*Input\*\*: Selected distribution group.   
\*\*Output\*\*: Distribution group removed from the system; confirmation message displayed.  
  
## 1.8 Email Account Management Function   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: Users and administrators can create, update, or delete email accounts. The system validates account information and ensures no active dependencies prevent deletion.   
\*\*Input\*\*: Email account details (email address, password, type).   
\*\*Output\*\*: Updated or deleted email account; confirmation message displayed.  
  
## 1.9 Shared Account Management Function   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: Administrators can manage shared accounts by assigning access rights and modifying configurations.   
\*\*Input\*\*: Shared account details, user/group access rights.   
\*\*Output\*\*: Updated or deleted shared account; access rights updated in the system.  
  
## 1.10 Email Archiving Function   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: Users or administrators can archive selected emails to the system archive. The system ensures that emails are moved and metadata is preserved.   
\*\*Input\*\*: Selected email(s) from inbox or folders.   
\*\*Output\*\*: Archived email(s) in the archive database; confirmation message displayed.  
  
## 1.11 Personal Archive Search Function   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: Users can search their personal archive using criteria such as keywords, date range, sender, or recipient.   
\*\*Input\*\*: Search criteria (keywords, dates, sender/recipient).   
\*\*Output\*\*: Matching archived emails displayed with metadata; selected emails can be restored.  
  
## 1.12 Server Archive Management Function   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: Administrators can manage the server-side archive, including backing up, purging expired emails, and restoring archived data.   
\*\*Input\*\*: Server archive management options (backup, purge, restore).   
\*\*Output\*\*: Updated server archive; logs of the action; confirmation message displayed.  
  
## 1.13 Email Flow Capture Function   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: Administrators can configure the system to capture the flow of emails, including sender, recipient, timestamps, and routing information.   
\*\*Input\*\*: Scope (user, shared account, distribution group).   
\*\*Output\*\*: Captured email flow data stored in the system; administrator can view or export the data.  
  
## 1.14 Retention Policy Application Function   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: Administrators can apply retention policies to email accounts or shared accounts, defining how long emails should be retained.   
\*\*Input\*\*: Selected retention policy and target email accounts or shared accounts.   
\*\*Output\*\*: Policy applied to the selected accounts; logs of the action; confirmation message displayed.  
  
## 1.15 Expired Email Management Function   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: Users or administrators can manage expired emails by either archiving or deleting them based on retention policy rules.   
\*\*Input\*\*: List of expired emails; action (archive or delete).   
\*\*Output\*\*: Emails either archived or deleted; logs of the action; confirmation message displayed.  
  
## 1.16 Email Recovery Function   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: Users or administrators can recover archived emails, restoring them to the inbox or a specified folder.   
\*\*Input\*\*: Selected email(s) from the archive.   
\*\*Output\*\*: Restored email(s) in the mailbox; logs of the recovery action; confirmation message displayed.  
  
## 1.17 Event Creation Function   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: Users can create calendar events by specifying a title, date, time, and participants. Email invitations are automatically sent to participants.   
\*\*Input\*\*: Event title, date, time, description, and participant list.   
\*\*Output\*\*: Event added to the user’s calendar; invitations sent to participants; confirmation message displayed.  
  
## 1.18 Event Update Function   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: Users can update an existing event, modifying details such as title, date, time, or participants. Updated notifications are sent to participants if configured.   
\*\*Input\*\*: Updated event details and participant list.   
\*\*Output\*\*: Updated event stored in the database; notification sent to participants; confirmation message displayed.  
  
## 1.19 Event Deletion Function   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: Users can delete an event after confirmation. Cancellation notifications are sent to participants if configured.   
\*\*Input\*\*: Selected event to delete.   
\*\*Output\*\*: Event removed from the calendar; cancellation notifications sent; confirmation message displayed.  
  
## 1.20 Calendar View Function   
\*\*Function ID\*\*: FR-20   
\*\*Description\*\*: Users can view their personal or shared calendar in a selected time frame (daily, weekly, monthly).   
\*\*Input\*\*: Time frame selection (daily, weekly, monthly).   
\*\*Output\*\*: Displayed calendar with scheduled events; logs of the view action.  
  
## 1.21 Calendar Sharing Function   
\*\*Function ID\*\*: FR-21   
\*\*Description\*\*: Users or administrators can share a personal calendar with other users, shared accounts, or distribution groups.   
\*\*Input\*\*: Selected calendar and recipients with access permissions.   
\*\*Output\*\*: Shared calendar accessible to recipients; confirmation message displayed.  
  
## 1.22 Task Creation Function   
\*\*Function ID\*\*: FR-22   
\*\*Description\*\*: Users can create tasks with a title, description, due date, and optional calendar event linkage.   
\*\*Input\*\*: Task title, description, due date, and optional event linkage.   
\*\*Output\*\*: Task added to the task list and optionally linked to a calendar event.  
  
## 1.23 Task Update Function   
\*\*Function ID\*\*: FR-23   
\*\*Description\*\*: Users can update task details, such as status, due date, or progress.   
\*\*Input\*\*: Updated task details.   
\*\*Output\*\*: Task updated in the database; confirmation message displayed.  
  
## 1.24 Task Deletion Function   
\*\*Function ID\*\*: FR-24   
\*\*Description\*\*: Users or administrators can delete a task after confirmation.   
\*\*Input\*\*: Selected task to delete.   
\*\*Output\*\*: Task removed from the task list; logs of the deletion action.  
  
## 1.25 Task Progress Viewing Function   
\*\*Function ID\*\*: FR-25   
\*\*Description\*\*: Users can view the progress of a task, including completion percentage and status.   
\*\*Input\*\*: Selected task.   
\*\*Output\*\*: Task progress displayed in a structured format; logs of the view action.  
  
## 1.26 User Permission Management Function   
\*\*Function ID\*\*: FR-26   
\*\*Description\*\*: Administrators can manage user permissions, enabling or restricting access to specific resources or actions.   
\*\*Input\*\*: Selected user and permissions (read, write, manage).   
\*\*Output\*\*: Updated user permissions in the database; confirmation message displayed.  
  
## 1.27 Compliance Report Generation Function   
\*\*Function ID\*\*: FR-27   
\*\*Description\*\*: Administrators can generate compliance reports based on defined criteria, such as user activity, retention policy compliance, and archive status.   
\*\*Input\*\*: Report scope, time frame, and filters.   
\*\*Output\*\*: Compliance report in selected format (PDF, CSV); logs of the action.  
  
## 1.28 Compliance Report Viewing Function   
\*\*Function ID\*\*: FR-28   
\*\*Description\*\*: Administrators can view existing compliance reports, including their metadata and contents.   
\*\*Input\*\*: Selected compliance report.   
\*\*Output\*\*: Compliance report displayed in a structured format; logs of the view action.  
  
## 1.29 Compliance Report Deletion Function   
\*\*Function ID\*\*: FR-29   
\*\*Description\*\*: Administrators can delete compliance reports that are no longer needed and not under a retention policy.   
\*\*Input\*\*: Selected compliance report.   
\*\*Output\*\*: Compliance report removed from the system; logs of the action; confirmation message displayed.  
  
## 1.30 Email Flow Management Function   
\*\*Function ID\*\*: FR-30   
\*\*Description\*\*: Administrators can create, edit, or delete email flow configurations to define how emails are processed or routed.   
\*\*Input\*\*: Email flow configuration details (name, rules, scope).   
\*\*Output\*\*: Updated email flow stored in the database; logs of the action; confirmation message displayed.  
  
## 1.31 Retention Policy Management Function   
\*\*Function ID\*\*: FR-31   
\*\*Description\*\*: Administrators can create, update, or delete retention policies that define the duration for which emails should be retained.   
\*\*Input\*\*: Retention policy details (name, duration, conditions).   
\*\*Output\*\*: Updated or deleted retention policy; logs of the action; confirmation message displayed.  
  
## 1.32 Distribution Group Viewing Function   
\*\*Function ID\*\*: FR-32   
\*\*Description\*\*: Users or administrators can view the details of a distribution group, including name, description, and member list.   
\*\*Input\*\*: Selected distribution group.   
\*\*Output\*\*: Group details displayed in the system; logs of the view action.  
  
## 1.33 System Usage Monitoring Function   
\*\*Function ID\*\*: FR-33   
\*\*Description\*\*: Administrators can monitor system usage to assess email and calendar activity across users and shared accounts.   
\*\*Input\*\*: Time frame and monitoring scope (users, shared accounts, groups).   
\*\*Output\*\*: System usage report generated; logs of the action; confirmation message displayed.

# External Description

# 2. External Interfaces   
  
This chapter defines the external interfaces of the system, including user interfaces, hardware interfaces, software interfaces, and communication interfaces. These interfaces are crucial for the system to interact with users, external systems, and hardware components. The descriptions are based on the inputs, outputs, and external data sources referenced in Chapter 1: Functional Requirements.   
  
## 2.1 User Interface Output   
  
The system interacts with users through a well-designed graphical user interface (GUI) and command-line interface (CLI) for administrative tasks. These interfaces support various functionalities such as email composition, formatting, contact management, calendar views, task creation, and compliance reporting.   
  
- \*\*Email Composition Interface\*\*:   
 A GUI interface that allows users to compose emails with options for formatting text, adding attachments, and selecting recipients from personal or shared contact lists.   
  
- \*\*Inbox and Archive Display Interface\*\*:   
 A GUI interface that displays incoming emails, archived emails, and calendar events. It supports sorting, filtering, and searching based on criteria such as sender, recipient, date, and keywords.   
  
- \*\*Contact and Distribution Group Management Interface\*\*:   
 A GUI interface for adding, editing, deleting, and viewing contacts and distribution groups. It includes validation of contact data and confirmation messages for actions.   
  
- \*\*Task and Calendar Management Interface\*\*:   
 A GUI interface where users can create, update, delete, and view tasks and calendar events. It supports optional linkage between tasks and calendar events.   
  
- \*\*Compliance Report Interface\*\*:   
 A GUI interface for viewing, generating, and deleting compliance reports. It allows administrators to define the scope, time frame, and filters for report generation and provides structured display for viewing reports.   
  
- \*\*User Permission and System Configuration Interface\*\*:   
 A GUI interface accessible only to administrators for managing user permissions, retention policies, email flow configurations, and system settings.   
  
- \*\*System Logs and Notifications Interface\*\*:   
 A GUI interface for viewing logs of user and administrative actions, including email sending, archiving, and deletion activities. Confirmation messages and alerts are also displayed here.   
  
## 2.2 Hardware Interface Output   
  
The system does not directly interact with any specific hardware devices. However, it is designed to run on standard server hardware and client devices such as desktops, laptops, and mobile devices. The following hardware interfaces are relevant:   
  
- \*\*Email Server Hardware\*\*:   
 The system interacts with an email server (e.g., SMTP/IMAP server) to send and receive emails. The hardware must support the necessary email protocols and have sufficient processing power and memory to handle email traffic.   
  
- \*\*Archive Storage Hardware\*\*:   
 The system stores archived emails and compliance reports on a dedicated archive storage system. This hardware must support high-capacity storage and ensure data integrity and compliance with retention policies.   
  
- \*\*Database Server Hardware\*\*:   
 The system uses a database server to store user data, email configurations, retention policies, and system logs. The hardware must be capable of handling concurrent database queries and ensuring data security and performance.   
  
## 2.3 Software Interface Output   
  
The system interacts with various software components, including databases, external APIs, and third-party tools, to perform its functions.   
  
