# Functional Requirement

1.1 Email Creation Function   
 Function ID: FR-01   
 Description: A user or administrator can create a new email by providing the recipient's email address, subject, and message body. The system validates the input and prepares the email for sending.   
 Input: Recipient’s email address, subject, message body   
 Output: A new Email object with valid data and a unique EmailID   
  
1.2 Email Sending Function   
 Function ID: FR-02   
 Description: A user or administrator can send a created email. The system updates the email status to "Sent", removes it from the draft (if applicable), and adds it to the recipient's inbox.   
 Input: Email object (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID)   
 Output: The email is added to the recipient's inbox and marked as sent in the system   
  
1.3 Email Receiving Function   
 Function ID: FR-03   
 Description: When an email is sent, the system delivers it to the recipient's inbox and updates the inbox status. The recipient is notified of the new email.   
 Input: Email object (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID)   
 Output: The email is added to the recipient's inbox and a notification is sent to the recipient   
  
1.4 Email Viewing Function   
 Function ID: FR-04   
 Description: A user or administrator can view the content of an email by selecting it from their inbox. The system retrieves the email and marks it as read.   
 Input: Selected EmailID and UserID/InboxID   
 Output: Email content (sender, subject, body, attachments) and updated read status   
  
1.5 Email Deletion Function   
 Function ID: FR-05   
 Description: A user or administrator can delete an email from their inbox. The system removes the email from the inbox and marks it as deleted in the database.   
 Input: Selected EmailID and UserID/InboxID   
 Output: Updated inbox without the deleted email and the email marked as deleted in the database   
  
1.6 User Creation Function   
 Function ID: FR-06   
 Description: An administrator can create a new user account by providing the username, password, and role. The system validates the inputs and stores the new user in the database.   
 Input: Username, password, role   
 Output: A new User object with a unique UserID and initial settings   
  
1.7 User Management Function   
 Function ID: FR-07   
 Description: An administrator can edit or delete an existing user account. The system updates the user information or removes the user from the database based on the action.   
 Input: Selected UserID and new user information (e.g., role, status)   
 Output: Updated or deleted User object and confirmation message   
  
1.8 Administrator Creation Function   
 Function ID: FR-08   
 Description: An administrator can create a new administrator account by providing the username, password, and role. The system validates and stores the new administrator in the database.   
 Input: Username, password, role   
 Output: A new Admin object with a unique AdminID and initial permissions   
  
1.9 Inbox Management Function   
 Function ID: FR-09   
 Description: A user or administrator can manage their inbox by performing actions such as marking an email as read, moving it to another folder, or deleting it.   
 Input: Selected EmailID, InboxID, and action type (e.g., mark as read, delete, move)   
 Output: Updated inbox status and confirmation message   
  
1.10 Email Status Management Function   
 Function ID: FR-10   
 Description: A user or administrator can update the status of an email (e.g., read, unread, flagged, archived) to reflect its current state in the system.   
 Input: Selected EmailID and new StatusID (from EmailStatus)   
 Output: Updated Email object with new status and refreshed inbox or sent items view

# External Description

# 2. External Interfaces  
  
## 2.1 User Interface Output  
  
The system interacts with users through a graphical user interface (GUI) that allows them to perform various actions related to email and user management. The following interactions are supported:  
  
- \*\*Email Creation Interface\*\*:   
 - \*Description\*: A form-based interface where users or administrators can input the recipient's email address, subject, and message body to create a new email.   
 - \*Interaction Method\*: The user fills out the form and submits it. The system validates the input fields and returns a confirmation message if the email is successfully created.   
 - \*Inputs\*: Recipient’s email address, subject, message body.   
 - \*Outputs\*: A new Email object with valid data and a unique EmailID, displayed to the user.  
  
- \*\*Email Sending Interface\*\*:   
 - \*Description\*: An interface that allows users or administrators to send a created email.   
 - \*Interaction Method\*: The user selects the email from their draft or outbox and clicks "Send". The system updates the email status to "Sent", removes it from the draft (if applicable), and confirms the action.   
 - \*Inputs\*: Email object (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID).   
 - \*Outputs\*: Confirmation message that the email has been sent and the updated view of the outbox or sent items.  
  
