项目文档

# Functional Requirement

1. Functional Requirements   
  
1.1 Email Creation Function   
 Function ID: FR-01   
 Description: A user or administrator can create a new email by filling in the subject, body, and recipient information. The email must be validated and stored in the email archive.   
 Input: Email subject, body, sender ID, recipient ID, timestamp, and formatting rules (optional).   
 Output: A new email stored in the email archive with a unique EmailID and status updated to "Draft" or "Pending Approval."  
  
1.2 Email Sending Function   
 Function ID: FR-02   
 Description: A user or administrator can send an existing email to the intended recipient after validation and approval (if required).   
 Input: Email to be sent, recipient validation result, and administrative approval status (if required).   
 Output: Email status updated to "Sent" in the email archive, and a confirmation message to the user.  
  
1.3 Email Receiving Function   
 Function ID: FR-03   
 Description: The system receives an email from an external source or another user, validates it, and stores it in the email archive under the recipient's information.   
 Input: Incoming email data including sender, recipient, subject, body, and timestamp.   
 Output: Email stored in the email archive with status "Received" and a notification sent to the recipient user.  
  
1.4 Email Reading Function   
 Function ID: FR-04   
 Description: A user or administrator can read the content of an existing email in the email archive. The system updates the email status to "Read" and logs the action if required.   
 Input: EmailID of the selected email.   
 Output: Displayed email content (subject, sender, body, attachments) and updated email status to "Read."  
  
1.5 Email Deletion Function   
 Function ID: FR-05   
 Description: A user or administrator can delete an email from the system. The system must confirm the deletion and log it if an administrator is involved.   
 Input: EmailID of the email to be deleted.   
 Output: Email removed from the email archive and a confirmation message sent to the user or administrator.  
  
1.6 Email Archiving Function   
 Function ID: FR-06   
 Description: A user or administrator can archive an email, moving it from the active inbox or sent items to the email archive.   
 Input: EmailID of the email to be archived.   
 Output: Email moved to the archive with updated status and a confirmation message to the user or administrator.  
  
1.7 Email Recovery Function   
 Function ID: FR-07   
 Description: A user or administrator can recover an archived email, moving it back to the active inbox or sent items.   
 Input: EmailID of the archived email.   
 Output: Email moved back to the active section with updated status and a confirmation message to the user or administrator.  
  
1.8 Contact Creation Function   
 Function ID: FR-08   
 Description: A user or administrator can create a new contact with name, email address, and phone number. The contact must be validated and stored in the user account database.   
 Input: Name, email address, phone number, and UserID.   
 Output: A new contact stored in the user account database with a unique ContactID and status updated to "Pending Approval" or "Active."  
  
1.9 Contact Update Function   
 Function ID: FR-09   
 Description: A user or administrator can update the information of an existing contact. The system must validate the new data and update the contact in the user account database.   
 Input: ContactID and updated contact information (name, email, phone number).   
 Output: Updated contact information stored in the user account database and a confirmation message to the user or administrator.  
  
1.10 Contact Deletion Function   
 Function ID: FR-10   
 Description: A user or administrator can delete a contact from the user account database. The system must confirm the deletion and log it if an administrator is involved.   
 Input: ContactID of the contact to be deleted.   
 Output: Contact removed from the database and a confirmation message sent to the user or administrator.  
  
1.11 Contact Viewing Function   
 Function ID: FR-11   
 Description: A user or administrator can view the details of an existing contact. The system must retrieve and display the contact information without modifying it.   
 Input: ContactID of the contact to be viewed.   
 Output: Displayed contact details (name, email, phone number, notes) and a log entry if an administrator is viewing it.  
  
1.12 User Account Creation Function   
 Function ID: FR-12   
 Description: An administrator can create a new user account by entering the username, email, password, and role. The system must validate the input and store the account in the user account database.   
 Input: Username, email address, password, and role.   
 Output: A new user account stored in the user account database with a unique AccountID and a confirmation message to the administrator.  
  
1.13 User Account Update Function   
 Function ID: FR-13   
 Description: A user or administrator can update the information of an existing user account. The system must validate the new data and update the account in the user account database.   
 Input: AccountID and updated user account information (username, password, email, role).   
 Output: Updated user account information stored in the user account database and a confirmation message to the user or administrator.  
  
1.14 User Account Deletion Function   
 Function ID: FR-14   
 Description: A user or administrator can delete a user account. The system must confirm the deletion, remove the account, and delete or archive associated data.   
 Input: AccountID of the account to be deleted.   
 Output: User account removed from the database, associated data deleted or archived, and a confirmation message to the user or administrator.  
  
1.15 User Account Viewing Function   
 Function ID: FR-15   
 Description: A user or administrator can view the details of an existing user account. The system must retrieve and display the account information without modifying it.   
 Input: AccountID of the user account to be viewed.   
 Output: Displayed user account details (username, email, role, creation date, last login) and a log entry if an administrator is viewing it.  
  
1.16 Email Folder Management Function   
 Function ID: FR-16   
 Description: A user or administrator can manage email folders by creating, renaming, deleting, or moving emails between folders.   
 Input: FolderID and folder management action (create, rename, delete, move).   
 Output: Updated email folder structure in the database and a confirmation message to the user or administrator.  
  
1.17 Email Format Management Function   
 Function ID: FR-17   
 Description: A user or administrator can manage email formatting rules by creating, editing, or deleting formatting templates.   
 Input: FormatID and formatting action (create, edit, delete).   
 Output: Updated email format configuration in the database and a confirmation message to the user or administrator.  
  
1.18 Email Format Application Function   
 Function ID: FR-18   
 Description: A user or administrator can apply formatting rules to an email, ensuring it is styled according to the system or user-defined template.   
 Input: EmailID and FormatID to be applied.   
 Output: Email updated with the specified formatting rules in the email archive.  
  