- \*\*Email Server (SMTP/IMAP)\*\*:   
 The system communicates with an email server using the SMTP (Simple Mail Transfer Protocol) for sending emails and IMAP (Internet Message Access Protocol) for receiving emails. The server must be configured to support authentication, encryption, and secure message transfer.   
 - \*\*Input\*\*: Email content, recipient list, and authentication credentials.   
 - \*\*Output\*\*: Sent emails and incoming emails processed and stored in the system.   
  
- \*\*Database System\*\*:   
 The system interacts with a database to store and retrieve data such as user information, contacts, distribution groups, email flow configurations, retention policies, tasks, and compliance reports.   
 - \*\*Input\*\*: Data entries for contacts, groups, policies, tasks, and system logs.   
 - \*\*Output\*\*: Updated or retrieved data entries and confirmation of database actions.   
  
- \*\*Compliance and Archiving Tools\*\*:   
 The system uses compliance tools to enforce retention policies and manage email archives. These tools must support data retention, purging, and backup functionalities.   
 - \*\*Input\*\*: Retention policy details, archive management commands (backup, purge, restore).   
 - \*\*Output\*\*: Compliance logs, updated archive status, and confirmation of actions.   
  
- \*\*Calendar and Task Management Software\*\*:   
 The system integrates with calendar and task management software to support event creation, updates, and task tracking. This integration includes sending email invitations and notifications for event changes.   
 - \*\*Input\*\*: Event details, task details, and user preferences for notifications.   
 - \*\*Output\*\*: Updated events and tasks in the database, and sent email notifications.   
  
- \*\*User Authentication and Authorization System\*\*:   
 The system integrates with an authentication and authorization system to manage user permissions and access rights.   
 - \*\*Input\*\*: User credentials and permission configurations.   
 - \*\*Output\*\*: Updated permission records and confirmation of access changes.   
  
- \*\*Reporting and Analytics Software\*\*:   
 The system interacts with reporting software to generate and display compliance reports.   
 - \*\*Input\*\*: Report filters, time frames, and scopes.   
 - \*\*Output\*\*: Generated reports in PDF or CSV format, and logs of report actions.   
  
## 2.4 Communication Interface Output   
  
The system communicates with external systems and users through various communication channels, including email, web protocols, and internal messaging.   
  
- \*\*Email Communication Interface\*\*:   
 The system sends and receives emails via the configured email server. This includes sending email invitations for calendar events, compliance reports, and user notifications.   
 - \*\*Input\*\*: Email content, recipient list, and email server configuration.   
 - \*\*Output\*\*: Emails delivered to recipient inboxes, and incoming emails processed and stored in the system.   
  
- \*\*Web Communication Interface\*\*:   
 The system supports web-based access through HTTP/HTTPS protocols. Users and administrators can interact with the system via a web browser.   
 - \*\*Input\*\*: User input through web forms and API requests for administrative actions.   
 - \*\*Output\*\*: Web page responses, API responses, and real-time updates in the web interface.   
  
- \*\*Internal Messaging Interface\*\*:   
 The system uses an internal messaging mechanism to notify users of event updates, task changes, and compliance actions.   
 - \*\*Input\*\*: Event/task updates and user preferences for notifications.   
 - \*\*Output\*\*: Notifications sent to users via in-app alerts or email.   
  
- \*\*System-to-System Communication for Email Flow\*\*:   
 The system communicates with itself or external systems to enforce email flow configurations. This includes routing emails to shared accounts, distribution groups, and applying email flow rules.   
 - \*\*Input\*\*: Email content and routing rules.   
 - \*\*Output\*\*: Emails processed according to the defined flow rules, and logs of the routing actions.   
  
- \*\*Archive and Backup Communication Interface\*\*:   
 The system communicates with archive and backup systems to store, retrieve, and purge emails based on retention policies.   
 - \*\*Input\*\*: Email data and retention policy commands.   
 - \*\*Output\*\*: Archived or purged emails, and logs of archive operations.   
  
- \*\*Compliance Report Communication Interface\*\*:   
 The system communicates with external reporting tools and internal compliance systems to generate and export reports.   
 - \*\*Input\*\*: Report configuration parameters and user/administrator requests.   
 - \*\*Output\*\*: Compliance reports in PDF or CSV format, and logs of report actions.   
  
- \*\*User Notification Communication Interface\*\*:   
 The system sends notifications to users via email or in-app messages to inform them of new emails, event changes, task updates, and compliance actions.   
 - \*\*Input\*\*: Notification triggers and user preferences.   
 - \*\*Output\*\*: Notifications delivered to users through selected communication channels.   
  
---  
  
This chapter has summarized all the external interfaces that the system will interact with, ensuring a clear understanding of how the system communicates with users, hardware, software, and external systems. Developers can use this information to design and implement the necessary integrations and interactions for the system.

# Use Case

Use Case Name: Send Email   
Use Case ID: UC-01   
Actors: User, Administrator, Email System, Distribution Group, Shared Account, Email Account   
Preconditions:   
1. The user must be authenticated and have access to the email system.   
2. The user must have a valid email account or shared account configured.   
3. The email content, recipients, and subject must be provided.   
4. The system must be connected to an email server and able to send emails.   
  
Postconditions:   
1. The email is successfully sent to the specified recipients.   
2. The email is recorded in the system's email archive.   
3. If applicable, the email is added to a distribution group or shared account.   
4. The user receives a confirmation message indicating the email was sent.   
  
Main Flow:   
1. The user opens the email system and navigates to the "Compose" option.   
2. The user enters the subject and body of the email.   
3. The user selects recipients from the contact list or manually enters their email addresses.   
4. The user verifies the email content and recipient list.   
5. The user clicks the "Send" button.   
6. The email system validates the email and recipient information.   
7. The email is sent via the configured email server.   
8. The system logs the email in the archive and updates the user interface.   
9. A confirmation message is displayed to the user.   
  
Alternative Flow:   
1. If the email system fails to send the email, it displays an error message.   
2. If the recipient list contains invalid email addresses, the system alerts the user and allows them to correct the entries.   
3. If the user cancels the send operation before the email is sent, the system discards the draft and returns to the inbox.   
4. If the user chooses to save the email as a draft, the system stores it in the draft folder for later use.   
5. If the email exceeds the system's size limit, the user is prompted to reduce the size or split the email.  
  
Use Case Name: Receive Email   
Use Case ID: UC-02   
Actors: User, Email System, Email Account, Administrator, Shared Account, Distribution Group   
Preconditions:   
1. The email system is connected to an email server and configured to receive emails.   
2. The user has an active email account or is using a shared account.   
3. The system has valid inbox and notification settings enabled.   
4. The email has been sent by a sender and is in the process of being delivered.   
  
Postconditions:   
1. The incoming email is successfully received and displayed in the user's inbox.   
2. The email is logged in the system's archive.   
3. The user is notified of the new email, either through an in-app alert or email notification.   
4. If the email is part of a distribution group, all members are notified accordingly.   
  
Main Flow:   
1. The email system checks the configured email server for new incoming emails.   
2. A new email arrives from a sender and is processed by the email system.   
3. The system validates the email and ensures it is not blocked or flagged as spam.   
4. The email is stored in the user's inbox or the shared account's inbox if applicable.   
5. The system updates the user interface to show the new email.   
6. The user is notified of the new email via a visual or auditory alert.   
7. The user opens the inbox and views the received email.   
8. The email is logged in the archive for retention and recovery purposes.   
  
Alternative Flow:   
1. If the email is flagged as spam or contains malicious content, it is moved to the spam folder or quarantined.   
2. If the user is not logged in, the notification is queued and delivered upon login.   
3. If the system is unable to receive the email due to server issues, an error message is logged and the user is notified.   
4. If the user has configured filters or rules, the email is automatically sorted into the appropriate folder.   
5. If the email is received but the user chooses to delete it without reading, it is moved to the trash folder.  
  
Use Case Name: Format Email   
Use Case ID: UC-03   
Actors: User, Email System, Email Account, Administrator, Shared Account   
Preconditions:   
1. The user must be authenticated and logged into the email system.   
2. The user must have an email account or be using a shared account.   
3. The email content must be either drafted or composed.   
4. The user must select the option to format the email before sending.   
  
Postconditions:   
1. The email is properly formatted with the specified settings (font, size, color, etc.).   
2. The formatting changes are saved in the draft or email content.   
3. The user receives a visual confirmation of the applied formatting.   
4. If the formatting exceeds system limits or causes errors, the user is notified and given corrective options.   
  
Main Flow:   
1. The user opens the email system and composes or selects a draft email.   
2. The user clicks on the "Format" button to open the formatting options.   
3. The system displays available formatting tools (e.g., font style, size, bold, italic, underline).   
4. The user selects the desired formatting options and applies them to the email content.   
5. The system updates the email with the applied formatting.   
6. The user reviews the formatted email for accuracy and completeness.   
7. The user saves the formatted email as a draft or proceeds to send it.   
  
Alternative Flow:   
1. If the user selects an unsupported font or formatting style, the system displays an error and suggests alternatives.   
2. If the formatting causes the email to exceed the system's character or file size limit, the user is prompted to adjust the formatting.   
3. If the user cancels the formatting process, the email reverts to its previous state.   
4. If the formatting option is not available due to system restrictions or account type, the user is informed and redirected to another action.   
5. If the user does not apply any formatting and chooses to send the email, the system proceeds to the send process without changes.  
  
Use Case Name: Manage Contacts   
Use Case ID: UC-04   
Actors: User, Administrator, Contact System, Email System, Shared Account, Distribution Group   
Preconditions:   
1. The user must be authenticated and have access to the contact system.   
2. The user must have an associated email account or shared account.   
3. The contact system must be connected to the email system to enable synchronization.   
4. The user must have appropriate permissions to add, edit, or delete contacts.   
  
Postconditions:   
1. The contact list is updated with the changes made (add, edit, or delete).   
2. If the contact is linked to an email or distribution group, the changes are reflected in the email system.   
3. The user receives a confirmation message for each contact management action performed.   
4. The system logs the contact management actions for audit and recovery purposes.   
  
Main Flow:   
1. The user navigates to the contact management section of the email system.   
2. The user selects an action: "Add Contact," "Edit Contact," or "Delete Contact."   
3. The system displays the appropriate form or interface for the selected action.   
4. The user fills in or modifies the contact details (name, email, phone, etc.).   
5. The user submits the changes.   
6. The system validates the contact information and checks for duplicates if applicable.   
7. The system updates the contact list in the database.   
8. If the contact is associated with an email or distribution group, the system updates the relevant configurations.   
9. A confirmation message is displayed to the user indicating the success of the action.   
  
Alternative Flow:   
1. If the user enters invalid contact information, the system displays an error and prompts for correction.   
2. If a duplicate contact is detected during the add process, the system asks the user if they want to merge or cancel the action.   
3. If the user cancels the contact management action, the system discards the changes and returns to the contact list.   
4. If the contact is part of a distribution group and the user deletes it, the system removes the contact from the group and confirms the change.   
5. If the system is unable to update the contact due to a database or server error, the user is notified, and the action is rolled back.  
  
Use Case Name: Create Distribution Group   
Use Case ID: UC-05   
Actors: User, Administrator, Contact System, Distribution Group, Email System, Shared Account   
  
Preconditions:   
1. The user must be authenticated and have the necessary permissions to create a distribution group.   
2. The user must have access to the contact system to select or add members.   
3. The system must be connected to the database to store the new distribution group.   
4. The user must provide a valid name and description for the distribution group.   
  
Postconditions:   
1. A new distribution group is created and stored in the system.   
2. The group members are added to the distribution list.   
3. The distribution group is available for use in email sending operations.   
4. The system logs the creation of the distribution group for audit and recovery purposes.   
  
