项目文档

# Functional Requirement

# Chapter 1: Functional Requirements  
  
## 1.1 Create User Account Function   
Function ID: FR-01   
Description: The system allows an Administrator to create a new User Account, including sending a confirmation email. The email address must be unique and valid.   
Input: User's name, contact information, email address, and optional password.   
Output: A new User Account in the system, confirmation email sent, and success message displayed to the Administrator.  
  
## 1.2 Delete User Account Function   
Function ID: FR-02   
Description: The system allows an Administrator to delete a User Account, which includes marking the email as deleted in the Email Archive database and sending a confirmation.   
Input: Email address of the User to be deleted.   
Output: The User Account is deleted from the system, associated email is marked as deleted in the Email Archive, and a confirmation message is displayed.  
  
## 1.3 View User Account Information Function   
Function ID: FR-03   
Description: The system allows an Administrator to view detailed information of a specific User Account and associated email data from the Email Archive.   
Input: Email address of the User whose information is to be viewed.   
Output: Display of the User Account details and associated email history.  
  
## 1.4 Update User Account Settings Function   
Function ID: FR-04   
Description: The system allows an Administrator to modify a User Account's settings such as password or contact information and update the Email Archive database accordingly.   
Input: Email address of the User, updated settings (e.g., new password, contact info).   
Output: Updated User Account in the system, associated email information updated in the Email Archive, and a confirmation message.  
  
## 1.5 Send Email Function   
Function ID: FR-05   
Description: The system allows an Administrator to compose and send an email, and logs the email in the Email Archive database.   
Input: Recipient's email address, subject, message content.   
Output: The email is sent to the recipient, and the details are recorded in the Email Archive.  
  
## 1.6 Receive Email Function   
Function ID: FR-06   
Description: The system receives incoming emails from the Email Server, validates the sender, and stores the email in the Email Archive database.   
Input: Incoming email from the Email Server (sender, recipient, subject, content).   
Output: The email is stored in the Email Archive, categorized, and logged for audit purposes.  
  
## 1.7 Format Email Content Function   
Function ID: FR-07   
Description: The system allows an Administrator to format the content of an email using predefined templates, fonts, and signatures.   
Input: Email content (either from an existing email or entered manually).   
Output: Formatted email content ready for sending or saving, and a confirmation message.  
  
## 1.8 View Email Function   
Function ID: FR-08   
Description: The system allows an Administrator to view the full content of an email stored in the Email Archive database.   
Input: Email identifier or selection from the Email Archive.   
Output: Display of the email content (sender, recipient, subject, timestamp, body), and logging of the viewing action.  
  
## 1.9 Delete Email Function   
Function ID: FR-09   
Description: The system allows an Administrator to delete an email from the Email Archive database after confirmation.   
Input: Email identifier or selection from the Email Archive.   
Output: The selected email is deleted from the Email Archive, the action is logged, and a confirmation message is displayed.  
  
## 1.10 Archive Email Function   
Function ID: FR-10   
Description: The system allows an Administrator to archive an email from the active email system to the Email Archive database.   
Input: Email identifier or selection from the active email system.   
Output: The email is archived in the Email Archive database, its status is updated, and the action is logged.  
  
## 1.11 Recover Archived Email Function   
Function ID: FR-11   
Description: The system allows an Administrator to recover an archived email and move it back to the active email system.   
Input: Email identifier or selection from the Email Archive.   
Output: The email is recovered and marked as active, the action is logged, and a confirmation message is displayed.  
  
## 1.12 Create Contact Function   
Function ID: FR-12   
Description: The system allows an Administrator to create a new Contact with associated email information, ensuring the email is valid and not already linked to another Contact.   
Input: Contact name, email address, phone number, and address.   
Output: A new Contact is created and stored in the Contact database, and the email is linked and validated.  
  
## 1.13 Delete Contact Function   
Function ID: FR-13   
Description: The system allows an Administrator to delete a Contact and update the Email Archive database to reflect the deletion.   
Input: Email address of the Contact to be deleted.   
Output: The Contact is deleted from the Contact database, the associated email is updated in the Email Archive, and a confirmation message is displayed.  
  
## 1.14 View Contact Function   
Function ID: FR-14   
Description: The system allows an Administrator to view the full details of a specific Contact and its associated email history.   
Input: Email address of the Contact to be viewed.   
Output: Display of the Contact's details and associated email information from the Email Archive.  
  
## 1.15 Update Contact Information Function   
Function ID: FR-15   
Description: The system allows an Administrator to modify the information of an existing Contact and update the Email Archive database accordingly.   
Input: Email address of the Contact to be updated and the new information.   
Output: Updated Contact in the Contact database, associated email information updated in the Email Archive, and a confirmation message.  
  
## 1.16 Manage Email Archive Function   
Function ID: FR-16   
Description: The system allows an Administrator to perform archive-related actions (view, recover, or delete) on archived emails.   
Input: Email identifier or selection from the Email Archive.   
Output: The Email Archive is updated based on the action taken, the status of the email is modified, and the action is logged.  
  
## 1.17 View Email Archive Function   
Function ID: FR-17   
Description: The system allows an Administrator to view a list of archived emails and the full content of selected emails.   
Input: Access to the Email Archive database.   
Output: Display of the list of archived emails and the content of selected emails, with logging of the viewing action.  
  
## 1.18 Manage Administrator Permissions Function   
Function ID: FR-18   
Description: The system allows a Super Administrator to assign, modify, or revoke permissions for other Administrators or Users.   
Input: Email address of the user or role, permission level to be assigned or modified.   
Output: Updated Permission Setting in the database, confirmation message, and logging of the action.  
  
## 1.19 Manage Email Draft Function   
Function ID: FR-19   
Description: The system allows an Administrator to manage email drafts, including saving, editing, deleting, or sending them.   
Input: Draft identifier or content of the draft email.   
Output: Updated or deleted draft in the Email Draft database, and if sent, the email is stored in the Email Archive.  
  
## 1.20 Manage Contact Group Function   
Function ID: FR-20   
Description: The system allows an Administrator to manage Contact Groups, including creating, updating, deleting, or viewing groups and their associated contacts.   
Input: Group name, selected contacts to be added or removed from the group.   
Output: Updated Contact Group in the database, associated contacts updated, and a confirmation message.  
  
## 1.21 Manage Email Folder Function   
Function ID: FR-21   
Description: The system allows an Administrator to manage Email Folders, including creating, updating, deleting, or associating emails with folders.   
Input: Folder name, selected emails to be associated with the folder.   
Output: Updated Email Folder in the database, associated emails updated, and a confirmation message.  
  