1.19 Email Archive Management Function   
 Function ID: FR-19   
 Description: An administrator can manage the email archive by setting retention policies, reorganizing folders, or modifying archive contents.   
 Input: ArchiveID and archive management action (set retention, reorganize, modify).   
 Output: Updated email archive structure and configuration and a confirmation message to the administrator.  
  
1.20 Email Archive Viewing Function   
 Function ID: FR-20   
 Description: A user or administrator can view the contents of the email archive, including filtering and searching for specific emails.   
 Input: ArchiveID or search/filter criteria.   
 Output: Displayed list of archived emails with details (subject, sender, timestamp, status) and a log entry if an administrator is viewing it.  
  
1.21 Administrator Assignment Function   
 Function ID: FR-21   
 Description: An administrator can assign administrative privileges to another user. The system must update the user account and log the action.   
 Input: Target UserID and administrative role assignment.   
 Output: Updated user account with administrative role and a log entry in the archive log.  
  
1.22 Administrator Permission Management Function   
 Function ID: FR-22   
 Description: An administrator can modify the permissions of another administrator, granting or revoking access to specific system functions.   
 Input: Target AdminID and permission update action (grant, revoke).   
 Output: Updated permissions in the user account database and a log entry in the archive log.  
  
1.23 Administrator Deletion Function   
 Function ID: FR-23   
 Description: An administrator can delete another administrator account. The system must confirm the deletion, remove the account, and log the action.   
 Input: AdminID of the administrator to be deleted.   
 Output: Administrator account removed from the database, permissions revoked, and a confirmation message to the administrator.  
  
1.24 Archive Log Management Function   
 Function ID: FR-24   
 Description: An administrator can view, filter, export, or delete entries in the archive log for auditing purposes.   
 Input: Log management action (view, filter, export, delete) and optional search criteria.   
 Output: Updated or retrieved archive log entries and a confirmation message to the administrator.

# External Description

2. External Interfaces   
  
This chapter describes the system’s external interfaces, including user interfaces, hardware interfaces, software interfaces, and communication interfaces. These interfaces define how the system interacts with users, external hardware, software components, and communication protocols.   
  
2.1 User Interface   
  
The system provides a graphical user interface (GUI) for users and administrators to interact with its functionalities. The interface includes screens for creating, sending, receiving, reading, deleting, archiving, and recovering emails, as well as managing contacts, email folders, formatting rules, and archive logs.   
  
- \*\*Email Management Interface\*\*:   
 - Input: Email subject, body, sender ID, recipient ID, timestamp, formatting rules, and actions (create, send, delete, archive, recover).   
 - Output: Confirmation messages, email status updates (e.g., "Draft," "Sent," "Read"), and displayed email content.   
  
- \*\*Contact Management Interface\*\*:   
 - Input: Name, email address, phone number, UserID, and actions (create, update, delete, view).   
 - Output: Confirmation messages, updated contact status (e.g., "Active," "Pending Approval"), and displayed contact details.   
  
- \*\*User Account Management Interface\*\*:   
 - Input: Username, email, password, role, AccountID, and actions (create, update, delete, view).   
 - Output: Confirmation messages, updated account status, and displayed account details (e.g., username, email, role, creation date, last login).   
  
- \*\*Email Folder Management Interface\*\*:   
 - Input: FolderID and folder management actions (create, rename, delete, move).   
 - Output: Confirmation messages and updated folder structure.   
  
- \*\*Email Format Management Interface\*\*:   
 - Input: FormatID and formatting actions (create, edit, delete).   
 - Output: Confirmation messages and updated formatting rules.   
  
- \*\*Email Format Application Interface\*\*:   
 - Input: EmailID and FormatID.   
 - Output: Confirmation message and updated email with applied formatting.   
  
- \*\*Email Archive Management Interface\*\*:   
 - Input: ArchiveID and archive management actions (set retention, reorganize, modify).   
 - Output: Confirmation messages and updated archive structure.   
  
- \*\*Email Archive Viewing Interface\*\*:   
 - Input: ArchiveID or search/filter criteria.   
 - Output: Displayed list of archived emails with details (subject, sender, timestamp, status) and a log entry if an administrator is viewing it.   
  
- \*\*Administrator Privilege Management Interface\*\*:   
 - Input: UserID, AdminID, and administrative actions (assign, modify, delete).   
 - Output: Confirmation messages, updated user/administrator roles, and log entries for auditing.   
  
- \*\*Archive Log Management Interface\*\*:   
 - Input: Log management actions (view, filter, export, delete) and optional search criteria.   
 - Output: Retrieved or modified archive log entries and confirmation messages.   
  
The user interface is designed to be intuitive, accessible via a web browser or desktop application, and supports user authentication and authorization to ensure secure access to administrative functions.   
  
2.2 Hardware Interface   
  
The system does not directly interact with specific hardware devices. However, it relies on the following hardware components for operation:   
  
- \*\*Storage Devices\*\*: The system stores emails, contacts, user accounts, formatting rules, and archive logs on server-based or cloud-based storage.   
- \*\*Computing Devices\*\*: Users and administrators access the system through computing devices such as desktops, laptops, tablets, or smartphones.   
  
No specialized hardware is required for the system to function. It is compatible with standard computing hardware and storage solutions.   
  
2.3 Software Interface   
  
The system interacts with the following software components and databases to manage its core functionalities:   
  
- \*\*Email Archive Database\*\*:   
 - Description: A database storing all email records, including subject, body, sender ID, recipient ID, timestamp, status, and formatting information.   
 - Interaction Method: The system performs CRUD (Create, Read, Update, Delete) operations on this database to manage emails.   
 - Input/Output:   
 - Input: Email subject, body, sender ID, recipient ID, timestamp, formatting rules, EmailID, and actions (create, send, read, delete, archive, recover).   
 - Output: Email records stored in the archive with status updates and confirmation messages.   
  