Main Flow:   
1. The user navigates to the "Distribution Groups" section in the email system.   
2. The user selects the "Create Distribution Group" option.   
3. The system prompts the user to enter a name and description for the new group.   
4. The user inputs the name and description and clicks "Next."   
5. The user selects existing contacts from the contact list or manually adds new email addresses.   
6. The system validates the selected contacts and email addresses.   
7. The user confirms the member list and clicks "Create."   
8. The system creates the distribution group and stores it in the database.   
9. A confirmation message is displayed to the user.   
  
Alternative Flow:   
1. If the user enters an invalid name (e.g., duplicate or empty), the system displays an error and prompts for a valid name.   
2. If the user adds an invalid email address, the system alerts the user and allows them to correct or remove the entry.   
3. If the user cancels the creation process, the system discards the group and returns to the distribution group list.   
4. If the system fails to create the group due to database or server issues, the user is notified, and the action is rolled back.   
5. If the user selects a shared account as a member, the system verifies the shared account's validity before adding it to the group.  
  
Use Case Name: Delete Distribution Group   
Use Case ID: UC-06   
Actors: User, Administrator, Distribution Group, Contact System, Email System, Shared Account   
  
Preconditions:   
1. The user or administrator must be authenticated and have the necessary permissions to delete a distribution group.   
2. The distribution group must exist in the system and be associated with the user or shared account.   
3. The system must be connected to the database to update or remove the group.   
4. The user must select the specific distribution group to be deleted.   
  
Postconditions:   
1. The selected distribution group is removed from the system.   
2. The group's members are no longer associated with it in the contact system.   
3. All references to the deleted group in the email system are removed or updated.   
4. The system logs the deletion of the distribution group for audit and recovery purposes.   
5. The user receives a confirmation message indicating the deletion was successful.   
  
Main Flow:   
1. The user or administrator navigates to the "Distribution Groups" section in the email system.   
2. The system displays a list of existing distribution groups.   
3. The user selects the distribution group to be deleted.   
4. The system prompts the user for confirmation before proceeding.   
5. The user confirms the deletion.   
6. The system removes the distribution group from the database.   
7. The system updates the contact list and any associated shared accounts to remove the group's references.   
8. The system logs the deletion event.   
9. A confirmation message is displayed to the user.   
  
Alternative Flow:   
1. If the user selects an invalid or non-existent distribution group, the system displays an error message.   
2. If the user cancels the deletion after selecting the group, the system retains the group and returns to the list.   
3. If the system is unable to delete the group due to database or server issues, an error message is displayed, and the deletion is rolled back.   
4. If the group is being used in an active email flow or retention policy, the system alerts the user and prevents deletion until the dependencies are resolved.   
5. If the administrator is performing the deletion, the system may log additional audit details for compliance tracking.  
  
Use Case Name: Update Distribution Group   
Use Case ID: UC-07   
Actors: User, Administrator, Distribution Group, Contact System, Email System, Shared Account   
  
Preconditions:   
1. The user or administrator must be authenticated and have the necessary permissions to update a distribution group.   
2. The distribution group must already exist in the system and be associated with the user or shared account.   
3. The system must be connected to the database to modify the group's information.   
4. The user must select the specific distribution group to be updated.   
  
Postconditions:   
1. The selected distribution group is updated with the new information (name, description, or member list).   
2. The contact system and email system reflect the updated distribution group configuration.   
3. The system logs the update of the distribution group for audit and recovery purposes.   
4. The user receives a confirmation message indicating the update was successful.   
  
Main Flow:   
1. The user or administrator navigates to the "Distribution Groups" section in the email system.   
2. The system displays a list of existing distribution groups.   
3. The user selects the distribution group they wish to update.   
4. The system opens the group's details for editing.   
5. The user modifies the group's name, description, or member list as needed.   
6. The system validates the updated information (e.g., checking for duplicate names, valid email addresses).   
7. The user confirms the changes and clicks "Update."   
8. The system updates the distribution group in the database.   
9. A confirmation message is displayed to the user.   
  
Alternative Flow:   
1. If the user enters an invalid or duplicate name for the distribution group, the system displays an error and prompts for a valid name.   
2. If the user adds an invalid email address to the member list, the system alerts them and allows for correction or removal.   
3. If the user cancels the update after making changes, the system reverts to the original group settings and returns to the group list.   
4. If the system fails to update the group due to database or server issues, the user is notified, and the action is rolled back.   
5. If the group is referenced in an active email flow or retention policy, the system alerts the user before updating to ensure no unintended consequences.  
  
Use Case Name: Manage Email Accounts   
Use Case ID: UC-08   
Actors: User, Administrator, Email System, Email Account, Shared Account   
Preconditions:   
1. The user or administrator must be authenticated and have the necessary permissions to manage email accounts.   
2. The email system must be connected to the database and email server to perform account-related operations.   
3. The user or administrator must select an existing email account or initiate the creation of a new one.   
4. For shared accounts, the user must have access rights or be an administrator.   
  
Postconditions:   
1. The selected email account is either created, updated, or deleted successfully.   
2. If the account is shared, the permissions and access rights are updated accordingly.   
3. The system logs all account management actions for audit and recovery purposes.   
4. The user or administrator receives a confirmation message for the performed action.   
  
Main Flow:   
1. The user or administrator navigates to the "Email Accounts" section in the email system.   
2. The system displays a list of existing email accounts, including personal and shared accounts.   
3. The user selects an action: "Create New Account," "Edit Account," or "Delete Account."   
4. If creating a new account, the system prompts for email address, password, and account type (personal or shared).   
5. If editing or deleting an account, the user selects the specific account and confirms the action.   
6. The system validates the input and checks for any dependencies (e.g., active emails, retention policies).   
7. The system performs the requested action (create, update, or delete) in the database and email server.   
8. A confirmation message is displayed to the user or administrator.   
  
Alternative Flow:   
1. If the user enters an invalid email address or password, the system displays an error and prompts for correction.   
2. If a duplicate email account is detected during creation, the system alerts the user and suggests alternatives.   
3. If the user or administrator cancels the action, the system reverts to the account list and discards changes.   
4. If the system fails to perform the action due to database or server issues, an error message is displayed, and the action is rolled back.   
5. If the selected account is part of an active email flow or retention policy, the system warns the user and prevents deletion until dependencies are resolved.  
  
Use Case Name: Manage Shared Accounts   
Use Case ID: UC-09   
Actors: Administrator, User, Email System, Shared Account, Email Account, Distribution Group   
  
Preconditions:   
1. The administrator or user must be authenticated and have the necessary permissions to manage shared accounts.   
2. The email system must be connected to the database and email server to perform shared account operations.   
3. A shared account must exist in the system, or the user must provide valid information to create a new one.   
4. The administrator must have access to the shared account management interface.   
  
Postconditions:   
1. The shared account is either created, updated, or deleted as per the requested action.   
2. Access rights and permissions for the shared account are updated in the system.   
3. The system logs the shared account management action for audit and recovery purposes.   
4. The user or administrator receives a confirmation message indicating the success of the action.   
5. If the shared account is part of an email flow or distribution group, the changes are reflected in those configurations.   
  
Main Flow:   
1. The administrator navigates to the "Shared Accounts" section in the email system.   
2. The system displays a list of existing shared accounts, including their access rights and usage status.   
3. The administrator selects an action: "Create Shared Account," "Edit Shared Account," or "Delete Shared Account."   
4. If creating a new shared account, the system prompts for the email address, password, and access permissions.   
5. The administrator fills in the required details and assigns users or groups with access rights.   
6. The system validates the input (e.g., unique email address, valid permissions).   
7. The shared account is created or updated in the database and email server.   
8. The system updates the access rights for the assigned users or groups.   
9. A confirmation message is displayed to the administrator.   
  
Alternative Flow:   
1. If the administrator enters an invalid email address or password, the system displays an error and prompts for correction.   
2. If a duplicate shared account is detected during creation, the system alerts the administrator and suggests alternatives.   
3. If the administrator cancels the action at any point, the system reverts to the shared account list and discards any changes.   
4. If the system fails to create or update the shared account due to database or server issues, an error message is displayed, and the action is rolled back.   
5. If the shared account is referenced in an active email flow, retention policy, or distribution group, the system warns the administrator before deletion to avoid unintended consequences.  
  
Use Case Name: Archive Emails   
Use Case ID: UC-10   
Actors: User, Administrator, Email System, Email Account, Shared Account, Distribution Group, Email Flow, Retention Policy, Archive, Recovery   
  
Preconditions:   
1. The user must be authenticated and have access to the email system.   
2. The email to be archived must exist in the user’s inbox, sent items, or another accessible folder.   
3. The system must be configured with an archive module and retention policies.   
4. The user must have the necessary permissions to archive emails or be part of a group with archiving privileges.   
5. The archive database or storage system must be available and functional.   
  
Postconditions:   
1. The selected email is moved to the archive and no longer visible in the active inbox or folders.   
2. The archive system records the email with metadata (sender, recipient, timestamp, etc.).   
3. If the email is governed by a retention policy, the system ensures it is stored for the required duration.   
4. The user receives a confirmation message that the email has been archived.   
5. The email remains recoverable through the recovery process if needed.   
  
Main Flow:   
1. The user logs into the email system and navigates to the inbox or another folder containing the email to be archived.   
2. The user selects the email and clicks the "Archive" option.   
3. The system checks if the email is eligible for archiving (e.g., not locked by a retention policy).   
4. The system moves the email to the archive database or storage system.   
5. The system updates the user interface to reflect the removal of the email from the active folder.   
6. The system logs the archiving action for audit and recovery purposes.   
7. A confirmation message is displayed to the user indicating the email has been successfully archived.   
  
Alternative Flow:   
1. If the selected email is locked due to an active retention policy, the system displays a warning and prevents archiving until the policy allows it.   
2. If the user selects multiple emails for archiving, the system archives all selected emails and provides a summary confirmation.   
3. If the archive system is unavailable or encounters an error, the system displays an error message and rolls back the action.   
4. If the user cancels the archiving operation after selecting the email, the system reverts the selection and leaves the email in its original folder.   
5. If the administrator archives an email on behalf of a user, the system logs the action under the administrator’s account and notifies the user.  
  
Use Case Name: Search Personal Archives   
Use Case ID: UC-11   
Actors: User, Administrator, Email System, Email Account, Archive, Email Flow, Retention Policy   
  
Preconditions:   
1. The user must be authenticated and have access to the email system and archive module.   
2. The archive must contain emails associated with the user or their email account.   
3. The system must be connected to the archive database and have a search functionality enabled.   
4. The user must initiate a search request.   
  
Postconditions:   
1. The user receives a list of emails that match the search criteria.   
2. The search results are displayed with relevant metadata (subject, date, sender, recipient, etc.).   
3. The system logs the search action for audit and recovery purposes.   
4. If the user selects an email, it can be viewed or restored to an active folder.   
5. Emails governed by retention policies are not deleted during the search process.   
  
Main Flow:   
1. The user logs into the email system and navigates to the "Archive" or "Search Archives" section.   
2. The user enters search criteria such as keywords, date range, sender, or recipient.   
3. The system processes the search request and queries the archive database.   
4. The system displays a list of matching archived emails with their metadata.   
5. The user selects an email to view or restore it to an active folder.   
6. The system updates the interface to show the selected email or moves it to the specified folder.   
7. A confirmation message is displayed to the user indicating the success of the search and any actions taken.   
  
Alternative Flow:   
1. If no emails match the search criteria, the system displays a message indicating that no results were found.   
2. If the system is unable to access the archive due to technical issues, an error message is displayed, and the user is informed.   
3. If the user selects an email that is governed by a retention policy, the system prevents deletion and displays a warning.   
4. If the user cancels the search or restoration process, the system reverts to the previous screen and discards the changes.   
5. If the search query is invalid or incomplete, the system prompts the user to refine their search criteria.  
  