- \*\*Email Receiving Interface\*\*:   
 - \*Description\*: The system provides an inbox interface where users or administrators can view incoming emails.   
 - \*Interaction Method\*: When an email is sent to the user, the system automatically adds it to the inbox and displays a notification.   
 - \*Inputs\*: Email object (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID).   
 - \*Outputs\*: The email is added to the inbox, and the user receives a notification.  
  
- \*\*Email Viewing Interface\*\*:   
 - \*Description\*: An interface where users or administrators can view the details of a selected email.   
 - \*Interaction Method\*: The user selects an email from their inbox and clicks "View". The system retrieves the email and marks it as read.   
 - \*Inputs\*: Selected EmailID and UserID/InboxID.   
 - \*Outputs\*: Email content (sender, subject, body, attachments) is displayed, and the read status is updated in the system.  
  
- \*\*Email Deletion Interface\*\*:   
 - \*Description\*: A user or administrator can delete an email from their inbox.   
 - \*Interaction Method\*: The user selects an email and clicks "Delete". The system removes the email from the inbox and updates the database.   
 - \*Inputs\*: Selected EmailID and UserID/InboxID.   
 - \*Outputs\*: The inbox is updated without the deleted email, and a confirmation message is displayed.  
  
- \*\*User Creation Interface\*\*:   
 - \*Description\*: An interface that allows administrators to create new user accounts.   
 - \*Interaction Method\*: The administrator inputs the username, password, and role and submits the form. The system validates the input and creates a new user.   
 - \*Inputs\*: Username, password, role.   
 - \*Outputs\*: A new User object with a unique UserID and initial settings is created and displayed.  
  
- \*\*User Management Interface\*\*:   
 - \*Description\*: An interface where administrators can edit or delete existing user accounts.   
 - \*Interaction Method\*: The administrator selects a user and modifies the user’s role or status. Alternatively, the administrator can delete the user. The system updates the database accordingly.   
 - \*Inputs\*: Selected UserID and new user information (e.g., role, status).   
 - \*Outputs\*: Updated or deleted User object and a confirmation message.  
  
- \*\*Inbox Management Interface\*\*:   
 - \*Description\*: A user or administrator can manage their inbox by performing actions such as marking an email as read, moving it to another folder, or deleting it.   
 - \*Interaction Method\*: The user selects an email and chooses an action (e.g., "Mark as Read", "Move to Folder", "Delete"). The system updates the inbox accordingly.   
 - \*Inputs\*: Selected EmailID, InboxID, and action type (e.g., mark as read, delete, move).   
 - \*Outputs\*: Updated inbox status and a confirmation message.  
  
- \*\*Email Status Management Interface\*\*:   
 - \*Description\*: A user or administrator can update the status of an email (e.g., read, unread, flagged, archived).   
 - \*Interaction Method\*: The user selects an email and changes its status through a dropdown or buttons. The system updates the Email object and the inbox view.   
 - \*Inputs\*: Selected EmailID and new StatusID (from EmailStatus).   
 - \*Outputs\*: Updated Email object with new status and a refreshed inbox or sent items view.  
  
## 2.2 Hardware Interface Output  
  
The system does not have direct hardware dependencies. It operates on standard computing hardware and is designed to function on a variety of devices, including desktop computers, laptops, and mobile devices. The system is compatible with the following hardware requirements:  
  
- \*Supported Device Types\*:   
 - Desktop computers running Windows, macOS, or Linux.   
 - Laptops with similar operating systems.   
 - Mobile devices with Android or iOS operating systems.   
  
- \*Communication Protocols\*:   
 - HTTP/HTTPS for secure communication between the client and server.   
 - WebSocket for real-time notifications and inbox updates.   
  
- \*Interaction Method\*:   
 - The system relies on standard input/output devices such as keyboards, mice, and touchscreens for user interaction.   
 - It also supports display output on monitors or mobile screens for viewing emails and managing the inbox.  
  