- \*\*User Account Database\*\*:   
 - Description: A database storing user account information, including username, email, password, role, AccountID, and login history.   
 - Interaction Method: The system performs CRUD operations on this database to manage user accounts and permissions.   
 - Input/Output:   
 - Input: Username, email, password, role, AccountID, and actions (create, update, delete, view).   
 - Output: Updated user account records, confirmation messages, and log entries for administrative actions.   
  
- \*\*Contact Database\*\*:   
 - Description: A database storing contact information, including name, email address, phone number, UserID, and ContactID.   
 - Interaction Method: The system performs CRUD operations to manage contacts.   
 - Input/Output:   
 - Input: Name, email, phone number, UserID, and actions (create, update, delete, view).   
 - Output: Updated contact records, confirmation messages, and log entries for administrative views.   
  
- \*\*Email Folder Database\*\*:   
 - Description: A database storing folder structures for organizing emails.   
 - Interaction Method: The system performs CRUD operations to manage folders and their contents.   
 - Input/Output:   
 - Input: FolderID and actions (create, rename, delete, move).   
 - Output: Updated folder structure and confirmation messages.   
  
- \*\*Email Format Configuration Database\*\*:   
 - Description: A database storing formatting templates and rules for emails.   
 - Interaction Method: The system performs CRUD operations to manage formatting configurations.   
 - Input/Output:   
 - Input: FormatID and actions (create, edit, delete).   
 - Output: Updated formatting rules and confirmation messages.   
  
- \*\*Archive Log Database\*\*:   
 - Description: A database storing logs of administrative actions, such as user account changes, email status updates, and archive modifications.   
 - Interaction Method: The system performs read, filter, export, and delete operations on this database.   
 - Input/Output:   
 - Input: Log management actions (view, filter, export, delete) and optional search criteria.   
 - Output: Retrieved or modified archive log entries and confirmation messages.   
  
The system may also integrate with third-party tools or libraries for email validation, formatting, and logging, as required for implementation.   
  
2.4 Communication Interface   
  
The system communicates with external sources using standard communication protocols and methods, primarily via email and web-based interactions.   
  
- \*\*Email Communication\*\*:   
 - Description: The system sends and receives emails using standard email protocols (e.g., SMTP for sending, POP/IMAP for receiving).   
 - Interaction Method:   
 - For sending emails: The system validates the recipient and sends the email using SMTP.   
 - For receiving emails: The system connects to an external mail server using POP or IMAP to retrieve emails and store them in the email archive.   
 - Input/Output:   
 - Input: Incoming email data (sender, recipient, subject, body, timestamp) and outgoing email data (subject, body, sender ID, recipient ID, timestamp, formatting rules).   
 - Output: Emails sent to external recipients and emails stored in the archive after receiving.   
  
- \*\*Web-based Communication\*\*:   
 - Description: The system is accessible via web browsers and may support RESTful APIs for external applications to interact with it.   
 - Interaction Method: HTTP/HTTPS protocols are used for communication between the client (browser or application) and the server.   
 - Input/Output:   
 - Input: API requests for email and account management (e.g., POST for creating an email, GET for viewing an email archive).   
 - Output: JSON/XML responses containing email/contact/account data, status updates, and confirmation messages.   
  
- \*\*Notification Services\*\*:   
 - Description: The system sends notifications to users when emails are received or actions are completed.   
 - Interaction Method: Notifications may be delivered via in-app messages, email, or SMS, depending on user preferences and system capabilities.   
 - Input/Output:   
 - Input: Recipient information and notification content.   
 - Output: Delivered notification to the user, with a log entry for administrative visibility.   
  
The communication interfaces ensure seamless interaction with external email servers, user devices, and administrative tools, enabling efficient and secure email and user account management.

# Use Case

Use Case Name: Create Email   
Use Case ID: UC-01   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to create an email.   
3. The email archive must be accessible and in an active state.   
  
Postconditions:   
1. A new email is created and stored in the email archive.   
2. The user receives a confirmation message that the email was created successfully.   
3. If an administrator is involved, the email is also reviewed and approved.   
  
Main Flow:   
1. The user navigates to the "Create Email" section of the system.   
2. The system displays a form for entering the email details (e.g., recipient, subject, body).   
3. The user fills in the required fields in the form.   
4. The user clicks the "Submit" button.   
5. The system validates the input data (e.g., checks for missing fields, valid email format).   
6. The system creates the new email and stores it in the email archive.   
7. The system sends a confirmation message to the user.   
  
Alternative Flow:   
1. If the input data is invalid or incomplete, the system displays an error message and prompts the user to correct the input.   
2. If the user does not have permission to create an email, the system denies the request and displays an access error.   
3. If the email archive is not accessible, the system displays an error message and prevents the email from being created.   
4. If an administrator is required to review the email, the system sends the email to the administrator for approval before storing it in the archive.  
  
Use Case Name: Send Email   
Use Case ID: UC-02   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to send an email.   
3. The email archive must be accessible and in an active state.   
4. An email must already exist in the system (created via the Create Email use case).   
  
Postconditions:   
1. The email is successfully sent to the intended recipient.   
2. The email is updated in the email archive with the sent status and timestamp.   
3. The user receives a confirmation message that the email was sent.   
4. If an administrator is involved, the email must have been reviewed and approved before being sent.   
  
Main Flow:   
1. The user navigates to the "Send Email" section of the system.   
2. The system displays a list of draft emails or allows the user to select an email to send.   
3. The user selects the email to be sent.   
4. The user clicks the "Send" button.   
5. The system validates the email (e.g., checks for approval status, valid recipient address).   
6. The system sends the email to the intended recipient.   
7. The system updates the email status in the email archive to "Sent" and records the timestamp.   
8. The system sends a confirmation message to the user.   
  
Alternative Flow:   
1. If the selected email is not approved by the administrator, the system displays an error and prevents sending.   
2. If the recipient email address is invalid, the system displays an error message and prompts the user to correct it.   
3. If the user does not have permission to send emails, the system denies the request and shows an access error.   
4. If the email archive is not accessible, the system displays an error message and prevents the email from being updated or sent.  
  