## 1.22 Manage Email Archive Record Function   
Function ID: FR-22   
Description: The system allows an Administrator to manage Email Archive Records, including creating, updating, or deleting them.   
Input: Email identifier, archive date, recovery date, and status.   
Output: Updated Email Archive Record in the database, associated email data updated, and a confirmation message.  
  
## 1.23 Create Administrator Function   
Function ID: FR-23   
Description: The system allows a Super Administrator to create a new Administrator account with the appropriate permissions and roles.   
Input: Name, contact information, and email address of the new Administrator.   
Output: New Administrator account created in the User Account database, email validated and stored in the Email Archive, and a confirmation email sent.  
  
## 1.24 Delete Administrator Function   
Function ID: FR-24   
Description: The system allows a Super Administrator to delete an Administrator account, ensuring that at least one Super Administrator remains for system integrity.   
Input: Email address of the Administrator to be deleted.   
Output: The Administrator account is deleted, associated email is marked as deleted in the Email Archive, and a confirmation message is displayed.

# External Description

## Chapter 2: External Interfaces   
  
This chapter defines the external interfaces that the system interacts with. These interfaces include user interfaces, hardware interfaces (if any), software interfaces, and communication interfaces. Each interface is described in detail, including its role, interaction method, and the data it exchanges with the system.   
  
---  
  
### 2.1 User Interface   
  
The user interface is the primary method of interaction between the system and the Administrator (or Super Administrator). It supports all administrative actions and provides feedback on the outcomes of those actions.   
  
#### 2.1.1 Account Management Interface   
- \*\*Description\*\*: A screen-based interface where Administrators can create, delete, view, or update User Accounts and Administrator Accounts.   
- \*\*Inputs\*\*:   
 - User's name, contact information, email address, optional password.   
 - Email address of the User or Administrator to be deleted or updated.   
 - Updated settings (e.g., new password, contact info).   
- \*\*Outputs\*\*:   
 - Success or confirmation messages for account creation, deletion, or update.   
 - Display of detailed account information (name, contact info, email).   
  
#### 2.1.2 Email Management Interface   
- \*\*Description\*\*: An interface that allows Administrators to send, receive, view, delete, format, and archive emails.   
- \*\*Inputs\*\*:   
 - Recipient's email address, subject, message content.   
 - Email identifier or selection from the Email Archive or active email system.   
 - Email content to be formatted using templates, fonts, or signatures.   
- \*\*Outputs\*\*:   
 - Display of email content (sender, recipient, subject, timestamp, body).   
 - Confirmation messages for sending, deleting, archiving, or recovering emails.   
  
#### 2.1.3 Contact Management Interface   
- \*\*Description\*\*: A screen-based interface for creating, deleting, viewing, and updating Contacts.   
- \*\*Inputs\*\*:   
 - Contact name, email address, phone number, address.   
 - Email address of the Contact to be deleted or updated.   
 - New information for a Contact.   
- \*\*Outputs\*\*:   
 - Display of Contact details and associated email history.   
 - Confirmation messages for creating, deleting, or updating Contacts.   
  
#### 2.1.4 Email Archive Management Interface   
- \*\*Description\*\*: A user interface for managing Email Archive Records, including viewing, deleting, or recovering archived emails.   
- \*\*Inputs\*\*:   
 - Email identifier or selection from the Email Archive.   
 - Archive date, recovery date, and status for an Email Archive Record.   
- \*\*Outputs\*\*:   
 - Display of archived emails and their content.   
 - Confirmation messages for actions on Email Archive Records.   
  
#### 2.1.5 Email Draft Management Interface   
- \*\*Description\*\*: An interface that allows Administrators to save, edit, delete, or send draft emails.   
- \*\*Inputs\*\*:   
 - Draft identifier or content of the draft email.   
- \*\*Outputs\*\*:   
 - Updated or deleted draft in the system.   
 - Confirmation message if the draft is sent.   
  
#### 2.1.6 Contact Group Management Interface   
- \*\*Description\*\*: A user interface for managing Contact Groups, including creating, updating, deleting, or viewing groups and their associated contacts.   
- \*\*Inputs\*\*:   
 - Group name.   
 - Contacts to be added or removed from the group.   
- \*\*Outputs\*\*:   
 - Updated Contact Group in the system.   
 - Confirmation message for changes made.   
  
#### 2.1.7 Email Folder Management Interface   
- \*\*Description\*\*: An interface for managing Email Folders, such as creating, updating, deleting, or associating emails with folders.   
- \*\*Inputs\*\*:   
 - Folder name.   
 - Emails to be associated with the folder.   
- \*\*Outputs\*\*:   
 - Updated Email Folder in the system.   
 - Confirmation message for changes made.   
  
#### 2.1.8 Permission Management Interface   
- \*\*Description\*\*: A screen-based interface that allows a Super Administrator to manage permissions for other Administrators or Users.   
- \*\*Inputs\*\*:   
 - Email address of the user or role.   
 - Permission level to be assigned or modified.   
- \*\*Outputs\*\*:   
 - Updated permission settings in the system.   
 - Confirmation message for the change in permissions.   
  
---  
  
### 2.2 Hardware Interface   
  
\*\*No hardware interfaces are required for this system.\*\*   
  
---  
  
### 2.3 Software Interface   
  
#### 2.3.1 Email Archive Database   
- \*\*Description\*\*: A relational database used to store all email records, including archived emails, their statuses, timestamps, and any associated metadata.   
- \*\*Interaction Method\*\*:   
 - The system interacts with the Email Archive database using SQL queries or ORM (Object-Relational Mapping) to perform create, read, update, and delete (CRUD) operations.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Email content, timestamps, statuses, archive/recovery dates.   
 - \*\*Outputs\*\*: Stored and retrieved email records, status updates, audit logs.   
  
#### 2.3.2 User Account Database   
- \*\*Description\*\*: A database that stores user account information, including names, contact details, email addresses, and passwords.   
- \*\*Interaction Method\*\*:   
 - The system uses SQL or ORM to manage user accounts, ensuring uniqueness of email addresses and validation of inputs.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: User name, contact information, email, password.   
 - \*\*Outputs\*\*: Stored user account details, confirmation of account creation/deletion, audit logs.   
  
#### 2.3.3 Contact Database   
- \*\*Description\*\*: A database that stores contact information, including names, email addresses, phone numbers, and addresses.   
- \*\*Interaction Method\*\*:   
 - The system uses SQL or ORM to manage Contact records and ensure that email addresses are not duplicated across contacts.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Contact name, email address, phone number, address.   
 - \*\*Outputs\*\*: Stored and retrieved Contact details, audit logs, status updates for deleted or updated contacts.   
  
#### 2.3.4 Email Draft Database   
- \*\*Description\*\*: A database to store unsent email drafts for Administrators.   
- \*\*Interaction Method\*\*:   
 - The system uses SQL or ORM to save, update, or delete draft emails.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Draft content, draft identifier.   
 - \*\*Outputs\*\*: Updated or deleted draft, audit logs.   
  