Use Case Name: Server Archive Management   
Use Case ID: UC-12   
Actors: Administrator, Email System, Archive, Retention Policy, Shared Account, Email Account, Recovery   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to manage the server archive.   
2. The email system must be connected to the archive database and have active retention policies in place.   
3. The archive system must be available and functional for the operation to proceed.   
4. The administrator must initiate the archive management process via the system’s administrative interface.   
  
Postconditions:   
1. The server archive is updated according to the administrator’s actions (e.g., backup, purge, or restore).   
2. Emails removed or modified in the archive are logged for audit and compliance.   
3. The system ensures that retention policies are enforced and no emails are deleted before their retention period ends.   
4. The administrator receives a confirmation message for each action performed.   
5. If a restore is performed, the email is made available in the user’s mailbox or shared account.   
  
Main Flow:   
1. The administrator logs into the email system and navigates to the "Server Archive Management" interface.   
2. The system displays options such as "Backup Archive," "Purge Expired Emails," or "Restore Archived Emails."   
3. The administrator selects the desired action and confirms the operation.   
4. The system verifies the action against active retention policies and archive rules.   
5. The system performs the selected action (e.g., backing up the archive, purging expired emails, restoring specific emails).   
6. The system logs the operation details, including the affected emails and the administrator who performed the action.   
7. The system updates the archive structure and ensures the integrity of the data.   
8. A confirmation message is displayed to the administrator indicating the success of the operation.   
  
Alternative Flow:   
1. If the administrator attempts to purge an email that is still under a retention policy, the system displays a warning and prevents the action.   
2. If the archive system is unavailable or encounters an error, the system logs the failure and displays an error message to the administrator.   
3. If the administrator cancels the operation before it is completed, the system rolls back any changes and returns to the management interface.   
4. If the selected action requires additional approval (e.g., bulk purge or restore), the system requests confirmation from another administrator before proceeding.   
5. If the system detects inconsistencies in the archive (e.g., corrupted data), it alerts the administrator and provides recovery options.  
  
Use Case Name: Capture Email Flow   
Use Case ID: UC-13   
Actors: Administrator, Email System, Email Account, Shared Account, Distribution Group, Email Flow, Retention Policy, Archive, Recovery   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to monitor or configure email flow.   
2. The email system must be connected to the email server and archive for tracking and recording purposes.   
3. Email flow monitoring must be enabled in the system configuration.   
4. The system must have access to retention policies and distribution groups to support flow tracking.   
  
Postconditions:   
1. The system captures and records the complete email flow (e.g., sending, receiving, archiving).   
2. All actions related to the email flow (e.g., routing, retention, deletion) are logged for audit and compliance.   
3. The administrator receives a summary or report of the captured email flow.   
4. Emails governed by retention policies are not prematurely removed during the flow.   
5. If recovery is needed, the system ensures the flow data remains accessible for restoration.   
  
Main Flow:   
1. The administrator logs into the email system and accesses the "Email Flow Monitoring" section.   
2. The system displays options to capture or view existing email flows.   
3. The administrator selects the option to "Capture Email Flow" and specifies the scope (e.g., specific user, shared account, distribution group).   
4. The system activates the flow capture mechanism for the specified scope.   
5. As emails are sent or received within the scope, the system logs the flow details (sender, recipient, timestamp, route, status).   
6. The system checks if the email meets any retention policies or archive criteria.   
7. The system updates the archive with the email and flow information as needed.   
8. The administrator can view or export the captured email flow data for analysis or compliance.   
9. The system displays a confirmation that the email flow was captured and stored.   
  
Alternative Flow:   
1. If the administrator selects an invalid scope, the system displays an error and prompts for a valid selection.   
2. If the system is unable to capture the flow due to configuration issues or server unavailability, an error message is displayed, and the operation is rolled back.   
3. If the administrator cancels the flow capture before it is completed, the system stops tracking and returns to the monitoring interface.   
4. If an email is flagged for retention during the flow, the system prevents it from being archived or deleted and logs the retention action.   
5. If the system detects an error in the flow (e.g., failed delivery), it logs the error and notifies the administrator for review.  
  
Use Case Name: Apply Retention Policy   
Use Case ID: UC-14   
Actors: Administrator, Email System, Email Account, Shared Account, Distribution Group, Retention Policy, Archive, Recovery   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to apply retention policies.   
2. The email system must be connected to the archive and have active retention policies configured.   
3. The email or set of emails to which the policy will be applied must exist in the system.   
4. The selected retention policy must be valid and defined by the system.   
5. The system must be able to enforce the policy without conflicting with existing configurations.   
  
Postconditions:   
1. The selected retention policy is applied to the specified emails or email accounts.   
2. The system enforces the policy, ensuring emails are retained for the specified period.   
3. Emails affected by the policy are marked in the archive with retention metadata.   
4. The system logs the application of the retention policy for audit and compliance.   
5. The administrator receives a confirmation message that the policy was successfully applied.   
  
Main Flow:   
1. The administrator logs into the email system and navigates to the "Retention Policies" section.   
2. The system displays a list of available retention policies and the emails or accounts to which they can be applied.   
3. The administrator selects a retention policy and chooses the target emails or email accounts.   
4. The system validates the selection and confirms the policy’s duration and scope.   
5. The administrator confirms the application of the policy.   
6. The system applies the policy to the selected emails or accounts and updates their metadata.   
7. The system logs the action and updates the archive or database to enforce the policy.   
8. A confirmation message is displayed to the administrator indicating the success of the operation.   
  
Alternative Flow:   
1. If the administrator selects an invalid or non-existent retention policy, the system displays an error and prompts for a valid selection.   
2. If the selected emails are already under another retention policy, the system alerts the administrator and asks whether to overwrite or merge policies.   
3. If the system is unable to apply the policy due to archive or server issues, an error message is displayed, and the action is rolled back.   
4. If the administrator cancels the operation after selecting the policy, the system reverts to the previous screen and does not apply the policy.   
5. If the policy application affects shared accounts or distribution groups, the system checks for user permissions and may require additional approvals.  
  
Use Case Name: Manage Expired Emails   
Use Case ID: UC-15   
Actors: User, Administrator, Email System, Email Account, Shared Account, Distribution Group, Archive, Recovery, Retention Policy, Expired Email   
  
Preconditions:   
1. The user or administrator must be authenticated and have the necessary permissions to manage expired emails.   
2. The system must have an active retention policy that defines the expiration criteria for emails.   
3. The email system must be connected to the archive and email server to process expired emails.   
4. Expired emails must be identified based on the retention policy rules.   
5. The system must have the capability to move or delete expired emails.   
  
Postconditions:   
1. Expired emails are either moved to the archive or deleted from the system according to the configured policy.   
2. The system logs the management of expired emails for audit and recovery purposes.   
3. The user or administrator is notified of the action taken on expired emails.   
4. If emails are archived, they remain recoverable through the recovery process.   
5. The system ensures that no emails are deleted before the retention period is complete.   
  
Main Flow:   
1. The email system automatically identifies emails that have exceeded their retention period as defined by the retention policy.   
2. The system categorizes these emails as "Expired" and prepares them for management.   
3. The administrator or user initiates the "Manage Expired Emails" process from the system interface.   
4. The system displays a list of expired emails, including details such as subject, sender, date, and folder.   
5. The user selects an action: "Archive Expired Emails" or "Delete Expired Emails."   
6. The system validates the action against the current retention policy and system rules.   
7. The system performs the selected action (archive or delete) on the expired emails.   
8. If archived, the system moves the emails to the archive and updates the user interface.   
9. If deleted, the system removes the emails from the system and logs the deletion.   
10. A confirmation message is displayed to the user or administrator indicating the success of the action.   
  
Alternative Flow:   
1. If the system is unable to access the archive or server, an error message is displayed, and the action is rolled back.   
2. If the user or administrator cancels the action after selecting expired emails, the system reverts to the previous screen and leaves the emails untouched.   
3. If an expired email is still referenced in an active email flow or distribution group, the system alerts the user and prevents deletion until the dependency is resolved.   
4. If the system detects an error during the archiving or deletion process, it logs the error and notifies the administrator for review.   
5. If the retention policy is updated during the process, the system re-evaluates the expiration status of the emails and adjusts the action accordingly.  
  
Use Case Name: Recover Emails, Manage Calendar   
Use Case ID: UC-16   
Actors: User, Administrator, Email System, Email Account, Shared Account, Archive, Recovery, Calendar System   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the email and calendar systems.   
2. The email system must be connected to the archive and recovery module.   
3. The calendar system must be accessible and configured for synchronization with the email system.   
4. The user must identify emails or calendar events that require recovery.   
5. The user must have the necessary permissions to recover emails or manage calendar entries.   
  
Postconditions:   
1. The selected emails are successfully recovered from the archive and restored to the user's mailbox or shared account.   
2. The calendar entries are updated or synchronized based on the user's actions.   
3. The system logs all recovery and calendar management actions for audit and compliance.   
4. The user receives a confirmation message for each successful recovery and calendar action.   
5. The system ensures that recovered emails do not conflict with existing retention policies or distribution group settings.   
  
Main Flow:   
1. The user or administrator logs into the email system and navigates to the "Recover Emails" or "Calendar Management" section.   
2. The system displays a list of archived emails or calendar events associated with the user or shared account.   
3. The user selects specific emails or calendar entries for recovery or modification.   
4. The system verifies the eligibility of the selected items (e.g., not locked by a retention policy).   
5. The system initiates the recovery process by restoring the selected emails to the user’s inbox or a specified folder.   
6. If calendar events are involved, the system updates or synchronizes the calendar with the recovered email data.   
7. The system logs the recovery and calendar management actions.   
8. A confirmation message is displayed to the user or administrator, indicating the success of the operation.   
  
Alternative Flow:   
1. If the selected email is governed by an active retention policy, the system displays a warning and prevents recovery until the policy allows it.   
2. If the user selects an invalid or non-existent email or calendar event, the system displays an error and prompts for a valid selection.   
3. If the recovery process encounters an error (e.g., database failure, corrupted data), the system logs the issue and displays an error message to the user.   
4. If the user cancels the recovery or calendar management action, the system reverts to the previous screen and discards any changes.   
5. If the system is unable to synchronize calendar events with the recovered emails, it alerts the user and offers manual correction options.  
  
Use Case Name: Create Event   
Use Case ID: UC-17   
Actors: User, Administrator, Calendar System, Email System, Contact, Distribution Group, Shared Account   
  
Preconditions:   
1. The user must be authenticated and have access to the calendar system.   
2. The calendar system must be connected to the email system for event notifications.   
3. The user must provide event details (title, date, time, participants, and description).   
4. The system must have valid contact and distribution group configurations for sending invitations.   
  
Postconditions:   
1. The event is successfully created in the calendar system.   
2. Invitations are sent to the specified participants via email.   
3. The event is synchronized with relevant calendars and shared accounts if applicable.   
4. The system logs the event creation for audit and recovery purposes.   
5. The user receives a confirmation message indicating the event was created.   
  
Main Flow:   
1. The user logs into the email and calendar system and navigates to the calendar section.   
2. The user selects the "Create Event" option and fills in the event title, date, time, and description.   
3. The user selects participants from the contact list or manually enters their email addresses.   
4. The system validates the selected contacts and ensures they are active and accessible.   
5. The user confirms the event details and clicks "Create."   
6. The system creates the event in the calendar and assigns it to the user or shared calendar.   
7. The system generates and sends email invitations to all selected participants using the email system.   
8. The system logs the event creation and related email actions for audit and compliance.   
9. A confirmation message is displayed to the user, indicating the event was successfully created.   
  