Use Case Name: Receive Email   
Use Case ID: UC-03   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The email must have been sent to the user or the system must have received it from an external source.   
3. The email archive must be accessible and in an active state.   
  
Postconditions:   
1. The received email is stored in the email archive.   
2. The user is notified of the new incoming email.   
3. If the email requires administrative review, the system flags it for approval.   
4. The user can access the email in their inbox.   
  
Main Flow:   
1. The system receives an email from an external source or from another user within the system.   
2. The system validates the email format and checks for spam or security threats.   
3. The system stores the email in the email archive under the recipient's contact information.   
4. The system sends a notification to the recipient user that a new email has arrived.   
5. The user can view the email in their inbox by navigating to the "Inbox" section of the system.   
  
Alternative Flow:   
1. If the email is invalid or flagged as spam, the system discards it or moves it to a quarantine folder.   
2. If the recipient's contact information is not found, the system logs the email and displays an error message.   
3. If the email archive is not accessible, the system prevents the email from being stored and displays an error message.   
4. If an administrator needs to review the email before it is delivered, the system holds the email and sends it for approval.  
  
Use Case Name: Read Email   
Use Case ID: UC-04   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The email must exist in the email archive.   
3. The email must be accessible to the user (e.g., in the inbox or sent items).   
4. The user must have permission to read the email.   
  
Postconditions:   
1. The user views the content of the selected email.   
2. The email status is updated to "Read" in the email archive.   
3. If an administrator is involved, the system logs the read action for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Inbox" or "Sent Items" section of the system.   
2. The system displays a list of emails available to the user.   
3. The user selects an email from the list.   
4. The system retrieves the selected email from the email archive.   
5. The system displays the email content (e.g., subject, sender, body, attachments).   
6. The system updates the email status to "Read" in the email archive.   
  
Alternative Flow:   
1. If the selected email is not found or has been deleted, the system displays an error message.   
2. If the user does not have permission to read the email, the system denies access and shows an access error.   
3. If the email archive is not accessible, the system displays an error message and prevents the email from being retrieved.   
4. If the email contains restricted or confidential content, the system prompts for additional authorization before displaying it.  
  
Use Case Name: Delete Email   
Use Case ID: UC-05   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to delete an email.   
3. The email archive must be accessible and in an active state.   
4. The email to be deleted must exist in the system and be associated with the user or accessible by the administrator.   
  
Postconditions:   
1. The selected email is removed from the email archive.   
2. The user receives a confirmation message that the email was deleted.   
3. If an administrator is involved, the deletion is logged for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Inbox" or "Sent Items" section of the system.   
2. The system displays a list of emails available to the user.   
3. The user selects the email they wish to delete.   
4. The user clicks the "Delete" button.   
5. The system confirms the deletion request with the user (e.g., through a pop-up dialog).   
6. The user confirms the deletion.   
7. The system removes the selected email from the email archive.   
8. The system sends a confirmation message to the user.   
  
Alternative Flow:   
1. If the selected email is not found or has already been deleted, the system displays an error message.   
2. If the user does not have permission to delete the email, the system denies the request and displays an access error.   
3. If the email archive is not accessible, the system displays an error message and prevents the email from being deleted.   
4. If an administrator is required to approve the deletion, the system sends a request for approval before proceeding.  
  
Use Case Name: Archive Email   
Use Case ID: UC-06   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to archive an email.   
3. The email archive must be accessible and in an active state.   
4. The email to be archived must exist in the system and be accessible to the user.   
  
Postconditions:   
1. The selected email is moved to the email archive.   
2. The email is removed from the user's active inbox or sent items.   
3. The user receives a confirmation message that the email was archived.   
4. If an administrator is involved, the archive action is logged for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Inbox" or "Sent Items" section of the system.   
2. The system displays a list of emails available to the user.   
3. The user selects the email they wish to archive.   
4. The user clicks the "Archive" button.   
5. The system verifies that the email can be archived (e.g., checks permissions and archive status).   
6. The system moves the selected email to the email archive and removes it from the active view.   
7. The system sends a confirmation message to the user that the email has been archived.   
  
Alternative Flow:   
1. If the selected email is not found or has already been archived, the system displays an error message.   
2. If the user does not have permission to archive the email, the system denies the request and displays an access error.   
3. If the email archive is not accessible, the system displays an error message and prevents the email from being archived.   
4. If the email is flagged for administrative review, the system requires the administrator to approve the archive action before it is completed.  
  
Use Case Name: Recover Email   
Use Case ID: UC-07   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The email to be recovered must have been previously archived and must still exist in the email archive.   
3. The user must have permission to recover an email.   
4. The email archive must be accessible and in an active state.   
  
Postconditions:   
1. The selected email is moved from the archive to the user's active inbox or sent items.   
2. The user receives a confirmation message that the email was successfully recovered.   
3. If an administrator is involved, the recovery action is logged for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Email Archive" section of the system.   
2. The system displays a list of archived emails accessible to the user.   
3. The user selects the email they wish to recover.   
4. The user clicks the "Recover" button.   
5. The system verifies that the email can be recovered (e.g., checks permissions and archive status).   
6. The system moves the selected email to the user's active inbox or sent items and updates its status accordingly.   
7. The system sends a confirmation message to the user that the email has been recovered.   
  
Alternative Flow:   
1. If the selected email is not found or has already been deleted, the system displays an error message.   
2. If the user does not have permission to recover the email, the system denies the request and shows an access error.   
3. If the email archive is not accessible, the system displays an error message and prevents the recovery process.   
4. If the email is flagged for administrative review, the system requires the administrator to approve the recovery action before it is completed.  
  
Use Case Name: Format Email   
Use Case ID: UC-08   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to format an email.   
3. The email archive must be accessible and in an active state.   
4. An email must already exist in the system (created via the Create Email use case).   
  