#### 2.3.5 Email Server   
- \*\*Description\*\*: An external email server responsible for sending and receiving emails.   
- \*\*Interaction Method\*\*:   
 - The system uses SMTP (Simple Mail Transfer Protocol) to send emails and POP3 or IMAP to receive emails.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Recipient’s email, subject, content (for sending).   
 - \*\*Outputs\*\*: Incoming emails with sender, recipient, subject, and content (for receiving).   
  
#### 2.3.6 Email Formatting Engine   
- \*\*Description\*\*: A component or library used to apply formatting (e.g., templates, fonts, signatures) to email content.   
- \*\*Interaction Method\*\*:   
 - The system integrates with the formatting engine through predefined functions or APIs to generate formatted email content.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Raw email content.   
 - \*\*Outputs\*\*: Formatted email content ready for sending or saving.   
  
---  
  
### 2.4 Communication Interface   
  
#### 2.4.1 Email Communication Interface   
- \*\*Description\*\*: This interface enables the system to send and receive emails via the Email Server.   
- \*\*Interaction Method\*\*:   
 - \*\*Sending Emails\*\*: Uses SMTP to deliver emails to recipients.   
 - \*\*Receiving Emails\*\*: Uses POP3 or IMAP to fetch and validate incoming emails.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Email content (subject, body, recipient), sender validation for incoming emails.   
 - \*\*Outputs\*\*: Emails sent to recipients, incoming emails stored in the Email Archive.   
  
#### 2.4.2 Confirmation Email Interface   
- \*\*Description\*\*: A communication interface for sending confirmation emails to users or administrators for account creation, deletion, or recovery actions.   
- \*\*Interaction Method\*\*:   
 - The system uses the same SMTP-based mechanism for sending confirmation emails.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Recipient’s email address, confirmation message content.   
 - \*\*Outputs\*\*: Confirmation email delivered to the recipient.   
  
#### 2.4.3 Web Communication Interface   
- \*\*Description\*\*: This interface supports system access through a web browser, enabling remote administration and management.   
- \*\*Interaction Method\*\*:   
 - The system is accessible via a web-based user interface using standard HTTP/HTTPS protocols.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: User credentials, administrative actions (e.g., send email, update account).   
 - \*\*Outputs\*\*: Web-based responses (e.g., success/failure messages, user interface updates).   
  
#### 2.4.4 API Interface for External Integration (if applicable)   
- \*\*Description\*\*: A potential API interface that allows external systems to interact with the Email Archive or User Account databases (if integration with other systems is required).   
- \*\*Interaction Method\*\*:   
 - RESTful API or GraphQL for external system integration.   
- \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: API requests for retrieving or updating data (e.g., GET/POST/PUT/DELETE methods).   
 - \*\*Outputs\*\*: JSON or XML formatted responses with success/failure status and relevant data.   
  
---  
  
### Summary of External Interfaces   
  
| Interface Name | Type | Description |  
|-----------------------------------------|-----------------------|-----------------------------------------------------------------------------|  
| Account Management Interface | User Interface | Supports user account creation, deletion, viewing, and updating. |  
| Email Management Interface | User Interface | Allows sending, receiving, formatting, and managing emails. |  
| Email Archive Management Interface | User Interface | Interface for managing archived emails and their statuses. |  
| Contact Management Interface | User Interface | Enables creating, deleting, viewing, and updating contacts. |  
| Email Draft Management Interface | User Interface | Manages unsent email drafts including saving, editing, and sending. |  
| Contact Group Management Interface | User Interface | Manages groups of contacts including creation, deletion, and updates. |  
| Email Folder Management Interface | User Interface | Manages folders for organizing emails. |  
| Permission Management Interface | User Interface | Allows Super Administrators to assign or revoke permissions. |  
| Email Archive Database | Software Interface | Stores and manages archived email records. |  
| User Account Database | Software Interface | Stores and manages user account details. |  
| Contact Database | Software Interface | Stores and manages contact information. |  
| Email Draft Database | Software Interface | Stores draft emails before sending. |  
| Email Server | Communication Interface | Used for sending and receiving emails via SMTP, POP3, or IMAP. |  
| Email Formatting Engine | Software Interface | Applies formatting (templates, fonts, signatures) to email content. |  
| Confirmation Email Interface | Communication Interface | Sends confirmation emails to users or administrators for account-related actions. |  
| Web Communication Interface | Communication Interface | Provides remote access to the system via a web browser. |  
| API Interface for External Integration | Communication Interface | Optional interface for external systems to interact with the Email Archive or User Account database. |  
  
---  
  
This section provides a clear and comprehensive overview of all external interfaces that the system interacts with. It ensures that developers have a complete understanding of the system's dependencies and communication methods.

# Use Case

Use Case Name: Create User Account   
Use Case ID: UC-01   
Actors: Administrator, User, Email Archive System   
Preconditions:   
1. The Administrator has access to the system and is authenticated.   
2. The system has access to the Email Archive database.   
3. The system is connected to the Email server for sending confirmation emails.   
  
Postconditions:   
1. A new User Account is successfully created in the system.   
2. A confirmation email is sent to the provided email address.   
3. The User can log in using the newly created account.   
  
Main Flow:   
1. The Administrator navigates to the "Create User Account" option in the system.   
2. The system prompts the Administrator to enter the User's contact information and email address.   
3. The Administrator enters the necessary details (name, contact, email) for the new User.   
4. The system verifies the email address by checking the Email Archive database for existing entries.   
5. If the email is unique and valid, the system generates a new User Account with a default password or prompts the Administrator to set one.   
6. The system sends a confirmation email to the User's provided email address.   
7. The system displays a success message to the Administrator, indicating that the account was created and the confirmation email was sent.   
  
Alternative Flow:   
1. If the email address already exists in the Email Archive database, the system displays an error message and prompts the Administrator to provide a different email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to send the confirmation email, it logs the error and displays a message to the Administrator indicating the failure, allowing them to retry or manually send the confirmation.  
  
Use Case Name: Delete User Account   
Use Case ID: UC-02   
Actors: Administrator, User Account System, Email Archive System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to delete user accounts.   
2. The system has access to the User Account database.   
3. The system is connected to the Email Archive database for verification purposes.   
  
Postconditions:   
1. The specified User Account is deleted from the system.   
2. The associated email address is marked as deleted in the Email Archive database.   
3. A confirmation email is sent to the User (if applicable) and/or the Administrator.   
  