Alternative Flow:   
1. If the user enters an invalid date or time, the system displays an error and prompts for correction.   
2. If the selected participants include invalid email addresses, the system alerts the user and allows for corrections or removals.   
3. If the user cancels the event creation before sending invitations, the system discards the draft and returns to the calendar view.   
4. If the email system is unavailable during invitation sending, the system logs the failure and displays an error to the user.   
5. If the event is part of a shared calendar and the user is not authorized, the system denies the action and displays a permission error.  
  
Use Case Name: Update Event   
Use Case ID: UC-18   
Actors: User, Administrator, Calendar System, Email System, Contact, Distribution Group, Shared Account   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the calendar system.   
2. The event to be updated must already exist in the system.   
3. The calendar system must be connected to the email system for sending updates or notifications.   
4. The user must select the event they wish to update.   
5. The system must have valid contact and distribution group configurations for sending notifications if needed.   
  
Postconditions:   
1. The selected event is updated with the new information (title, date, time, participants, etc.).   
2. If the event includes participants, updated notifications are sent to all relevant attendees via email.   
3. The event is synchronized with any associated shared calendars or distribution groups.   
4. The system logs the event update for audit and recovery purposes.   
5. The user receives a confirmation message indicating the event was successfully updated.   
  
Main Flow:   
1. The user logs into the email and calendar system and navigates to the calendar section.   
2. The user selects an existing event from the calendar view.   
3. The system displays the event details in an editable format.   
4. The user modifies one or more fields (e.g., title, date, time, participants, description).   
5. The user clicks the "Save" or "Update" button.   
6. The system validates the updated event details (e.g., valid date/time, active participants).   
7. The system updates the event in the calendar database.   
8. If participants are updated, the system generates and sends email notifications to all affected attendees.   
9. The system logs the update and displays a confirmation message to the user.   
  
Alternative Flow:   
1. If the user enters an invalid date or time, the system displays an error and prompts for correction.   
2. If the updated event includes invalid or inactive participants, the system alerts the user and allows for removal or correction.   
3. If the user cancels the update after making changes, the system reverts to the previous event details and returns to the calendar view.   
4. If the email system is unavailable when sending notifications, the system logs the failure and displays an error message to the user.   
5. If the event is part of a shared calendar and the user lacks the necessary permissions, the system denies the update and displays a permission error.  
  
Use Case Name: Delete Event   
Use Case ID: UC-19   
Actors: User, Administrator, Calendar System, Email System, Contact, Distribution Group, Shared Account   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the calendar system.   
2. The event to be deleted must exist in the calendar system.   
3. The system must be connected to the email system to notify participants if required.   
4. The user must select the event they wish to delete.   
5. The system must ensure that no active retention policies prevent the deletion of related email notifications.   
  
Postconditions:   
1. The selected event is removed from the calendar system.   
2. If the event includes participants, cancellation notifications are sent to all relevant attendees via email.   
3. The system logs the event deletion for audit and recovery purposes.   
4. The calendar view is updated to reflect the deletion.   
5. The user receives a confirmation message indicating the event was successfully deleted.   
  
Main Flow:   
1. The user logs into the email and calendar system and navigates to the calendar section.   
2. The user selects an existing event from the calendar view.   
3. The system displays the event details and a "Delete" option.   
4. The user clicks the "Delete" button.   
5. The system prompts the user to confirm the deletion.   
6. The user confirms the deletion.   
7. The system removes the event from the calendar database.   
8. If the event had participants, the system generates and sends email notifications to inform them of the cancellation.   
9. The system updates the calendar view and logs the deletion action.   
10. A confirmation message is displayed to the user indicating the event was successfully deleted.   
  
Alternative Flow:   
1. If the user selects an invalid or non-existent event, the system displays an error and prompts for a valid selection.   
2. If the system is unable to send cancellation emails due to email system unavailability, the system logs the failure and displays an error message to the user.   
3. If the user cancels the deletion after selecting the event, the system reverts to the calendar view and retains the event.   
4. If the event is part of a shared calendar and the user lacks the necessary permissions, the system denies the deletion and displays a permission error.   
5. If the event is governed by a retention policy that locks it from deletion, the system displays a warning and prevents the action until the policy allows it.  
  
Use Case Name: View Schedule   
Use Case ID: UC-20   
Actors: User, Administrator, Calendar System, Email System, Contact, Distribution Group, Shared Account   
  
Preconditions:   
1. The user must be authenticated and have access to the calendar system.   
2. The calendar system must be connected to the email system for synchronization of event data.   
3. The user must have at least one calendar (personal or shared) configured in the system.   
4. The calendar must contain events or be empty, depending on the user's schedule.   
  
Postconditions:   
1. The user's calendar is displayed with all scheduled events.   
2. If the calendar is shared, the view reflects the shared event data and permissions.   
3. The system logs the calendar access for audit and compliance.   
4. The user receives a visual confirmation that the schedule has been successfully loaded.   
5. The system ensures that events governed by retention policies are not removed during the viewing process.   
  
Main Flow:   
1. The user logs into the email and calendar system and navigates to the calendar section.   
2. The system loads the user's calendar view, including personal and shared calendars if applicable.   
3. The system displays all scheduled events in the selected time frame (e.g., daily, weekly, monthly view).   
4. The user selects an event to view its details (title, date, time, participants, description).   
5. The system retrieves and displays the event details, including any related email notifications or updates.   
6. The user can filter or search for specific events using keywords or date ranges.   
7. The system updates the interface based on the user's filters or search criteria.   
8. The system logs the user’s access and any filters applied for audit purposes.   
9. A confirmation message is displayed to the user, indicating the schedule is ready for viewing.   
  
Alternative Flow:   
1. If the system fails to load the calendar due to server or database issues, an error message is displayed, and the user is informed.   
2. If the user has no events in the selected time frame, the system displays a message indicating the calendar is empty.   
3. If the user selects a shared calendar and lacks permission, the system denies access and displays a permission error.   
4. If the user attempts to view an event governed by a retention policy that restricts access, the system displays a warning and prevents viewing.   
5. If the user cancels the viewing process or navigates away, the system reverts to the main calendar interface and retains no unsaved changes.  
  
Use Case Name: Share Calendar   
Use Case ID: UC-21   
Actors: User, Administrator, Calendar System, Email System, Shared Account, Contact, Distribution Group   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the calendar system.   
2. The calendar system must be connected to the email system to enable sharing and notifications.   
3. The user must have a calendar (personal or shared) to be shared with others.   
4. The recipient must have a valid email account, shared account, or be part of a distribution group.   
5. The system must have proper permissions and access rights configured for the sharing operation.   
  
Postconditions:   
1. The selected calendar is successfully shared with the specified recipients.   
2. Recipients receive access to the shared calendar and are notified via email if configured.   
3. The system logs the sharing action for audit and compliance tracking.   
4. The shared calendar is synchronized with the recipient's calendar system.   
5. The user receives a confirmation message indicating the calendar was successfully shared.   
  
Main Flow:   
1. The user logs into the email and calendar system and navigates to the calendar section.   
2. The user selects the calendar they wish to share (personal or shared).   
3. The system displays options to share the calendar with specific contacts, email accounts, or distribution groups.   
4. The user selects the recipients and configures access permissions (e.g., view-only, edit).   
5. The user clicks the "Share" button.   
6. The system validates the selected recipients and permissions.   
7. The system grants access to the shared calendar for the specified users or groups.   
8. If notifications are enabled, the system generates and sends an email to the recipients informing them of the shared calendar.   
9. The system logs the sharing action and displays a confirmation message to the user.   
  
Alternative Flow:   
1. If the user selects an invalid or non-existent calendar, the system displays an error and prompts for a valid selection.   
2. If the system is unable to send the sharing notification due to email system unavailability, an error message is displayed to the user.   
3. If the user cancels the sharing process after selecting recipients, the system reverts to the calendar view and does not share the calendar.   
4. If the selected recipient lacks the necessary permissions to access shared calendars, the system denies the action and displays a permission error.   
5. If the system detects a conflict in access rights (e.g., multiple sharing requests), it alerts the user and provides options to resolve the conflict.  
  
Use Case Name: Manage Invitations   
Use Case ID: UC-22   
Actors: User, Administrator, Email System, Calendar System, Contact, Distribution Group, Shared Account   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the calendar system.   
2. The calendar system must be connected to the email system to send and manage invitations.   
3. The event to which the invitation is related must exist in the calendar system.   
4. The user must select recipients from the contact list, distribution group, or manually enter email addresses.   
5. The user must have the necessary permissions to manage invitations for the event.   
  
Postconditions:   
1. Invitations are successfully sent, updated, or revoked for the selected event.   
2. All invitation-related emails are recorded in the email archive.   
3. The system logs the invitation management action for audit and compliance.   
4. Recipients receive appropriate notifications (e.g., invitation, update, cancellation).   
5. The user receives a confirmation message for each action performed.   
  
Main Flow:   
1. The user or administrator navigates to the calendar section and selects an event.   
2. The system displays the event details, including the list of invited participants.   
3. The user selects an action: "Send Invitations," "Update Invitations," or "Revoke Invitations."   
4. For "Send Invitations," the user selects recipients from the contact list or distribution group and manually adds any additional email addresses.   
5. The system validates the selected recipients and generates the invitation email.   
6. The invitation is sent to the recipients via the email system.   
7. The system updates the event with the invitation status and logs the action.   
8. A confirmation message is displayed to the user, indicating the invitations were successfully sent.   
  
Alternative Flow:   
1. If the user selects an invalid or non-existent event, the system displays an error and prompts for a valid selection.   
2. If the system is unable to send invitations due to email server unavailability, an error message is displayed, and the action is rolled back.   
3. If the user cancels the invitation management process before sending, the system reverts to the event details view and discards changes.   
4. If the system detects duplicate invitations for the same recipient, it alerts the user and offers to merge or remove duplicates.   
5. If the event is governed by a retention policy that restricts modifications, the system displays a warning and prevents the invitation action until the policy allows it.  
  
Use Case Name: Monitor System Usage   
Use Case ID: UC-23   
Actors: Administrator, Email System, Calendar System, Email Account, Shared Account, Distribution Group, Archive, Retention Policy   
  
Preconditions:   
1. The administrator must be authenticated and have the appropriate permissions to monitor system usage.   
2. The email and calendar systems must be connected to the monitoring module and database.   
3. The system must have usage data (e.g., email activity, calendar access, account usage) available for review.   
4. The administrator must initiate the monitoring process through the system's administrative interface.   
  
Postconditions:   
1. The system provides a summary or report of current and historical usage data.   
2. The report includes metrics such as email volume, calendar activity, account access, and distribution group usage.   
3. The system logs the monitoring action for audit and compliance tracking.   
4. The administrator receives a confirmation message that the report has been generated.   
5. The system ensures that sensitive user data is anonymized or restricted based on privacy policies.   
  
Main Flow:   
1. The administrator logs into the email and calendar system and navigates to the "System Monitoring" section.   
2. The system displays options to monitor current usage or retrieve historical usage reports.   
3. The administrator selects the time frame and scope (e.g., all users, specific accounts, shared accounts).   
4. The system queries the database for relevant usage data, including email sends/receives, calendar event creation/modifications, and distribution group activity.   
5. The system compiles the data into a structured report (e.g., graphical dashboard, summary table).   
6. The administrator reviews the report and may export it for further analysis.   
7. The system logs the monitoring action and displays a confirmation message.   
  