Postconditions:   
1. The selected email is formatted according to the user's preferences or system standards.   
2. The formatted email is updated in the email archive.   
3. The user receives a confirmation message that the formatting was completed.   
4. If an administrator is involved, the formatting action is logged for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Format Email" section of the system.   
2. The system displays a list of draft or editable emails.   
3. The user selects an email to format.   
4. The system opens the email and displays formatting options (e.g., font, style, layout).   
5. The user applies the desired formatting changes.   
6. The user clicks the "Save Format" button.   
7. The system updates the email with the new formatting and saves it in the email archive.   
8. The system sends a confirmation message to the user that the formatting was completed.   
  
Alternative Flow:   
1. If the selected email is not found or has been deleted, the system displays an error message.   
2. If the user does not have permission to format the email, the system denies the request and shows an access error.   
3. If the email archive is not accessible, the system displays an error message and prevents the email from being updated.   
4. If the email is flagged for administrative review, the system requires the administrator to approve the formatting changes before saving.  
  
Use Case Name: Create Contact   
Use Case ID: UC-09   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to create a contact.   
3. The system must have access to the user account database.   
4. The contact information must not already exist in the system.   
  
Postconditions:   
1. A new contact is created and stored in the system.   
2. The user receives a confirmation message that the contact was created successfully.   
3. If an administrator is involved, the contact is reviewed and approved.   
  
Main Flow:   
1. The user navigates to the "Create Contact" section of the system.   
2. The system displays a form for entering contact details (e.g., name, email, phone number).   
3. The user fills in the required fields in the form.   
4. The user clicks the "Submit" button.   
5. The system validates the input data (e.g., checks for missing fields, valid email format).   
6. The system creates the new contact and stores it in the user account database.   
7. The system sends a confirmation message to the user.   
  
Alternative Flow:   
1. If the input data is invalid or incomplete, the system displays an error message and prompts the user to correct the input.   
2. If the user does not have permission to create a contact, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the contact from being created.   
4. If an administrator is required to review the contact, the system sends the contact for approval before storing it in the database.  
  
Use Case Name: Update Contact   
Use Case ID: UC-10   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to update a contact.   
3. The system must have access to the user account database.   
4. A contact must already exist in the system (created via the Create Contact use case).   
  
Postconditions:   
1. The selected contact is updated with the new information and stored in the user account database.   
2. The user receives a confirmation message that the contact was successfully updated.   
3. If an administrator is involved, the contact update is reviewed and approved.   
4. The updated contact is available for use in other email-related use cases.   
  
Main Flow:   
1. The user navigates to the "Update Contact" section of the system.   
2. The system displays a list of existing contacts or allows the user to search for a specific contact.   
3. The user selects the contact to be updated.   
4. The system opens the contact details in an editable form.   
5. The user modifies the necessary fields (e.g., name, email, phone number).   
6. The user clicks the "Save Changes" button.   
7. The system validates the updated data (e.g., checks for missing fields, valid email format).   
8. The system updates the contact in the user account database.   
9. The system sends a confirmation message to the user.   
  
Alternative Flow:   
1. If the updated data is invalid or incomplete, the system displays an error message and prompts the user to correct the input.   
2. If the user does not have permission to update the contact, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the contact from being updated.   
4. If an administrator is required to review the updated contact, the system sends the contact for approval before saving the changes.  
  
Use Case Name: Delete Contact   
Use Case ID: UC-11   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to delete a contact.   
3. The system must have access to the user account database.   
4. The contact to be deleted must exist in the system and be associated with the user or accessible by the administrator.   
  
Postconditions:   
1. The selected contact is removed from the user account database.   
2. The user receives a confirmation message that the contact was deleted.   
3. If an administrator is involved, the deletion is logged for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Contacts" section of the system.   
2. The system displays a list of contacts associated with the user.   
3. The user selects the contact they wish to delete.   
4. The user clicks the "Delete" button.   
5. The system confirms the deletion request with the user (e.g., through a pop-up dialog).   
6. The user confirms the deletion.   
7. The system removes the selected contact from the user account database.   
8. The system sends a confirmation message to the user that the contact has been deleted.   
  
Alternative Flow:   
1. If the selected contact is not found or has already been deleted, the system displays an error message.   
2. If the user does not have permission to delete the contact, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the contact from being deleted.   
4. If the administrator is required to approve the deletion, the system sends a request for approval before proceeding.  
  
Use Case Name: View Contact   
Use Case ID: UC-12   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to view a contact.   
3. The system must have access to the user account database.   
4. The contact to be viewed must exist in the system and be associated with the user or accessible by the administrator.   
  
Postconditions:   
1. The user views the details of the selected contact (e.g., name, email, phone number).   
2. The contact information remains unchanged in the system.   
3. If an administrator is involved, the system logs the view action for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Contacts" section of the system.   
2. The system displays a list of contacts associated with the user.   
3. The user selects a contact from the list.   
4. The system retrieves the contact details from the user account database.   
5. The system displays the contact information (e.g., name, email, phone number, notes).   
6. The user reviews the contact details.   
  
Alternative Flow:   
1. If the selected contact is not found or has been deleted, the system displays an error message.   
2. If the user does not have permission to view the contact, the system denies access and shows an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the contact from being retrieved.   
4. If the contact is flagged for administrative review, the system requires the administrator to approve the view action before displaying the information.  
  
Use Case Name: Manage User Account   
Use Case ID: UC-13   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to manage their own account or the administrator must have permission to manage user accounts.   
3. The system must have access to the user account database.   
4. The target user account must exist in the system.   
  
Postconditions:   
1. The user account is updated with the new information.   
2. The user receives a confirmation message that the account was successfully updated.   
3. If an administrator is involved, the change is logged for audit purposes.   
4. The updated user account is available for use in other system functionalities.   
  