Main Flow:   
1. The Administrator navigates to the "Delete User Account" option in the system.   
2. The system prompts the Administrator to enter the email address of the User whose account is to be deleted.   
3. The Administrator inputs the email address.   
4. The system verifies the email address exists in the User Account database.   
5. The system confirms the deletion with the Administrator (e.g., through a pop-up dialog or checkbox).   
6. The system deletes the User Account and marks the email as deleted in the Email Archive database.   
7. The system sends a confirmation email to the User (if the email server is available) and/or displays a success message to the Administrator.   
  
Alternative Flow:   
1. If the email address does not exist in the User Account database, the system displays an error message and prompts the Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to send the confirmation email, it logs the error and displays a message to the Administrator indicating the failure, allowing them to retry or manually send the confirmation.  
  
Use Case Name: View User Account Information   
Use Case ID: UC-03   
Actors: Administrator, User Account System, Email Archive System   
Preconditions:   
1. The Administrator is authenticated and logged into the system.   
2. The system has access to the User Account database.   
3. The system is connected to the Email Archive database for cross-referencing information.   
  
Postconditions:   
1. The Administrator is shown the User Account information.   
2. The associated email information is retrieved and displayed from the Email Archive database.   
3. The system maintains the integrity and confidentiality of the data displayed.   
  
Main Flow:   
1. The Administrator navigates to the "View User Account Information" option in the system.   
2. The system prompts the Administrator to enter the email address of the User whose information is to be viewed.   
3. The Administrator inputs the email address.   
4. The system verifies that the email exists in the User Account database.   
5. The system retrieves the User Account details and cross-references the Email Archive database for additional email-related information.   
6. The system displays the User Account information and associated email history to the Administrator.   
  
Alternative Flow:   
1. If the email address does not exist in the User Account database, the system displays an error message and prompts the Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to retrieve email-related information from the Email Archive database, it logs the error and displays a message to the Administrator indicating the issue.  
  
Use Case Name: Update User Account Settings   
Use Case ID: UC-04   
Actors: Administrator, User Account System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has access to the system.   
2. The system has access to the User Account database.   
3. The system is connected to the Email Archive database for verification purposes.   
  
Postconditions:   
1. The User Account settings are updated successfully.   
2. The associated email information is updated or verified in the Email Archive database.   
3. A confirmation email is sent to the User (if applicable) and/or the Administrator.   
  
Main Flow:   
1. The Administrator navigates to the "Update User Account Settings" option in the system.   
2. The system prompts the Administrator to enter the email address of the User whose settings are to be updated.   
3. The Administrator inputs the email address.   
4. The system verifies the email exists in the User Account database.   
5. The system displays the current settings of the User Account.   
6. The Administrator modifies the required settings (e.g., password, contact information).   
7. The system updates the User Account with the new settings and verifies the email address in the Email Archive database.   
8. The system sends a confirmation email to the User (if the email server is available) and/or displays a success message to the Administrator.   
  
Alternative Flow:   
1. If the email address does not exist in the User Account database, the system displays an error message and prompts the Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to send the confirmation email, it logs the error and displays a message to the Administrator indicating the failure, allowing them to retry or manually send the confirmation.  
  
Use Case Name: Send Email   
Use Case ID: UC-05   
Actors: Administrator, Email Archive System, Email Server   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to send emails.   
2. The system is connected to the Email Archive database for email verification.   
3. The system is connected to the Email server to facilitate the sending of emails.   
  
Postconditions:   
1. The email is successfully sent to the intended recipient.   
2. The email details are recorded in the Email Archive database.   
3. The Administrator receives a confirmation message indicating the success or failure of the email delivery.   
  
Main Flow:   
1. The Administrator navigates to the "Send Email" option in the system.   
2. The system prompts the Administrator to enter the recipient's email address, subject, and message content.   
3. The Administrator inputs the required information.   
4. The system verifies the recipient's email address by checking the Email Archive database for validity and existence.   
5. If the email address is valid, the system prepares the email for sending.   
6. The system sends the email via the connected Email server.   
7. The system logs the email in the Email Archive database with a timestamp and status.   
8. The system displays a success message to the Administrator confirming the email was sent.   
  
Alternative Flow:   
1. If the email address does not exist in the Email Archive database, the system displays an error message and prompts the Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to send the email, it logs the error and displays a message to the Administrator indicating the failure, allowing them to retry or investigate the issue.  
  
Use Case Name: Receive Email   
Use Case ID: UC-06   
Actors: Email Archive System, Email Server, Administrator   
Preconditions:   
1. The system is connected to the Email Server to receive incoming emails.   
2. The Email Archive database is accessible and ready to store received emails.   
3. The system is configured to process and categorize emails automatically or manually.   
  
Postconditions:   
1. The incoming email is successfully received and stored in the Email Archive database.   
2. The email is categorized or flagged appropriately for further processing.   
3. The Administrator is notified of the receipt of the email, if required.   
  
Main Flow:   
1. The system detects an incoming email from the Email Server.   
2. The system validates the sender's email address and content.   
3. The system stores the email in the Email Archive database with a timestamp and status.   
4. The system categorizes the email based on predefined rules (e.g., spam, urgent, regular).   
5. If the email is associated with a User Account, the system updates the user's email history.   
6. The system sends a notification to the Administrator (if configured) about the received email.   
7. The system logs the successful receipt of the email.   
  
Alternative Flow:   
1. If the sender's email address is invalid or not properly formatted, the system logs the error and discards the email.   
2. If the system fails to connect to the Email Server during the reception process, it logs the error and waits for the next connection attempt.   
3. If the system cannot categorize the email due to missing or ambiguous data, it marks the email as "uncategorized" and prompts the Administrator to review it manually.   
4. If the system fails to store the email in the Email Archive database, it logs the error and retries the storage operation.  
  
Use Case Name: Format Email Content   
Use Case ID: UC-07   
Actors: Administrator, Email Archive System, Email Server   
  
Preconditions:   
1. The Administrator is authenticated and has access to the system.   
2. The system is connected to the Email Archive database for content retrieval.   
3. The system has access to the Email Server for sending emails.   
  
Postconditions:   
1. The email content is formatted according to the system's standards.   
2. The formatted email is stored in the Email Archive database.   
3. The Administrator receives a confirmation that the formatting is complete and the email is ready to be sent.   
  
Main Flow:   
1. The Administrator navigates to the "Format Email Content" option in the system.   
2. The system prompts the Administrator to select an email from the Email Archive database or input new content.   
3. The Administrator selects an email or enters the content to be formatted.   
4. The system retrieves the email content and applies predefined formatting rules (e.g., template, font, signature).   
5. The system displays the formatted email to the Administrator for review.   
6. The Administrator confirms the formatting is correct.   
7. The system saves the formatted email in the Email Archive database.   
8. The system provides an option to send the email or return to the main menu.   
  