Alternative Flow:   
1. If the system is unable to retrieve usage data due to database or server issues, an error message is displayed to the administrator.   
2. If the selected scope includes restricted data (e.g., private user activity), the system filters out unauthorized information and displays only accessible data.   
3. If the administrator cancels the monitoring process before generating the report, the system returns to the previous screen and discards any pending data.   
4. If the system detects anomalies in the usage data (e.g., unusually high email volume), it alerts the administrator for further investigation.   
5. If the selected time frame is invalid or exceeds system limits, the system prompts the administrator to adjust the time frame before proceeding.  
  
Use Case Name: Generate Compliance Reports   
Use Case ID: UC-24   
Actors: Administrator, Email System, Archive, Retention Policy, Distribution Group, Email Account, Shared Account   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to generate compliance reports.   
2. The email system must be connected to the archive and retention policy database.   
3. The system must have access to user activity logs, email metadata, and calendar data if relevant.   
4. The administrator must specify the report scope, time frame, and filters (e.g., users, groups, policies).   
5. The system must be configured to support report generation and export.   
  
Postconditions:   
1. A compliance report is generated based on the specified criteria.   
2. The report includes details such as email activity, retention policy compliance, archive status, and system usage.   
3. The system logs the report generation action for audit purposes.   
4. The administrator receives the report in the selected format (e.g., PDF, CSV).   
5. The report is available for review, sharing, or storage as per system settings.   
  
Main Flow:   
1. The administrator logs into the system and navigates to the "Compliance Reports" section.   
2. The system displays options to define the report scope (e.g., specific users, shared accounts, distribution groups).   
3. The administrator selects the time frame (e.g., daily, weekly, custom dates).   
4. The administrator applies filters (e.g., retention policy status, email type, calendar events).   
5. The system validates the selected parameters and prepares to generate the report.   
6. The system queries the archive, retention policy database, and email activity logs for the required data.   
7. The system compiles the data into a structured compliance report.   
8. The administrator selects the desired report format (e.g., PDF, CSV, Excel).   
9. The system generates the report and provides it for download or sharing.   
10. A confirmation message is displayed to the administrator.   
  
Alternative Flow:   
1. If the selected time frame is invalid or exceeds system limits, the system displays an error and prompts the administrator to adjust the dates.   
2. If the system cannot access the archive or logs due to server or database issues, an error message is displayed, and the report generation is rolled back.   
3. If no data matches the specified criteria, the system generates an empty report and informs the administrator.   
4. If the administrator cancels the report generation before completion, the system discards the in-progress data and returns to the compliance interface.   
5. If the report exceeds the system’s file size limit, the administrator is prompted to adjust the filters or split the report into smaller segments.  
  
Use Case Name: Update Task Status   
Use Case ID: UC-25   
Actors: User, Administrator, Email System, Calendar System, Email Account, Shared Account, Distribution Group, Task Module   
  
Preconditions:   
1. The user must be authenticated and have access to the task management module integrated with the email and calendar systems.   
2. The task to be updated must exist in the system and be associated with a user or shared account.   
3. The system must be connected to the database to store task status changes.   
4. The user must select the task they wish to update.   
5. The system must support task status options (e.g., In Progress, Completed, Deferred).   
  
Postconditions:   
1. The selected task’s status is updated in the system.   
2. The task update is reflected in the user’s calendar and email system if linked.   
3. The system logs the task status update for audit and recovery purposes.   
4. The user receives a confirmation message indicating the task status was successfully updated.   
5. If the task is part of a distribution group or shared account, the status change is synchronized for all relevant users.   
  
Main Flow:   
1. The user logs into the system and navigates to the task management interface.   
2. The system displays a list of tasks assigned to the user or shared account.   
3. The user selects the task they wish to update.   
4. The system opens the task details and displays the current status.   
5. The user selects a new status from the available options.   
6. The user confirms the update.   
7. The system validates the new status and updates the task in the database.   
8. If the task is linked to a calendar event, the system updates the event status or notes accordingly.   
9. If the task is associated with an email or email flow, the system updates the related metadata.   
10. A confirmation message is displayed to the user.   
  
Alternative Flow:   
1. If the user selects an invalid or unsupported status, the system displays an error and prompts for a valid selection.   
2. If the system is unable to update the task due to database or server issues, an error message is displayed, and the action is rolled back.   
3. If the user cancels the update after selecting a new status, the system reverts to the task details view and retains the original status.   
4. If the task is governed by a retention policy that restricts modifications, the system displays a warning and prevents the update until the policy allows it.   
5. If the task is part of a shared account or distribution group and the user lacks necessary permissions, the system denies the update and displays a permission error.  
  
Use Case Name: Delete Task   
Use Case ID: UC-26   
  
Actors: User, Administrator, Task Module, Calendar System, Email System, Email Account, Shared Account, Distribution Group   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the task management module.   
2. The task to be deleted must exist in the system and be associated with the user or shared account.   
3. The system must be connected to the database to remove the task.   
4. The user must select the task they wish to delete.   
5. The system must have the capability to synchronize task deletion with calendar and email systems if applicable.   
  
Postconditions:   
1. The selected task is removed from the system.   
2. If the task is linked to a calendar event, the event is updated or removed accordingly.   
3. If the task is associated with an email or email flow, the system updates the metadata to reflect the deletion.   
4. The system logs the task deletion for audit and recovery purposes.   
5. The user receives a confirmation message indicating the task was successfully deleted.   
  
Main Flow:   
1. The user logs into the system and navigates to the task management interface.   
2. The system displays a list of tasks assigned to the user or shared account.   
3. The user selects the task to be deleted.   
4. The system prompts the user to confirm the deletion.   
5. The user confirms the deletion.   
6. The system removes the task from the database.   
7. If the task is linked to a calendar event, the system updates or removes the event.   
8. If the task is associated with an email or email flow, the system updates the relevant records.   
9. The system logs the deletion action and updates the task list.   
10. A confirmation message is displayed to the user.   
  
Alternative Flow:   
1. If the user selects an invalid or non-existent task, the system displays an error and prompts for a valid selection.   
2. If the task is governed by a retention policy that restricts deletion, the system displays a warning and prevents the action until the policy allows it.   
3. If the system is unable to delete the task due to database or server issues, an error message is displayed, and the action is rolled back.   
4. If the user cancels the deletion after selecting the task, the system reverts to the task list and retains the task.   
5. If the task is part of a shared account or distribution group and the user lacks necessary permissions, the system denies the deletion and displays a permission error.  
  
Use Case Name: View Task Progress   
Use Case ID: UC-27   
Actors: User, Administrator, Task Module, Calendar System, Email System, Email Account, Shared Account, Distribution Group   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the task management module.   
2. The task to be viewed must exist in the system and be associated with the user or shared account.   
3. The system must be connected to the database to retrieve task progress data.   
4. The task must have been previously created or assigned to the user or group.   
5. The system must support task progress tracking and display features.   
  
Postconditions:   
1. The task progress is displayed to the user or administrator.   
2. The system updates the user interface to reflect the latest status and progress of the task.   
3. If the task is linked to a calendar or email, the system synchronizes the progress information accordingly.   
4. The system logs the access to the task progress for audit and compliance.   
5. The user receives a visual confirmation that the task progress is successfully displayed.   
  
Main Flow:   
1. The user logs into the system and navigates to the task management interface.   
2. The system displays a list of tasks assigned to the user or shared account.   
3. The user selects a task from the list to view its progress.   
4. The system retrieves the task's current status, due date, assigned users, and any associated notes or updates.   
5. The system presents the task progress in a structured format, including percentage completed, milestones, or due dates.   
6. The user can optionally add or modify progress notes within the task details.   
7. The system updates the task progress in the database and logs the changes.   
8. If the task is linked to a calendar event, the system updates the event status accordingly.   
9. A confirmation message is displayed to the user, indicating that the progress has been successfully viewed and updated.   
  
Alternative Flow:   
1. If the user selects an invalid or non-existent task, the system displays an error and prompts for a valid selection.   
2. If the system is unable to retrieve task progress due to database or server issues, an error message is displayed, and the task is marked as unavailable.   
3. If the task is governed by a retention policy that restricts viewing or modifying progress, the system displays a warning and prevents the action until the policy allows it.   
4. If the user cancels the progress update after viewing, the system reverts to the task list and retains the original task status.   
5. If the task is part of a shared account or distribution group and the user lacks necessary permissions, the system denies access and displays a permission error.  
  
Use Case Name: Manage Event   
Use Case ID: UC-28   
Actors: User, Administrator, Calendar System, Email System, Contact, Distribution Group, Shared Account   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the calendar system.   
2. The event to be managed must exist in the calendar system and be associated with the user or shared account.   
3. The system must be connected to the database to update event details.   
4. The user must select the event they wish to manage.   
5. The system must have valid configurations for sending email notifications to participants if required.   
  
Postconditions:   
1. The selected event is updated with new information (title, date, time, participants, description, or recurrence settings).   
2. If the event includes participants, the system sends updated notifications via email as configured.   
3. The system logs the event management action for audit and compliance tracking.   
4. The event is synchronized with the calendar view and any associated shared calendars.   
5. The user receives a confirmation message indicating the event was successfully managed.   
  
Main Flow:   
1. The user logs into the email and calendar system and navigates to the calendar section.   
2. The user selects an existing event from the calendar view.   
3. The system opens the event details and displays available management options (e.g., edit, reschedule, update participants).   
4. The user modifies the event details (e.g., changes the date, adds or removes participants, updates the description).   
5. The system validates the updated event information (e.g., valid date/time, active participants).   
6. The user confirms the changes and clicks the "Save" or "Update" button.   
7. The system updates the event in the calendar database.   
8. If the event has participants, the system sends updated email notifications to inform them of the changes.   
9. The system updates the calendar view and logs the management action.   
10. A confirmation message is displayed to the user, indicating the event was successfully managed.   
  
Alternative Flow:   
1. If the user enters an invalid date or time, the system displays an error and prompts for correction.   
2. If the updated participant list contains invalid or inactive email addresses, the system alerts the user and allows for correction or removal.   
3. If the user cancels the management action after making changes, the system reverts to the previous event details and returns to the calendar view.   
4. If the system is unable to send updated notifications due to email system unavailability, an error message is displayed, and the action is rolled back.   
5. If the event is part of a shared calendar and the user lacks the necessary permissions, the system denies the management action and displays a permission error.  
  
Use Case Name: Manage Task   
Use Case ID: UC-29   
Actors: User, Administrator, Task Module, Calendar System, Email System, Email Account, Shared Account, Distribution Group   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the task management module.   
2. The task module must be integrated with the email and calendar systems for synchronization.   
3. The task to be managed must exist in the system and be associated with the user or shared account.   
4. The system must be connected to the database to perform task-related operations.   
5. The user must have the necessary permissions to create, edit, or delete tasks.   
  
Postconditions:   
1. The selected task is either created, edited, or deleted based on the requested action.   
2. If the task is linked to a calendar event or email, the system updates the associated records accordingly.   
3. The system logs the task management action for audit and compliance tracking.   
4. The user receives a confirmation message for the performed action.   
5. The task list is updated to reflect the changes made.   
  
Main Flow:   
1. The user logs into the system and navigates to the task management interface.   
2. The system displays a list of tasks assigned to the user or shared account.   
3. The user selects an action: "Create Task," "Edit Task," or "Delete Task."   
4. For "Create Task," the user inputs task details such as title, due date, priority, and description.   
5. The user selects participants from the contact list or distribution group and manually adds any additional email addresses if applicable.   
6. The system validates the task details and participant information.   
7. The system creates or updates the task in the database.   
8. If the task is linked to a calendar event, the system synchronizes the task with the calendar.   
9. If the task is associated with an email or email flow, the system updates the relevant metadata.   
10. The system displays a confirmation message indicating the task has been successfully managed.   
  