Main Flow:   
1. The user or administrator navigates to the "Manage User Account" section of the system.   
2. The system displays a list of user accounts or allows the user to search for a specific account.   
3. The user selects the account to be managed.   
4. The system opens the account details in an editable form.   
5. The user modifies the necessary fields (e.g., password, email, name, role).   
6. The user clicks the "Save Changes" button.   
7. The system validates the updated data (e.g., checks for missing fields, valid email format, correct password policy).   
8. The system updates the user account in the user account database.   
9. The system sends a confirmation message to the user or administrator.   
  
Alternative Flow:   
1. If the updated data is invalid or incomplete, the system displays an error message and prompts the user to correct the input.   
2. If the user or administrator does not have permission to manage the account, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the account from being updated.   
4. If the administrator is required to approve the account update, the system sends a request for approval before saving the changes.  
  
Use Case Name: Create User Account   
Use Case ID: UC-14   
Actors: Administrator   
Preconditions:   
1. The administrator must be logged into the system.   
2. The system must have access to the user account database.   
3. The required user account details must not already exist in the system.   
  
Postconditions:   
1. A new user account is created and stored in the user account database.   
2. The administrator receives a confirmation message that the account was created successfully.   
3. The user account is available for use in other system functionalities.   
  
Main Flow:   
1. The administrator navigates to the "Create User Account" section of the system.   
2. The system displays a form for entering user account details (e.g., username, email, password, role).   
3. The administrator fills in the required fields in the form.   
4. The administrator clicks the "Submit" button.   
5. The system validates the input data (e.g., checks for missing fields, valid email format, password strength).   
6. The system creates the new user account and stores it in the user account database.   
7. The system sends a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the input data is invalid or incomplete, the system displays an error message and prompts the administrator to correct the input.   
2. If the user account database is not accessible, the system displays an error message and prevents the account from being created.   
3. If the username or email already exists, the system displays an error message and prevents duplicate account creation.   
4. If the administrator does not have permission to create user accounts, the system denies the request and shows an access error.  
  
Use Case Name: Update User Account   
Use Case ID: UC-15   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to update their own account or the administrator must have permission to update user accounts.   
3. The system must have access to the user account database.   
4. The user account to be updated must exist in the system.   
  
Postconditions:   
1. The selected user account is updated with the new information and stored in the user account database.   
2. The user or administrator receives a confirmation message that the account was successfully updated.   
3. If an administrator is involved, the update is logged for audit purposes.   
4. The updated user account is available for use in other system functionalities.   
  
Main Flow:   
1. The user or administrator navigates to the "Update User Account" section of the system.   
2. The system displays a list of user accounts or allows the user to search for a specific account.   
3. The user selects the account to be updated.   
4. The system opens the account details in an editable form.   
5. The user modifies the necessary fields (e.g., password, email, name, role).   
6. The user clicks the "Save Changes" button.   
7. The system validates the updated data (e.g., checks for missing fields, valid email format, correct password policy).   
8. The system updates the user account in the user account database.   
9. The system sends a confirmation message to the user or administrator.   
  
Alternative Flow:   
1. If the updated data is invalid or incomplete, the system displays an error message and prompts the user to correct the input.   
2. If the user or administrator does not have permission to update the account, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the account from being updated.   
4. If the administrator is required to approve the account update, the system sends a request for approval before saving the changes.  
  
Use Case Name: Delete User Account   
Use Case ID: UC-16   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to delete their own account or the administrator must have permission to delete user accounts.   
3. The system must have access to the user account database.   
4. The user account to be deleted must exist in the system.   
  
Postconditions:   
1. The selected user account is removed from the user account database.   
2. The user receives a confirmation message that the account was deleted successfully.   
3. If an administrator is involved, the deletion is logged for audit purposes.   
4. All associated data (e.g., emails, contacts) is either deleted or archived based on system policies.   
  
Main Flow:   
1. The user or administrator navigates to the "Delete User Account" section of the system.   
2. The system displays a list of user accounts or allows the user to search for a specific account.   
3. The user selects the account to be deleted.   
4. The user clicks the "Delete" button.   
5. The system confirms the deletion request with the user (e.g., through a pop-up dialog).   
6. The user confirms the deletion.   
7. The system removes the selected user account from the user account database.   
8. The system deletes or archives all data associated with the user (e.g., emails, contacts).   
9. The system sends a confirmation message to the user or administrator.   
  
Alternative Flow:   
1. If the selected user account is not found or has already been deleted, the system displays an error message.   
2. If the user or administrator does not have permission to delete the account, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the deletion process.   
4. If the administrator is required to approve the deletion, the system sends a request for approval before proceeding.  
  
Use Case Name: Manage Email Archive   
Use Case ID: UC-17   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to manage the email archive.   
3. The email archive must be accessible and in an active state.   
4. The system must have access to the email archive database.   
  
Postconditions:   
1. The email archive is updated based on the user's or administrator's actions (e.g., emails are added, removed, or reorganized).   
2. The user receives a confirmation message that the archive management was successful.   
3. If an administrator is involved, all actions are logged for audit purposes.   
4. The updated archive remains available for future use in email-related operations.   
  
Main Flow:   
1. The user or administrator navigates to the "Manage Email Archive" section of the system.   
2. The system displays the contents of the email archive, including options to organize, filter, or modify it.   
3. The user or administrator selects an action (e.g., reorganize folders, set retention policies, add or remove emails).   
4. The system performs the selected action, updating the archive accordingly.   
5. The system sends a confirmation message indicating the archive has been successfully managed.   
  
Alternative Flow:   
1. If the user or administrator does not have permission to manage the archive, the system denies the request and displays an access error.   
2. If the email archive is not accessible, the system displays an error message and prevents the archive from being modified.   
3. If the selected action cannot be completed due to system constraints (e.g., invalid folder structure), the system displays an error and provides guidance.   
4. If an administrator is required to approve changes to the archive, the system holds the action until approval is received.  
  