## 2.3 Software Interface Output  
  
The system interacts with various software components, including databases, third-party libraries, and external APIs. These interfaces are essential for storing and retrieving data, validating inputs, and supporting the core functionality of the system.  
  
- \*\*Database Interface\*\*:   
 - \*Description\*: The system uses a relational database to store and manage email data, user data, and inbox status.   
 - \*Interaction Method\*: The system communicates with the database using SQL queries or an ORM (Object-Relational Mapping) tool to perform CRUD (Create, Read, Update, Delete) operations.   
 - \*Data Sources\*:   
 - Email table to store Email objects (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID).   
 - User table to store User objects (UserID, Username, Password, Role).   
 - Admin table to store Admin objects (AdminID, Username, Password, Role).   
 - Inbox table to track emails associated with a user or administrator (InboxID, UserID/InboxID, EmailID).   
 - EmailStatus table to define valid email statuses (StatusID, StatusName).  
  
- \*\*Email Validation Library\*\*:   
 - \*Description\*: A third-party or in-house library used to validate email addresses, subjects, and message bodies.   
 - \*Interaction Method\*: The system calls the library's validation functions when an email is created or sent.   
 - \*Inputs\*: Recipient’s email address, subject, message body.   
 - \*Outputs\*: Validation result (valid/invalid) and error messages if the input is invalid.  
  
- \*\*Email Sending Service (Optional API)\*\*:   
 - \*Description\*: An external API or service used to send emails over the internet. This is optional and may be used in addition to the system's internal sending function.   
 - \*Interaction Method\*: The system sends an HTTP POST request to the API endpoint with the email data. The API then handles the delivery to the recipient.   
 - \*Inputs\*: Email object (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID).   
 - \*Outputs\*: API response indicating the success or failure of the email delivery.  
  
- \*\*Email Notification Service\*\*:   
 - \*Description\*: A service that sends notifications to users when a new email arrives in their inbox.   
 - \*Interaction Method\*: The system triggers a notification through the service when an email is successfully sent and received.   
 - \*Inputs\*: Email object (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID).   
 - \*Outputs\*: A notification is sent to the recipient's device or displayed within the application.  
  
## 2.4 Communication Interface Output  
  
The system utilizes network communication to facilitate email delivery, notifications, and remote access to the email management functionalities.  
  
- \*\*Email Delivery Communication\*\*:   
 - \*Description\*: The system communicates with email servers or external email services to send and receive emails.   
 - \*Interaction Method\*: When an email is sent, the system uses SMTP (Simple Mail Transfer Protocol) or a third-party email delivery API to transmit the email.   
 - \*Inputs\*: Email object (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID).   
 - \*Outputs\*: The email is delivered to the recipient's email server or inbox.  
  
- \*\*Email Notification Communication\*\*:   
 - \*Description\*: The system communicates with the user's device or web browser to notify them of new emails.   
 - \*Interaction Method\*: The system uses HTTP push notifications or WebSocket connections to deliver real-time alerts.   
 - \*Inputs\*: Email object (EmailID, SenderID, ReceiverID, Subject, Body, Timestamp, StatusID).   
 - \*Outputs\*: A notification is displayed to the user, indicating that a new email has been received.  
  
- \*\*User Interface Communication\*\*:   
 - \*Description\*: The system communicates with the web browser or mobile application to display the user interface and handle user inputs.   
 - \*Interaction Method\*: The system uses HTTP requests and responses to provide dynamic content to the client application.   
 - \*Inputs\*: User inputs such as EmailID, UserID, action types, etc.   
 - \*Outputs\*: Updated views of the inbox, sent items, and user management screens.  
  
- \*\*User Management Communication (Optional API)\*\*:   
 - \*Description\*: If the system is integrated with a centralized user management service, it communicates with that service to create, update, or delete user accounts.   
 - \*Interaction Method\*: The system sends HTTP requests to the user management API with the relevant user data.   
 - \*Inputs\*: Username, password, role, and action (create, update, delete).   
 - \*Outputs\*: API response confirming the success or failure of the user management operation.  
  
All external interfaces are designed to be consistent with the terminology and structure defined in the functional requirements, ensuring clarity and reducing ambiguity for developers implementing the system.