Alternative Flow:   
1. If the selected email cannot be retrieved from the Email Archive database, the system displays an error and prompts the Administrator to try again.   
2. If the Administrator enters invalid or incomplete content, the system displays a validation error and allows them to correct the input.   
3. If the system fails to apply the formatting rules, it logs the error and displays a message to the Administrator, allowing them to choose a different template or rule set.   
4. If the Administrator chooses not to proceed with the formatting, the system cancels the operation and returns to the previous menu.  
  
Use Case Name: View Email   
Use Case ID: UC-08   
Actors: Administrator, Email Archive System   
Preconditions:   
1. The Administrator is authenticated and logged into the system.   
2. The system has access to the Email Archive database.   
3. The email to be viewed exists in the Email Archive database.   
  
Postconditions:   
1. The Administrator is shown the full content of the selected email.   
2. The system maintains the integrity and confidentiality of the email data.   
3. The viewing action is logged in the system for audit purposes.   
  
Main Flow:   
1. The Administrator navigates to the "View Email" option in the system.   
2. The system prompts the Administrator to select an email from the Email Archive database.   
3. The Administrator selects the desired email.   
4. The system retrieves the selected email from the Email Archive database.   
5. The system displays the email content, including sender, recipient, subject, timestamp, and message body.   
  
Alternative Flow:   
1. If the selected email cannot be retrieved, the system displays an error message and prompts the Administrator to select a valid email.   
2. If the email content is incomplete or corrupted, the system logs the issue and displays a message to the Administrator for review.   
3. If the Administrator attempts to view an email that is marked as deleted or restricted, the system denies access and displays an appropriate error message.  
  
Use Case Name: Delete Email   
Use Case ID: UC-09   
Actors: Administrator, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has access to the system.   
2. The system has access to the Email Archive database.   
3. The email to be deleted exists in the Email Archive database.   
  
Postconditions:   
1. The specified email is successfully deleted from the Email Archive database.   
2. A confirmation message is displayed to the Administrator.   
3. The deletion is logged in the system for audit purposes.   
  
Main Flow:   
1. The Administrator navigates to the "Delete Email" option in the system.   
2. The system prompts the Administrator to search for or select an email from the Email Archive database.   
3. The Administrator selects the email to be deleted.   
4. The system verifies the email's existence in the Email Archive database.   
5. The system confirms the deletion with the Administrator (e.g., through a pop-up dialog or checkbox).   
6. The system deletes the selected email from the Email Archive database.   
7. The system logs the deletion and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the selected email does not exist in the Email Archive database, the system displays an error message and prompts the Administrator to select a valid email.   
2. If the Administrator cancels the deletion confirmation, the system aborts the operation and returns to the previous menu.   
3. If the system fails to delete the email due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: Archive Email   
Use Case ID: UC-10   
Actors: Administrator, Email Archive System, Email Server   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to archive emails.   
2. The system has access to the Email Archive database.   
3. The system is connected to the Email Server for retrieving emails.   
4. The email to be archived exists in the system or is available for retrieval.   
  
Postconditions:   
1. The selected email is successfully archived in the Email Archive database.   
2. The email is marked as archived and no longer available for regular email operations.   
3. The system logs the archiving action for audit purposes.   
4. The Administrator receives a confirmation message indicating the email has been archived.   
  
Main Flow:   
1. The Administrator navigates to the "Archive Email" option in the system.   
2. The system prompts the Administrator to select an email from the list of emails or input a specific email identifier.   
3. The Administrator selects the email to be archived.   
4. The system verifies the email's existence and confirms it is not already archived.   
5. The system prompts the Administrator to confirm the archiving action.   
6. Upon confirmation, the system moves the email to the Email Archive database and updates its status to "archived".   
7. The system logs the archiving action and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the selected email is not found or has already been archived, the system displays an error and prompts the Administrator to select a valid email.   
2. If the Administrator cancels the archiving confirmation, the system aborts the operation and returns to the previous menu.   
3. If the system fails to archive the email due to database errors or constraints, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: Recover Archived Email   
Use Case ID: UC-11   
Actors: Administrator, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has access to the system.   
2. The system has access to the Email Archive database.   
3. The email to be recovered is archived and exists in the Email Archive database.   
  
Postconditions:   
1. The selected archived email is successfully recovered and moved back to the active email system.   
2. The email status is updated to "active" in the Email Archive database.   
3. The system logs the recovery action for audit purposes.   
4. The Administrator receives a confirmation message indicating the email has been recovered.   
  
Main Flow:   
1. The Administrator navigates to the "Recover Archived Email" option in the system.   
2. The system prompts the Administrator to search for or select an email from the Email Archive database.   
3. The Administrator selects the email to be recovered.   
4. The system verifies the email's existence in the Email Archive database and confirms it is marked as "archived".   
5. The system prompts the Administrator to confirm the recovery action.   
6. Upon confirmation, the system moves the email from the Email Archive database back to the active email system and updates its status.   
7. The system logs the recovery action and displays a success message to the Administrator.   
  
Alternative Flow:   
1. If the selected email is not found in the Email Archive database, the system displays an error and prompts the Administrator to select a valid email.   
2. If the selected email is already marked as "active", the system displays a message indicating that it is not archived and cannot be recovered.   
3. If the Administrator cancels the recovery confirmation, the system aborts the operation and returns to the previous menu.   
4. If the system fails to recover the email due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: Create Contact   
Use Case ID: UC-12   
Actors: Administrator, Contact System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to create contacts.   
2. The system has access to the Contact database.   
3. The system is connected to the Email Archive database for email verification.   
  
Postconditions:   
1. A new Contact is successfully created in the system.   
2. The associated email address is verified and linked to the Contact.   
3. The system logs the creation of the Contact for audit purposes.   
  
Main Flow:   
1. The Administrator navigates to the "Create Contact" option in the system.   
2. The system prompts the Administrator to enter the Contact's details (name, email, phone number, address).   
3. The Administrator inputs the required information.   
4. The system checks the Email Archive database to verify the email address is valid and not already associated with another Contact.   
5. If the email is valid and unique, the system creates the Contact and stores the information in the Contact database.   
6. The system logs the creation and displays a success message to the Administrator.   
  
Alternative Flow:   
1. If the email address already exists in the Email Archive database and is associated with another Contact, the system displays an error message and prompts the Administrator to provide a different email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to create the Contact due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: Delete Contact   
Use Case ID: UC-13   
Actors: Administrator, Contact System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to delete contacts.   
2. The system has access to the Contact database.   
3. The system is connected to the Email Archive database for verification purposes.   
  
Postconditions:   
1. The specified Contact is deleted from the system.   
2. The associated email address is updated in the Email Archive database to reflect the deletion.   
3. A confirmation message is displayed to the Administrator.   
  