Use Case Name: View Email Archive   
Use Case ID: UC-07   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to view the email archive.   
3. The email archive must be accessible and in an active state.   
4. The email archive must contain at least one email for the user to view.   
  
Postconditions:   
1. The user views the contents of the email archive.   
2. The system displays the selected emails in a readable format (e.g., list, search results).   
3. The email archive remains unchanged after the view operation.   
4. If an administrator is involved, the view action is logged for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Email Archive" section of the system.   
2. The system displays a list of archived emails accessible to the user.   
3. The user selects a specific email or applies filters/search criteria.   
4. The system retrieves the selected email(s) from the email archive.   
5. The system displays the retrieved email(s) with relevant details (e.g., subject, sender, date, status).   
6. The user reviews the displayed email(s).   
  
Alternative Flow:   
1. If the email archive is empty or does not contain any emails for the user, the system displays a message indicating no archived emails are available.   
2. If the user does not have permission to view the email archive, the system denies the request and displays an access error.   
3. If the email archive is not accessible, the system displays an error message and prevents the view operation.   
4. If the selected email is flagged for administrative review, the system requires the administrator to approve the view action before displaying the information.  
  
Use Case Name: Assign Administrator   
Use Case ID: UC-18   
Actors: Administrator   
Preconditions:   
1. The administrator must be logged into the system.   
2. The system must have access to the user account database.   
3. The user account to be assigned as an administrator must exist in the system.   
4. The requesting administrator must have the authority to assign administrative roles.   
  
Postconditions:   
1. The selected user is assigned the role of Administrator in the system.   
2. The user account database is updated to reflect the new role.   
3. The system logs the assignment for audit purposes.   
4. The assigned user receives a notification of their new administrative privileges.   
  
Main Flow:   
1. The administrator navigates to the "Assign Administrator" section of the system.   
2. The system displays a list of existing user accounts or allows the administrator to search for a specific user.   
3. The administrator selects the user to be assigned as an administrator.   
4. The system confirms the selection and opens a role assignment interface.   
5. The administrator confirms the assignment of the "Administrator" role.   
6. The system updates the user account in the database to reflect the new role.   
7. The system logs the action in the audit trail.   
8. The system sends a confirmation message to the administrator.   
9. The system sends a notification to the assigned user.   
  
Alternative Flow:   
1. If the selected user does not exist or has already been assigned as an administrator, the system displays an error message.   
2. If the requesting administrator does not have permission to assign roles, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the assignment.   
4. If the system requires additional approval (e.g., by a higher-level administrator), the system holds the assignment until approval is received.  
  
Use Case Name: Manage Administrator Permissions   
Use Case ID: UC-19   
Actors: Administrator   
Preconditions:   
1. The administrator must be logged into the system.   
2. The administrator must have the authority to manage other administrators' permissions.   
3. The system must have access to the user account database.   
4. The target administrator account must exist in the system.   
  
Postconditions:   
1. The permissions of the target administrator are updated according to the specified changes.   
2. The user account database is updated to reflect the new permissions.   
3. The system logs the permission change for audit purposes.   
4. The updated permissions are immediately enforced in the system.   
  
Main Flow:   
1. The administrator navigates to the "Manage Administrator Permissions" section of the system.   
2. The system displays a list of existing administrator accounts or allows the administrator to search for a specific account.   
3. The administrator selects the target administrator account.   
4. The system opens the account details, showing the current permissions assigned.   
5. The administrator modifies the permissions (e.g., grant or revoke access to specific functions).   
6. The administrator clicks the "Save Permissions" button.   
7. The system validates the changes (e.g., ensures no invalid permissions are assigned).   
8. The system updates the user account database with the new permissions.   
9. The system logs the change in the audit trail.   
10. The system sends a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected administrator account is not found or has already been deleted, the system displays an error message.   
2. If the requesting administrator does not have permission to manage permissions, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the permission change.   
4. If the system requires approval for certain permission changes (e.g., by a higher-level administrator), the system holds the change until approval is received.  
  
Use Case Name: Delete Administrator   
Use Case ID: UC-20   
Actors: Administrator   
Preconditions:   
1. The administrator must be logged into the system.   
2. The administrator must have permission to delete other administrators.   
3. The system must have access to the user account database.   
4. The administrator account to be deleted must exist in the system.   
  
Postconditions:   
1. The selected administrator account is removed from the user account database.   
2. The administrator receives a confirmation message that the account was deleted successfully.   
3. The system logs the deletion for audit purposes.   
4. All administrative privileges associated with the deleted account are revoked.   
  
Main Flow:   
1. The administrator navigates to the "Delete Administrator" section of the system.   
2. The system displays a list of existing administrator accounts or allows the administrator to search for a specific account.   
3. The administrator selects the account to be deleted.   
4. The administrator clicks the "Delete" button.   
5. The system confirms the deletion request with the administrator (e.g., through a pop-up dialog).   
6. The administrator confirms the deletion.   
7. The system removes the selected administrator account from the user account database.   
8. The system logs the deletion action.   
9. The system sends a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected administrator account is not found or has already been deleted, the system displays an error message.   
2. If the requesting administrator does not have permission to delete other administrators, the system denies the request and displays an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the deletion process.   
4. If the system requires additional approval for the deletion (e.g., from a higher-level administrator), the system holds the action until approval is received.  
  
Use Case Name: View Administrator Information   
Use Case ID: UC-21   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to view administrator information.   
3. The system must have access to the user account database.   
4. The administrator account to be viewed must exist in the system.   
  
Postconditions:   
1. The user views the details of the selected administrator (e.g., name, email, role).   
2. The administrator information remains unchanged in the system.   
3. If an administrator is involved, the system logs the view action for audit purposes.   
  
Main Flow:   
1. The user navigates to the "Administrator Information" section of the system.   
2. The system displays a list of administrator accounts or allows the user to search for a specific administrator.   
3. The user selects the administrator to view.   
4. The system retrieves the administrator's details from the user account database.   
5. The system displays the information (e.g., name, email, role, date assigned).   
6. The user reviews the displayed administrator details.   
  