Alternative Flow:   
1. If the user enters an invalid due date or priority level, the system displays an error and prompts for correction.   
2. If the system is unable to create or update the task due to database or server issues, an error message is displayed, and the action is rolled back.   
3. If the user selects "Edit Task" for an invalid or non-existent task, the system displays an error and prompts for a valid selection.   
4. If the task is governed by a retention policy that restricts modifications, the system displays a warning and prevents the action until the policy allows it.   
5. If the user selects "Delete Task" and the task is part of a shared account or distribution group with restricted permissions, the system denies the deletion and displays a permission error.  
  
Use Case Name: Manage User Permissions   
Use Case ID: UC-30   
Actors: Administrator, User, Email System, Shared Account, Distribution Group, User Permission Module   
  
Preconditions:   
1. The administrator must be authenticated and have the appropriate permissions to manage user access and roles.   
2. The system must have a configured user permission module with predefined roles and access levels.   
3. The target user or shared account must exist in the system.   
4. The system must be connected to the database to update permission settings.   
5. The administrator must select the user or account they wish to modify permissions for.   
  
Postconditions:   
1. The user’s or shared account’s permissions are updated in the system based on the administrator's input.   
2. The system logs the permission change for audit and compliance tracking.   
3. The affected user receives a notification of the permission update if configured.   
4. The user permission module reflects the updated access rights in the system.   
5. The system ensures that no unauthorized access is granted and that all changes comply with internal security policies.   
  
Main Flow:   
1. The administrator logs into the system and navigates to the "User Permissions" section in the administrative interface.   
2. The system displays a list of users or shared accounts with their current permission levels.   
3. The administrator selects a specific user or shared account for permission modification.   
4. The system opens the user permission configuration panel, displaying available roles and access rights (e.g., send, receive, archive, delete, manage shared accounts, create distribution groups).   
5. The administrator modifies the user's permissions by enabling or disabling specific access rights.   
6. The administrator confirms the changes and clicks "Save."   
7. The system validates the new permission configuration against predefined roles and system policies.   
8. The system updates the user permission module in the database with the new access rights.   
9. If the user is part of shared accounts or distribution groups, the system updates the associated configurations.   
10. The system logs the permission management action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator selects an invalid or non-existent user or shared account, the system displays an error and prompts for a valid selection.   
2. If the system is unable to update the user permissions due to database or server issues, an error message is displayed, and the action is rolled back.   
3. If the administrator attempts to assign conflicting or unrestricted permissions that violate system policies, the system displays a warning and prevents the action until the configuration is corrected.   
4. If the user is governed by a retention policy or compliance rule that restricts permission changes, the system displays a warning and prevents the modification until the policy allows it.   
5. If the administrator cancels the permission update after making changes, the system reverts to the previous permission settings and returns to the user list.  
  
Use Case Name: Generate Compliance Report   
Use Case ID: UC-31   
Actors: Administrator, Email System, Archive, Retention Policy, Distribution Group, Email Account, Shared Account   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to generate compliance reports.   
2. The system must have access to the archive and retention policy database to retrieve compliance-related data.   
3. The system must be connected to the email and calendar modules to gather relevant activity logs.   
4. The administrator must specify the report criteria, such as the time frame, users, groups, or policies to include.   
5. The system must have the capability to export or display the report in a selected format (e.g., PDF, CSV).   
  
Postconditions:   
1. The compliance report is generated and delivered to the administrator in the specified format.   
2. The report includes all requested compliance data, such as retention policy adherence, archived email status, and system usage metrics.   
3. The system logs the report generation action for audit purposes.   
4. The administrator receives a confirmation message that the report is ready for download or sharing.   
5. The system ensures that the report complies with internal or external regulatory standards and does not include unauthorized or sensitive data.   
  
Main Flow:   
1. The administrator logs into the system and navigates to the "Compliance Reports" section.   
2. The system displays options to define the report scope (e.g., specific users, shared accounts, distribution groups, or retention policies).   
3. The administrator selects the time frame (e.g., daily, weekly, custom dates).   
4. The administrator applies filters (e.g., retention policy compliance, email activity, calendar events).   
5. The system validates the selected parameters and prepares to generate the report.   
6. The system queries the archive, retention policy database, and email activity logs for the required data.   
7. The system compiles the data into a structured compliance report.   
8. The administrator selects the desired report format (e.g., PDF, CSV, Excel).   
9. The system generates the report and provides it for download or sharing.   
10. A confirmation message is displayed to the administrator indicating the report is ready.   
  
Alternative Flow:   
1. If the selected time frame is invalid or exceeds system limits, the system displays an error and prompts the administrator to adjust the dates.   
2. If the system cannot access the archive or logs due to server or database issues, an error message is displayed, and the report generation is rolled back.   
3. If no data matches the specified criteria, the system generates an empty report and informs the administrator.   
4. If the administrator cancels the report generation before completion, the system discards the in-progress data and returns to the compliance interface.   
5. If the report exceeds the system’s file size limit, the administrator is prompted to adjust the filters or split the report into smaller segments.  
  
Use Case Name: View Task Progress   
Use Case ID: UC-32   
Actors: User, Administrator, Task Module, Calendar System, Email System, Email Account, Shared Account, Distribution Group   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the task management module.   
2. The task to be viewed must exist in the system and be associated with the user, shared account, or distribution group.   
3. The system must be connected to the database to retrieve task progress data.   
4. The task must have been previously created or assigned and must contain progress tracking information (e.g., status, completion percentage).   
5. The system must have the necessary configurations to display task progress (e.g., task status fields, progress indicators).   
  
Postconditions:   
1. The task progress is displayed to the user or administrator in a clear and structured format.   
2. The system updates the user interface to reflect the latest status and progress of the task.   
3. If the task is linked to a calendar or email, the system ensures progress synchronization is up to date.   
4. The system logs the viewing of task progress for audit and compliance tracking.   
5. The user receives a visual confirmation that the task progress has been successfully retrieved and displayed.   
  
Main Flow:   
1. The user or administrator logs into the system and navigates to the task management interface.   
2. The system displays a list of tasks associated with the user, shared account, or group.   
3. The user selects a task from the list to view its progress.   
4. The system retrieves the task’s current status, completion percentage, due date, and any associated progress notes or updates.   
5. The system presents the task progress in a structured format, such as a progress bar, status summary, or detailed timeline.   
6. If the task is linked to a calendar event, the system displays the event status and related details.   
7. The system logs the access to the task progress for audit and compliance purposes.   
8. The user or administrator reviews the progress information and may choose to close the view or take further action (e.g., update task status).   
9. A confirmation message is displayed to the user or administrator indicating the task progress has been successfully viewed.   
  
Alternative Flow:   
1. If the user selects an invalid or non-existent task, the system displays an error message and prompts for a valid selection.   
2. If the system is unable to retrieve task progress due to database or server issues, an error message is displayed, and the task is marked as unavailable.   
3. If the task is governed by a retention policy that restricts viewing progress, the system displays a warning and prevents the action until the policy allows it.   
4. If the user or administrator cancels the view after retrieving the task progress, the system returns to the task list without further action.   
5. If the task is part of a shared account or distribution group and the user lacks the necessary permissions, the system denies access and displays a permission error.  
  
Use Case Name: View Distribution Group   
Use Case ID: UC-33   
Actors: User, Administrator, Distribution Group, Contact System, Email System   
  
Preconditions:   
1. The user or administrator must be authenticated and have the necessary permissions to view distribution groups.   
2. The distribution group must exist in the system and be associated with the user or shared account.   
3. The system must be connected to the database to retrieve group details.   
4. The user must select the specific distribution group to be viewed.   
5. The system must have the capability to display group members and related information (e.g., name, description, member list).   
  
Postconditions:   
1. The distribution group’s details are displayed to the user or administrator, including name, description, and member list.   
2. The system logs the viewing action for audit and compliance tracking.   
3. The user or administrator receives a confirmation message indicating the group information has been successfully retrieved.   
4. If the group is associated with an email flow or retention policy, the system displays relevant metadata.   
5. The system ensures that sensitive information is not disclosed and that access complies with internal security policies.   
  
Main Flow:   
1. The user or administrator logs into the email system and navigates to the "Distribution Groups" section.   
2. The system displays a list of distribution groups the user or administrator is authorized to view.   
3. The user selects a specific distribution group from the list.   
4. The system retrieves the group’s information, including its name, description, and current member list.   
5. The system displays the retrieved details in a structured format, allowing the user to review the group configuration.   
6. If the group is associated with an email flow or retention policy, the system highlights those connections.   
7. The system logs the view action, including the user, timestamp, and group name.   
8. A confirmation message is displayed to the user or administrator indicating the group has been successfully viewed.   
  
Alternative Flow:   
1. If the user selects an invalid or non-existent distribution group, the system displays an error message and prompts for a valid selection.   
2. If the system is unable to retrieve the distribution group due to database or server issues, an error message is displayed, and the action is rolled back.   
3. If the user lacks permission to view the selected group, the system denies access and displays a permission error.   
4. If the user cancels the viewing process after selecting the group, the system returns to the distribution group list without further action.   
5. If the group is governed by a retention policy or compliance rule that restricts access to its details, the system displays a warning and prevents the view until the policy allows it.  
  
Use Case Name: Manage Email Flow   
Use Case ID: UC-34   
Actors: Administrator, Email System, Email Account, Shared Account, Distribution Group, Email Flow, Retention Policy, Archive   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to configure or manage email flows.   
2. The email system must be connected to the email server, archive, and retention policy module to enable flow management.   
3. The system must have an existing email flow or allow the administrator to create a new one.   
4. The email flow must be associated with an email account, shared account, or distribution group.   
5. The system must be able to enforce the configured flow rules without conflicting with existing policies.   
  
Postconditions:   
1. The email flow is either created, updated, or deleted as per the administrator's request.   
2. The system enforces the flow rules for the associated accounts or groups (e.g., routing, filtering, or forwarding).   
3. The system logs all flow management actions for audit and compliance tracking.   
4. The administrator receives a confirmation message indicating the success of the action.   
5. If the flow is deleted or modified, any dependent configurations (e.g., retention policies, distribution groups) are updated accordingly.   
  
Main Flow:   
1. The administrator logs into the email system and navigates to the "Email Flow Management" section.   
2. The system displays a list of existing email flows and provides options to create, edit, or delete a flow.   
3. The administrator selects an action: "Create Email Flow," "Edit Email Flow," or "Delete Email Flow."   
4. For "Create Email Flow," the administrator inputs a name for the flow and defines the rules (e.g., source account, conditions, destination folder or shared account).   
5. The system validates the flow configuration (e.g., valid source and destination, proper conditions).   
6. The administrator assigns the flow to an email account, shared account, or distribution group.   
7. The system updates the email flow settings in the database and applies the rules to the associated accounts or groups.   
8. The system logs the flow management action, including the administrator, timestamp, and details of the flow.   
9. A confirmation message is displayed to the administrator indicating the email flow has been successfully managed.   
  
Alternative Flow:   
1. If the administrator inputs an invalid or duplicate flow name, the system displays an error and prompts for a valid name.   
2. If the system is unable to apply the flow due to server or database issues, an error message is displayed, and the action is rolled back.   
3. If the administrator selects "Edit Email Flow" for an invalid or non-existent flow, the system displays an error and prompts for a valid selection.   
4. If the email flow is referenced in an active retention policy or distribution group, the system alerts the administrator and prevents deletion or modification until dependencies are resolved.   
5. If the administrator cancels the flow management operation after making changes, the system reverts to the flow list and retains the original configuration.  
  