Main Flow:   
1. The Administrator navigates to the "Delete Contact" option in the system.   
2. The system prompts the Administrator to enter the email address of the Contact to be deleted.   
3. The Administrator inputs the email address.   
4. The system verifies the email address exists in the Contact database.   
5. The system confirms the deletion with the Administrator (e.g., through a pop-up dialog or checkbox).   
6. The system deletes the Contact and updates the email status in the Email Archive database.   
7. The system logs the deletion and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the email address does not exist in the Contact database, the system displays an error message and prompts the Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to delete the Contact due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: View Contact   
Use Case ID: UC-14   
Actors: Administrator, Contact System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has access to the system.   
2. The system has access to the Contact database.   
3. The system is connected to the Email Archive database for cross-referencing email information.   
4. The Contact to be viewed exists in the Contact database.   
  
Postconditions:   
1. The Administrator is shown the full details of the selected Contact.   
2. The associated email information is retrieved and displayed from the Email Archive database.   
3. The system maintains the integrity and confidentiality of the data displayed.   
  
Main Flow:   
1. The Administrator navigates to the "View Contact" option in the system.   
2. The system prompts the Administrator to enter or select the email address of the Contact to be viewed.   
3. The Administrator inputs the email address.   
4. The system verifies that the email exists in the Contact database.   
5. The system retrieves the Contact details (name, phone number, address) and cross-references the Email Archive database for additional email-related data.   
6. The system displays the Contact information and associated email history to the Administrator.   
  
Alternative Flow:   
1. If the email address does not exist in the Contact database, the system displays an error message and prompts the Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to retrieve email-related information from the Email Archive database, it logs the error and displays a message to the Administrator indicating the issue.  
  
Use Case Name: Update Contact Information   
Use Case ID: UC-15   
Actors: Administrator, Contact System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to update contact information.   
2. The system has access to the Contact database.   
3. The system is connected to the Email Archive database for verification and email-related data.   
4. The Contact to be updated exists in the Contact database.   
  
Postconditions:   
1. The Contact information is successfully updated in the system.   
2. The associated email address is verified and updated in the Email Archive database.   
3. A confirmation message is displayed to the Administrator.   
4. The system logs the update for audit purposes.   
  
Main Flow:   
1. The Administrator navigates to the "Update Contact Information" option in the system.   
2. The system prompts the Administrator to enter the email address of the Contact whose information is to be updated.   
3. The Administrator inputs the email address.   
4. The system verifies the email exists in the Contact database and retrieves the current contact details.   
5. The system displays the current Contact information and allows the Administrator to edit it.   
6. The Administrator modifies the required information (e.g., name, phone number, address).   
7. The system updates the Contact database with the new information.   
8. The system verifies and updates the associated email in the Email Archive database.   
9. The system logs the update and displays a success message to the Administrator.   
  
Alternative Flow:   
1. If the email address does not exist in the Contact database, the system displays an error message and prompts the Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to update the Contact due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: Manage Email Archive   
Use Case ID: UC-16   
Actors: Administrator, Email Archive System, Email Server   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage the email archive.   
2. The system has access to the Email Archive database.   
3. The system is connected to the Email Server to retrieve or send emails if required.   
  
Postconditions:   
1. The email archive is updated based on the Administrator's actions (e.g., emails added, removed, or modified).   
2. The system logs all archive management actions for audit purposes.   
3. The status of the emails in the Email Archive database is updated accordingly.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Email Archive" option in the system.   
2. The system displays a list of archived emails with options to view, recover, or delete them.   
3. The Administrator selects an action (e.g., recover an email or delete it permanently).   
4. The system verifies the selected email's status and availability.   
5. The system performs the requested action (e.g., recover or delete) and updates the Email Archive database.   
6. The system logs the action and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the selected email does not exist in the Email Archive database, the system displays an error message and prompts the Administrator to select a valid email.   
2. If the system fails to perform the requested action (e.g., due to database constraints), it logs the error and displays a message to the Administrator, allowing them to retry or investigate the issue.   
3. If the system cannot connect to the Email Server during the process, it logs the error and displays a message to the Administrator for resolution.  
  
Use Case Name: View Email Archive   
Use Case ID: UC-17   
Actors: Administrator, Email Archive System   
Preconditions:   
1. The Administrator is authenticated and has access to the system.   
2. The system has access to the Email Archive database.   
3. The Email Archive database contains emails that can be viewed.   
  
Postconditions:   
1. The Administrator is shown a list of archived emails and can view the content of selected emails.   
2. The system maintains the integrity and confidentiality of the email data.   
3. The viewing action is logged in the system for audit purposes.   
  
Main Flow:   
1. The Administrator navigates to the "View Email Archive" option in the system.   
2. The system retrieves and displays a list of archived emails from the Email Archive database.   
3. The Administrator selects an email from the list.   
4. The system verifies the selected email's existence and status in the Email Archive database.   
5. The system retrieves the full content of the selected email, including sender, recipient, subject, timestamp, and message body.   
6. The system displays the email content to the Administrator.   
7. The system logs the viewing action and displays a confirmation message.   
  
Alternative Flow:   
1. If the selected email does not exist in the Email Archive database, the system displays an error message and prompts the Administrator to select a valid email.   
2. If the system fails to retrieve the email content due to database issues, it logs the error and displays a message to the Administrator, allowing them to retry or investigate the issue.   
3. If the Administrator attempts to view an email marked as deleted or restricted, the system denies access and displays an appropriate error message.  
  
Use Case Name: Manage Administrator Permissions   
Use Case ID: UC-18   
Actors: Administrator, User Account System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage administrator roles.   
2. The system has access to the User Account database to identify and modify administrator accounts.   
3. The system is connected to the Email Archive database for verification and communication purposes.   
  
Postconditions:   
1. The Administrator's permissions are successfully assigned, modified, or revoked.   
2. The changes are recorded in the User Account database.   
3. A confirmation email is sent to the affected Administrator (if applicable) and/or the system displays a success message.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Administrator Permissions" option in the system.   
2. The system prompts the Administrator to select an existing user or input an email to assign or modify administrator permissions.   
3. The Administrator selects or inputs the email address.   
4. The system verifies that the email exists in the User Account database and checks if the user is already an administrator.   
5. The system displays the current permission level of the selected user and provides options to grant, revoke, or modify administrator status.   
6. The Administrator selects the desired action and confirms the change.   
7. The system updates the User Account database with the new permission status.   
8. The system sends a confirmation email to the affected user (if the email server is available) and/or displays a success message to the Administrator.   
  
Alternative Flow:   
1. If the email address does not exist in the User Account database, the system displays an error message and prompts the Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it.   
3. If the system fails to send the confirmation email, it logs the error and displays a message to the Administrator indicating the failure, allowing them to retry or manually send the confirmation.   
4. If the Administrator cancels the permission change confirmation, the system aborts the operation and returns to the previous menu.   
5. If the system fails to update the User Account database due to constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: Manage Email Draft   
Use Case ID: UC-19   
Actors: Administrator, Email Draft System, Email Archive System, Email Server   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage email drafts.   
2. The system has access to the Email Draft database.   
3. The system is connected to the Email Archive database for verification and storage purposes.   
4. The system is connected to the Email Server for sending emails if required.   
  