Alternative Flow:   
1. If the selected administrator account is not found or has been deleted, the system displays an error message.   
2. If the user does not have permission to view administrator information, the system denies access and shows an access error.   
3. If the user account database is not accessible, the system displays an error message and prevents the administrator information from being retrieved.   
4. If the administrator account is flagged for administrative review, the system requires the administrator to approve the view action before displaying the information.  
  
Use Case Name: Manage Email Folder   
Use Case ID: UC-22   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to manage email folders.   
3. The email archive must be accessible and in an active state.   
4. The system must have access to the email folder database.   
  
Postconditions:   
1. The email folder is updated based on the user's or administrator's actions (e.g., folders are created, renamed, deleted, or emails are moved).   
2. The user receives a confirmation message that the folder management was successful.   
3. If an administrator is involved, all actions are logged for audit purposes.   
4. The updated folder structure remains available for future use in email-related operations.   
  
Main Flow:   
1. The user or administrator navigates to the "Manage Email Folder" section of the system.   
2. The system displays the current folder structure and provides options to create, rename, delete folders, or move emails between folders.   
3. The user or administrator selects the desired folder management action (e.g., create new folder, delete folder, move email).   
4. The system prompts for additional input, if necessary (e.g., folder name, target folder for moving emails).   
5. The user or administrator provides the required information.   
6. The system validates the input and performs the selected action (e.g., creates a new folder, updates folder names, moves emails).   
7. The system updates the email archive and folder structure accordingly.   
8. The system logs the action in the archive log, if an administrator is involved.   
9. The system sends a confirmation message to the user or administrator that the folder management was completed.   
  
Alternative Flow:   
1. If the user or administrator does not have permission to manage the folder, the system denies the request and displays an access error.   
2. If the email archive or folder database is not accessible, the system displays an error message and prevents the folder management action.   
3. If the selected folder or email is not found, the system displays an error message and prompts the user or administrator to correct the selection.   
4. If the system requires administrative approval for certain actions (e.g., deleting a folder with important emails), the system holds the action until approval is received.  
  
Use Case Name: Manage Email Format   
Use Case ID: UC-23   
Actors: User, Administrator   
Preconditions:   
1. The user must be logged into the system.   
2. The user must have permission to manage email formats.   
3. The email archive must be accessible and in an active state.   
4. The system must have access to the email format database or configuration settings.   
  
Postconditions:   
1. The email format is updated or customized according to the user's or administrator's specifications.   
2. The system stores the updated email format configuration.   
3. The user receives a confirmation message that the email format management was successful.   
4. If an administrator is involved, all actions are logged for audit purposes.   
  
Main Flow:   
1. The user or administrator navigates to the "Manage Email Format" section of the system.   
2. The system displays the current email format settings or templates available for modification.   
3. The user or administrator selects the desired action (e.g., create a new format, edit an existing format, delete a format).   
4. The system opens the selected format in an editable interface and displays available formatting options (e.g., signature, layout, branding, font styles).   
5. The user or administrator modifies the format as needed.   
6. The user or administrator clicks the "Save Format" or "Apply Changes" button.   
7. The system validates the format configuration (e.g., checks for invalid settings or conflicts).   
8. The system updates the email format configuration in the database or settings.   
9. The system logs the action in the archive log if an administrator is involved.   
10. The system sends a confirmation message to the user or administrator that the format has been successfully managed.   
  
Alternative Flow:   
1. If the user or administrator does not have permission to manage email formats, the system denies the request and displays an access error.   
2. If the email archive or format database is not accessible, the system displays an error message and prevents the format from being updated.   
3. If the selected format is not found or has already been deleted, the system displays an error message and prompts the user or administrator to select a valid format.   
4. If the system requires administrative approval for certain format changes (e.g., modifying a system-wide template), the system holds the action until approval is received.  
  
Use Case Name: Manage Archive Log   
Use Case ID: UC-24   
Actors: Administrator   
Preconditions:   
1. The administrator must be logged into the system.   
2. The administrator must have permission to manage the archive log.   
3. The system must have access to the archive log database.   
4. The archive log must contain records for viewing, filtering, exporting, or deleting.   
  
Postconditions:   
1. The archive log is updated or modified based on the administrator's actions (e.g., entries are viewed, filtered, exported, or deleted).   
2. The administrator receives a confirmation message that the archive log management was successful.   
3. All changes to the archive log are logged for audit purposes.   
4. The archive log remains available for future reference and system auditing.   
  
Main Flow:   
1. The administrator navigates to the "Manage Archive Log" section of the system.   
2. The system displays a list of archive log entries with details such as timestamp, actor, action performed, and affected data.   
3. The administrator selects an action to perform on the archive log (e.g., view specific entries, filter by date or user, export the log, or delete outdated entries).   
4. The system provides a filtering or search interface if the administrator chooses to view or filter entries.   
5. The administrator specifies criteria (e.g., date range, user, action type) and submits the request.   
6. The system retrieves the relevant archive log entries and displays them.   
7. If the administrator chooses to export the log, the system generates a report in the requested format (e.g., CSV, PDF) and provides it for download.   
8. If the administrator chooses to delete entries, the system confirms the deletion request and removes the specified entries from the archive log.   
9. The system updates the archive log and logs the administrator's management action.   
10. The system sends a confirmation message to the administrator that the archive log has been successfully managed.   
  
Alternative Flow:   
1. If the administrator does not have permission to manage the archive log, the system denies the request and displays an access error.   
2. If the archive log database is not accessible, the system displays an error message and prevents any modifications or viewing of the log.   
3. If no archive log entries match the administrator's search or filter criteria, the system displays a message indicating no results found.   
4. If the system requires additional approval for certain log management actions (e.g., deleting entries), the system holds the action until approval is received.