Use Case Name: Manage Retention Policy   
Use Case ID: UC-35   
Actors: Administrator, Email System, Retention Policy Module, Email Account, Shared Account, Distribution Group, Archive   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to manage retention policies.   
2. The retention policy module must be active and connected to the email system and archive for enforcement and tracking.   
3. The system must be able to access the database to store or retrieve retention policy configurations.   
4. The administrator must select the specific retention policy to be created, updated, or deleted.   
5. The system must support the definition of retention periods, conditions, and associated email accounts or groups.   
  
Postconditions:   
1. The retention policy is either created, updated, or deleted based on the administrator's action.   
2. The system enforces the retention policy for the associated email accounts, shared accounts, or distribution groups.   
3. The system logs the retention policy management action for audit and compliance tracking.   
4. The administrator receives a confirmation message indicating the success of the action.   
5. The system ensures that existing emails governed by the policy are not deleted prematurely.   
  
Main Flow:   
1. The administrator logs into the email system and navigates to the "Retention Policies" section in the administrative interface.   
2. The system displays a list of existing retention policies, including their names, durations, and associated accounts or groups.   
3. The administrator selects an action: "Create Retention Policy," "Edit Retention Policy," or "Delete Retention Policy."   
4. For "Create Retention Policy," the administrator inputs a name for the policy and defines the retention period (e.g., 1 year, 6 months) and conditions (e.g., email type, folder, sender/recipient criteria).   
5. The administrator selects the target email accounts, shared accounts, or distribution groups to apply the policy to.   
6. The system validates the policy configuration (e.g., valid retention period, no conflicting rules).   
7. The system updates the retention policy module in the database with the new or modified policy.   
8. The system enforces the policy on the associated accounts or groups.   
9. A confirmation message is displayed to the administrator indicating the retention policy has been successfully managed.   
  
Alternative Flow:   
1. If the administrator inputs an invalid or duplicate policy name, the system displays an error and prompts for a valid name.   
2. If the system is unable to update or store the retention policy due to database or server issues, an error message is displayed, and the action is rolled back.   
3. If the administrator selects "Edit Retention Policy" for an invalid or non-existent policy, the system displays an error and prompts for a valid selection.   
4. If the retention policy is actively governing emails, the system alerts the administrator and prevents deletion or modification until the policy is no longer in use or dependencies are resolved.   
5. If the administrator cancels the policy management operation after making changes, the system reverts to the policy list and retains the original configuration.  
  
Use Case Name: Manage Expired Email   
Use Case ID: UC-36   
Actors: Administrator, Email System, Email Account, Shared Account, Archive, Retention Policy, Expired Email   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to manage expired emails.   
2. The system must have an active retention policy that defines the expiration criteria for emails.   
3. The email system must be connected to the archive and email server to process expired emails.   
4. Expired emails must be identified based on the retention policy rules.   
5. The system must have the capability to either archive or delete expired emails.   
  
Postconditions:   
1. Expired emails are either moved to the archive or deleted from the system according to the configured policy or manual decision.   
2. The system logs the management of expired emails for audit and recovery purposes.   
3. The administrator is notified of the action taken on the expired emails.   
4. If archived, the emails remain recoverable through the recovery process.   
5. The system ensures that no emails are deleted before the retention period is complete unless manually authorized.   
  
Main Flow:   
1. The email system automatically identifies emails that have exceeded their retention period as defined by the retention policy.   
2. The system categorizes these emails as "Expired" and prepares them for management.   
3. The administrator initiates the "Manage Expired Emails" process from the system interface.   
4. The system displays a list of expired emails, including details such as subject, sender, recipient, date, and folder.   
5. The administrator selects an action: "Archive Expired Emails" or "Delete Expired Emails."   
6. The system validates the action against the current retention policy and system rules.   
7. The system performs the selected action (archive or delete) on the expired emails.   
8. If archived, the system moves the emails to the archive and updates the user interface.   
9. If deleted, the system removes the emails from the system and logs the deletion.   
10. A confirmation message is displayed to the administrator indicating the success of the action.   
  
Alternative Flow:   
1. If the system is unable to access the archive or server, an error message is displayed, and the action is rolled back.   
2. If the administrator cancels the action after selecting expired emails, the system reverts to the previous screen and leaves the emails untouched.   
3. If an expired email is still referenced in an active email flow or distribution group, the system alerts the administrator and prevents deletion until the dependency is resolved.   
4. If the system detects an error during the archiving or deletion process, it logs the error and notifies the administrator for review.   
5. If the retention policy is updated during the process, the system re-evaluates the expiration status of the emails and adjusts the action accordingly.  
  
Use Case Name: Manage Recovery   
Use Case ID: UC-37   
Actors: User, Administrator, Email System, Archive, Recovery Module, Email Account, Shared Account   
  
Preconditions:   
1. The user or administrator must be authenticated and have access to the recovery module.   
2. The system must be connected to the archive database to retrieve recoverable data.   
3. The recovery module must be enabled and configured to allow restoration of archived emails.   
4. The user or administrator must identify the email or set of emails that require recovery.   
5. The system must ensure that the recovery action does not conflict with active retention policies.   
  
Postconditions:   
1. The selected emails are successfully recovered from the archive and restored to the user’s inbox or a specified folder.   
2. The recovery action is logged for audit and compliance tracking.   
3. The system updates the archive to reflect the recovery and ensures data integrity.   
4. The user or administrator receives a confirmation message indicating the recovery was successful.   
5. If the recovered emails are linked to calendar events or tasks, the system synchronizes the related information accordingly.   
  
Main Flow:   
1. The user or administrator logs into the email system and navigates to the "Recovery" section.   
2. The system displays a list of recoverable emails based on the archive database, including metadata such as subject, date, and original folder.   
3. The user or administrator selects one or more emails to recover.   
4. The system checks if the selected emails are eligible for recovery (e.g., not locked by a retention policy).   
5. The system initiates the recovery process by retrieving the selected emails from the archive.   
6. The system restores the recovered emails to the user's inbox or a specified folder (e.g., "Recovered Emails").   
7. If the email is linked to calendar events or tasks, the system updates the associated records.   
8. The system logs the recovery action, including the user or administrator who performed the action, the timestamp, and the emails recovered.   
9. A confirmation message is displayed to the user or administrator, indicating the success of the recovery operation.   
  
Alternative Flow:   
1. If the selected email is governed by an active retention policy that prevents recovery, the system displays a warning and prevents the action until the policy allows it.   
2. If the system is unable to access the archive due to technical issues, an error message is displayed, and the recovery is rolled back.   
3. If the user or administrator selects an invalid or non-existent email, the system displays an error and prompts for a valid selection.   
4. If the recovery process encounters an error (e.g., corrupted data, insufficient permissions), the system logs the issue and displays an error message to the user or administrator.   
5. If the user or administrator cancels the recovery process after selecting emails, the system reverts to the recovery interface and leaves the emails in the archive.  
  
Use Case Name: View User Permission   
Use Case ID: UC-38   
Actors: Administrator, User, User Permission Module, Email System, Shared Account, Distribution Group   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to view user access rights and roles.   
2. The user permission module must be active and connected to the email system and shared account configurations.   
3. The target user or shared account must exist in the system.   
4. The system must be connected to the database to retrieve the current permission settings.   
5. The administrator must select the specific user or shared account to view their permissions.   
  
Postconditions:   
1. The user’s or shared account’s permissions are displayed in a structured format (e.g., roles, access rights, restrictions).   
2. The system logs the viewing action for audit and compliance tracking.   
3. The administrator receives a confirmation message that the permission details have been successfully retrieved.   
4. The user permission module reflects the current access rights in the system.   
5. The system ensures that sensitive permission data is not disclosed to unauthorized users.   
  
Main Flow:   
1. The administrator logs into the system and navigates to the "User Permissions" section in the administrative interface.   
2. The system displays a list of users and shared accounts for which permissions can be viewed.   
3. The administrator selects a specific user or shared account from the list.   
4. The system retrieves the permission details associated with the selected user or shared account.   
5. The system presents the permissions in a structured view (e.g., read, write, send, receive, archive, delete, manage shared accounts, create distribution groups).   
6. If the user or shared account is part of distribution groups or email flows, the system displays related permission dependencies.   
7. The system logs the viewing action, including the administrator, timestamp, and user or shared account being viewed.   
8. The administrator reviews the displayed permissions and may choose to close the view or proceed to modify them.   
9. A confirmation message is displayed to the administrator, indicating the permission details have been successfully viewed.   
  
Alternative Flow:   
1. If the administrator selects an invalid or non-existent user or shared account, the system displays an error and prompts for a valid selection.   
2. If the system is unable to retrieve the permission details due to database or server issues, an error message is displayed, and the viewing action is rolled back.   
3. If the administrator lacks permission to view the selected user’s or shared account’s permissions, the system denies access and displays a permission error.   
4. If the administrator cancels the viewing process after selecting the user or shared account, the system returns to the user or shared account list without further action.   
5. If the user or shared account is governed by a retention policy or compliance rule that restricts access to permission details, the system displays a warning and prevents the view until the policy allows it.  
  
Use Case Name: Manage Compliance Report   
Use Case ID: UC-39   
Actors: Administrator, Email System, Compliance Report Module, Archive, Retention Policy, Email Account, Shared Account, Distribution Group   
  
Preconditions:   
1. The administrator must be authenticated and have the necessary permissions to manage compliance reports.   
2. The compliance report module must be active and connected to the archive, email system, and retention policy database.   
3. At least one compliance report must exist in the system, or the administrator must provide valid parameters to generate a new report.   
4. The system must be configured to allow the viewing, exporting, or deletion of compliance reports.   
5. The system must ensure that all actions taken are in compliance with internal and external data governance policies.   
  
Postconditions:   
1. The compliance report is either created, viewed, exported, or deleted, based on the administrator's action.   
2. The system logs the compliance report management action for audit and compliance tracking.   
3. The administrator receives a confirmation message for each action performed.   
4. If the report is exported, the file is generated and made available for download or sharing.   
5. If the report is deleted, the system ensures it is removed only if not protected by retention or compliance rules.   
  
Main Flow:   
1. The administrator logs into the system and navigates to the "Compliance Reports" section in the administrative interface.   
2. The system displays a list of existing compliance reports, including their names, generation dates, and associated policies or accounts.   
3. The administrator selects a compliance report from the list and chooses an action: "View Report," "Export Report," or "Delete Report."   
4. If the action is "View Report," the system retrieves the report and displays it in a readable format.   
5. If the action is "Export Report," the system prompts the administrator to choose the export format (e.g., PDF, CSV, Excel).   
6. If the action is "Delete Report," the system checks if the report is under any retention policy.   
7. The system validates the action and ensures it does not violate any data governance rules.   
8. The system performs the selected action, updating the compliance report module and relevant data stores.   
9. A confirmation message is displayed to the administrator indicating the action was successfully completed.   
10. The system logs the operation in the audit trail for traceability and compliance.   
  
Alternative Flow:   
1. If the selected compliance report does not exist or is invalid, the system displays an error and prompts the administrator to select a valid report.   
2. If the system is unable to retrieve or export the report due to database or server issues, an error message is displayed, and the action is rolled back.   
3. If the administrator attempts to delete a compliance report that is protected by a retention policy, the system displays a warning and prevents the deletion until the policy allows it.   
4. If the administrator cancels the action at any point, the system reverts to the compliance report list and discards any changes.   
5. If the export format selected is not supported or not available for the report type, the system displays an error and suggests valid options.