Postconditions:   
1. The email draft is either saved, updated, deleted, or sent based on the Administrator's action.   
2. The status of the draft (e.g., saved, sent, deleted) is recorded in the Email Draft database.   
3. If the draft is sent, the email is stored in the Email Archive database with the appropriate status.   
4. The system logs all draft management actions for audit purposes.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Email Draft" option in the system.   
2. The system displays a list of available email drafts with options to view, edit, delete, or send them.   
3. The Administrator selects a draft for an action (e.g., edit, delete, send).   
4. The system retrieves the selected draft from the Email Draft database.   
5. If the action is to edit or view, the system displays the draft's content (sender, recipient, subject, message body).   
6. If the action is to edit, the Administrator modifies the draft's content as needed.   
7. The system saves the updated draft or, if the action is to send, prepares the email for delivery.   
8. If the action is to send, the system verifies the recipient's email address and sends the email via the connected Email Server.   
9. The system updates the status of the draft in the Email Draft database and, if sent, logs the email in the Email Archive database.   
10. The system logs the action and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the selected draft does not exist in the Email Draft database, the system displays an error message and prompts the Administrator to select a valid draft.   
2. If the email address of the recipient is invalid or not properly formatted, the system displays a validation error and allows the Administrator to correct it before sending.   
3. If the system fails to send the email (e.g., due to server issues), it logs the error and displays a message to the Administrator, allowing them to retry or investigate the issue.   
4. If the Administrator cancels the action (e.g., sending or deleting a draft), the system aborts the operation and returns to the previous menu.   
5. If the system fails to update or delete the draft due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: Manage Contact Group   
Use Case ID: UC-20   
Actors: Administrator, Contact System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage contact groups.   
2. The system has access to the Contact Group database.   
3. The system is connected to the Contact database and the Email Archive database for data retrieval and verification.   
  
Postconditions:   
1. The Contact Group is either created, updated, deleted, or modified based on the Administrator's action.   
2. The changes to the Contact Group are recorded in the system for audit purposes.   
3. The associated contact and email information is updated or verified in the relevant databases.   
4. A confirmation message is displayed to the Administrator for the action performed.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Contact Group" option in the system.   
2. The system displays a list of existing Contact Groups along with options to create, update, delete, or view them.   
3. The Administrator selects an action (e.g., create a new group, update an existing one, delete a group).   
4. If the action is to create a new group, the system prompts the Administrator to enter the group name and select contacts to add to the group.   
5. The Administrator inputs the group name and selects the relevant contacts from the Contact database.   
6. The system verifies the selected contacts exist in the Contact database and are not already in another group (if such a constraint is in place).   
7. The system creates or updates the Contact Group and saves the changes to the Contact Group database.   
8. If the action is to delete a group, the system confirms the deletion with the Administrator and removes the group and its associations.   
9. The system logs the action and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the group name is invalid, duplicate, or not provided, the system displays a validation error and prompts the Administrator to correct it.   
2. If the selected contacts do not exist in the Contact database, the system displays an error message and prompts the Administrator to select valid contacts.   
3. If the system fails to update or delete the Contact Group due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.   
4. If the Administrator cancels the action (e.g., group deletion or creation), the system aborts the operation and returns to the previous menu.  
  
Use Case Name: Manage Email Folder   
Use Case ID: UC-21   
Actors: Administrator, Email Folder System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage email folders.   
2. The system has access to the Email Folder database.   
3. The system is connected to the Email Archive database for associating emails with folders.   
4. The Email Folder System is operational and supports folder creation, deletion, and modification.   
  
Postconditions:   
1. The Email Folder is either created, updated, deleted, or modified based on the Administrator's action.   
2. The changes to the Email Folder are recorded in the system for audit purposes.   
3. The associated emails are updated to reflect folder changes in the Email Archive database.   
4. A confirmation message is displayed to the Administrator for the action performed.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Email Folder" option in the system.   
2. The system displays a list of existing email folders along with options to create, update, delete, or view them.   
3. The Administrator selects an action (e.g., create a new folder, update an existing folder, delete a folder).   
4. If the action is to create a new folder, the system prompts the Administrator to enter the folder name and select associated emails from the Email Archive database.   
5. The Administrator inputs the folder name and selects the relevant emails to associate with the folder.   
6. The system verifies the folder name is valid and not already in use.   
7. The system creates the new Email Folder and links the selected emails to it in the Email Archive database.   
8. If the action is to update an existing folder, the system allows the Administrator to rename the folder or reassign emails to it.   
9. The system updates the Email Folder and modifies the associated email records in the Email Archive database.   
10. If the action is to delete a folder, the system confirms the deletion with the Administrator and removes the folder and its associations.   
11. The system logs the action and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the folder name is invalid, duplicate, or not provided, the system displays a validation error and prompts the Administrator to correct it.   
2. If the selected emails do not exist in the Email Archive database, the system displays an error message and prompts the Administrator to select valid emails.   
3. If the system fails to update or delete the Email Folder due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.   
4. If the Administrator cancels the action (e.g., folder deletion or creation), the system aborts the operation and returns to the previous menu.  
  
Use Case Name: Manage Email Archive Record   
Use Case ID: UC-22   
Actors: Administrator, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage email archive records.   
2. The system has access to the Email Archive Record database.   
3. The Email Archive database is accessible for retrieving and updating email records.   
4. The system is configured to perform operations such as editing, deleting, or viewing records.   
  
Postconditions:   
1. The Email Archive Record is either created, updated, deleted, or viewed based on the Administrator's action.   
2. The changes to the Email Archive Record are recorded in the system for audit purposes.   
3. The associated email data in the Email Archive database is updated accordingly.   
4. A confirmation message is displayed to the Administrator for the action performed.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Email Archive Record" option in the system.   
2. The system displays a list of existing Email Archive Records along with options to create, update, delete, or view them.   
3. The Administrator selects an action (e.g., create a new record, update an existing one, delete a record).   
4. If the action is to create a new record, the system prompts the Administrator to enter the record details, such as email identifier, archive date, and status.   
5. The Administrator inputs the required information.   
6. The system verifies the email identifier exists in the Email Archive database and checks for any existing records.   
7. The system creates the new Email Archive Record and saves the information in the Email Archive Record database.   
8. If the action is to update an existing record, the system retrieves the record, displays it to the Administrator, and allows modifications to the record fields.   
9. The system updates the Email Archive Record with the new information and logs the change.   
10. If the action is to delete a record, the system confirms the deletion with the Administrator and removes the record from the Email Archive Record database.   
11. The system logs the action and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the email identifier does not exist in the Email Archive database, the system displays an error message and prompts the Administrator to provide a valid identifier.   
2. If the record details provided are invalid or incomplete, the system displays a validation error and allows the Administrator to correct the input.   
3. If the system fails to create, update, or delete the Email Archive Record due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.   
4. If the Administrator cancels the action (e.g., record deletion or creation), the system aborts the operation and returns to the previous menu.  
  
Use Case Name: Manage Permission Setting   
Use Case ID: UC-23   
Actors: Administrator, Permission Setting System, User Account System, Email Archive System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage permission settings.   
2. The system has access to the Permission Setting database.   
3. The system is connected to the User Account database to identify users for whom permission settings will be managed.   
4. The system is connected to the Email Archive database for verification and communication purposes if email notifications are enabled.   
  
Postconditions:   
1. The Permission Setting is either created, updated, or deleted based on the Administrator's action.   
2. The changes to the Permission Setting are recorded in the system for audit purposes.   
3. The associated user account or system roles are updated accordingly to reflect the new permission settings.   
4. A confirmation message is displayed to the Administrator for the action performed.   
5. If email notifications are enabled, a confirmation email is sent to the affected user(s) or Administrator.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Permission Setting" option in the system.   
2. The system displays a list of existing permission settings along with options to create, update, or delete them.   
3. The Administrator selects an action (e.g., create a new permission setting, update an existing one, delete a permission setting).   
4. If the action is to create a new permission setting, the system prompts the Administrator to enter the setting name, description, and assign it to specific user roles or individual users.   
5. The Administrator inputs the required information and confirms the assignment.   
6. The system validates the user roles or individual users exist in the User Account database.   
7. The system creates or updates the Permission Setting in the Permission Setting database and applies the changes to the relevant user accounts.   
8. If the action is to delete a permission setting, the system confirms the deletion with the Administrator and removes the setting, revoking its permissions from the affected users or roles.   
9. The system logs the action and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the setting name is invalid, duplicate, or not provided, the system displays a validation error and prompts the Administrator to correct it.   
2. If the selected user roles or individual users do not exist in the User Account database, the system displays an error message and prompts the Administrator to select valid users or roles.   
3. If the system fails to update or delete the Permission Setting due to database constraints or errors, it logs the failure and displays an error message to the Administrator, allowing them to retry or investigate the issue.   
4. If the system fails to send a confirmation email due to server issues, it logs the failure and displays a message to the Administrator, allowing them to retry or manually send the confirmation.   
5. If the Administrator cancels the action (e.g., setting creation, update, or deletion), the system aborts the operation and returns to the previous menu.  
  
Use Case Name: Create Administrator   
Use Case ID: UC-24   
Actors: Super Administrator, User Account System, Email Archive System   
  
Preconditions:   
1. The Super Administrator is authenticated and has the necessary permissions to create administrator accounts.   
2. The system has access to the User Account database.   
3. The system is connected to the Email Archive database for email validation and verification.   
4. The system is connected to the Email Server for sending confirmation emails.   
  
Postconditions:   
1. A new Administrator account is successfully created in the User Account database.   
2. The email address is verified and stored in the Email Archive database.   
3. A confirmation email is sent to the provided email address.   
4. The new Administrator is assigned the appropriate permissions and roles.   
5. The system logs the creation of the Administrator account for audit purposes.   
  
Main Flow:   
1. The Super Administrator navigates to the "Create Administrator" option in the system.   
2. The system prompts the Super Administrator to enter the new Administrator's contact information and email address.   
3. The Super Administrator inputs the required details (name, contact, email).   
4. The system verifies the email address by checking the Email Archive database for existing entries.   
5. The system generates a new User Account with the role of "Administrator" and sets a default password or prompts the Super Administrator to set one.   
6. The system sends a confirmation email to the new Administrator's provided email address.   
7. The system logs the creation of the Administrator account and displays a success message to the Super Administrator.   
  
Alternative Flow:   
1. If the email address already exists in the Email Archive database and is associated with another Administrator or User account, the system displays an error message and prompts the Super Administrator to provide a different email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Super Administrator to correct it.   
3. If the system fails to send the confirmation email, it logs the error and displays a message to the Super Administrator indicating the failure, allowing them to retry or manually send the confirmation.   
4. If the system fails to create the Administrator account due to database constraints or errors, it logs the failure and displays an error message to the Super Administrator, allowing them to retry or investigate the issue.  
  
Use Case Name: Delete Administrator   
Use Case ID: UC-25   
Actors: Super Administrator, User Account System, Email Archive System   
  
Preconditions:   
1. The Super Administrator is authenticated and has the necessary permissions to delete administrator accounts.   
2. The system has access to the User Account database to identify and remove the administrator account.   
3. The system is connected to the Email Archive database to verify and update associated email information.   
4. The administrator account to be deleted exists in the system and is not the only Super Administrator.   
  
Postconditions:   
1. The specified Administrator account is successfully deleted from the User Account database.   
2. The associated email address is marked as deleted or unassigned in the Email Archive database.   
3. A confirmation message is displayed to the Super Administrator.   
4. The system logs the deletion action for audit purposes.   
  
Main Flow:   
1. The Super Administrator navigates to the "Delete Administrator" option in the system.   
2. The system prompts the Super Administrator to enter the email address of the Administrator to be deleted.   
3. The Super Administrator inputs the email address.   
4. The system verifies that the email exists in the User Account database and is associated with an Administrator account.   
5. The system confirms that there is at least one other Super Administrator to ensure system integrity.   
6. The system confirms the deletion with the Super Administrator (e.g., through a pop-up dialog or checkbox).   
7. Upon confirmation, the system deletes the Administrator account from the User Account database.   
8. The system updates the associated email address in the Email Archive database to reflect the deletion.   
9. The system logs the deletion action and displays a success message to the Super Administrator.   
  
Alternative Flow:   
1. If the email address does not exist in the User Account database, the system displays an error message and prompts the Super Administrator to provide a valid email.   
2. If the email address is invalid or not properly formatted, the system displays a validation error and allows the Super Administrator to correct it.   
3. If the system detects that the Administrator to be deleted is the last Super Administrator, it displays an error message and prevents the deletion to maintain system access.   
4. If the system fails to update the Email Archive database, it logs the error and displays a message to the Super Administrator, allowing them to retry or investigate the issue.   
5. If the system fails to delete the Administrator account due to database constraints or errors, it logs the failure and displays an error message to the Super Administrator, allowing them to retry or investigate the issue.   
6. If the Super Administrator cancels the deletion confirmation, the system aborts the operation and returns to the previous menu.