# Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Email Creation Function   
Function ID: FR-01   
Description: Users can compose and create new emails with specified recipient, subject, and body content. The system supports additional options such as file attachments, priority settings, and scheduling.   
Input: Recipient email address, subject, body content, optional attachments, priority level, and scheduling preferences.   
Output: A new email object with attributes (EmailID, Subject, Body, SenderID, ReceiverID, SendDate, Status) saved in the system.  
  
## 1.2 Email Sending Function   
Function ID: FR-02   
Description: After creating or editing an email, users can send it to the intended recipient(s). The system validates the email and ensures it is sent successfully.   
Input: A composed email object with valid recipient, subject, and body.   
Output: The email is sent to the recipient, and a record is created in the EmailRecord table (RecordID, EmailID, ActionType, ActionDate, Status).  
  
## 1.3 Email Receiving Function   
Function ID: FR-03   
Description: The system receives incoming emails from the email server or other users, verifies the sender, and stores the email in the appropriate user's inbox.   
Input: An incoming email from the email server or another user with attributes (Subject, Body, SenderID, ReceiverID).   
Output: The email is stored in the user's inbox with attributes (InboxID, UserID, EmailID, ReadStatus, DateAdded) and a record is created in the EmailRecord table.  
  
## 1.4 Email Deletion Function   
Function ID: FR-04   
Description: Users can delete emails from their inbox or other folders. The system prompts for confirmation and optionally moves the email to a trash folder before permanent deletion.   
Input: Selected email(s) to be deleted from a folder or inbox.   
Output: The selected email(s) are removed from the folder or inbox, and a record is created in the InboxRecord table (RecordID, InboxID, ActionType, ActionDate, Status).  
  
## 1.5 Email Viewing Function   
Function ID: FR-05   
Description: Users can view the content of an email, including subject, body, and attachments. The system updates the email's read status and logs the view action.   
Input: A selected email from the inbox or another folder.   
Output: The email content is displayed to the user, the email is marked as "read", and a record is created in the InboxRecord table.  
  
## 1.6 User Creation Function   
Function ID: FR-06   
Description: Administrators can create new user accounts in the system. The system validates the input data and stores the user information securely.   
Input: User details including name, email address, password, and role.   
Output: A new user object with attributes (UserID, Name, EmailAddress, Password, Role) stored in the database.  
  
## 1.7 User Registration Function   
Function ID: FR-07   
Description: Administrators can register new users in the system. The system ensures the email is unique and the password meets security requirements.   
Input: User details including full name, email address, password, and role.   
Output: A registered user object with attributes (UserID, Name, EmailAddress, Password, Role) stored in the database.  
  
## 1.8 User Management Function   
Function ID: FR-08   
Description: Administrators can manage existing user accounts by editing, disabling, or deleting them. The system updates the user's status and logs the action.   
Input: Target user ID and the action to be performed (edit, disable, delete).   
Output: Updated user object or user deletion confirmation, with a record created in the EmailRecord or InboxRecord table as needed.  
  
## 1.9 Administrator Creation Function   
Function ID: FR-09   
Description: Administrators can create new administrator accounts in the system. The system validates the input data and stores the administrator information securely.   
Input: Administrator details including name, email address, password, and permissions.   
Output: A new administrator object with attributes (AdminID, Name, EmailAddress, Password, Role) stored in the database.  
  
## 1.10 Administrator Management Function   
Function ID: FR-10   
Description: Administrators can manage existing administrator accounts by editing, disabling, or deleting them. The system updates the administrator's status and logs the action.   
Input: Target administrator ID and the action to be performed (edit, disable, delete).   
Output: Updated administrator object or administrator deletion confirmation, with a record created in the EmailRecord or InboxRecord table as needed.  
  
## 1.11 Contact Management Function   
Function ID: FR-11   
Description: Users and administrators can add, edit, or delete contacts in the user's contact list. The system validates the contact information and logs the action.   
Input: Contact details including name, email address, phone number, and address.   
Output: Updated contact list with a new, modified, or removed contact, and a record created in the EmailRecord or InboxRecord table as needed.  
  
## 1.12 Contact Addition Function   
Function ID: FR-12   
Description: Users can add a new contact to their contact list. The system validates the contact's email address and ensures it is not a duplicate.   
Input: New contact details including name, email address, phone number, and address.   
Output: A new contact object with attributes (ContactID, UserID, Name, EmailAddress, PhoneNumber, Address) stored in the database.  
  
## 1.13 Contact Deletion Function   
Function ID: FR-13   
Description: Users can delete a contact from their contact list. The system prompts for confirmation and optionally moves the contact to a trash or archive folder.   
Input: Selected contact(s) to be deleted.   
Output: The selected contact(s) are removed from the contact list, and a record is created in the EmailRecord or InboxRecord table as needed.

# External Description

# 2. External Interfaces  
  
## 2.1 User Interface Output  
  
The system interacts with users through a graphical user interface (GUI) that supports the following functionalities:  
  
- \*\*Email Creation Interface\*\*: Users can compose new emails by entering the recipient’s email address, subject, body content, and optionally attaching files, setting priority levels, and scheduling emails for later sending.   
 - \*\*Interaction Method\*\*: Form-based input with fields for recipient, subject, body, and additional options.  
 - \*\*Inputs\*\*: Recipient email address, subject, body content, optional attachments, priority level, and scheduling preferences.  
 - \*\*Outputs\*\*: Confirmation of the new email being saved with attributes (EmailID, Subject, Body, SenderID, ReceiverID, SendDate, Status).  
  
- \*\*Email Sending Interface\*\*: After composing or editing an email, users can send it to the recipient(s). The system provides feedback on the success or failure of the sending process.   
 - \*\*Interaction Method\*\*: Button click or command input to trigger the sending action.  
 - \*\*Inputs\*\*: A composed email object with valid recipient, subject, and body.  
 - \*\*Outputs\*\*: Confirmation of the email being sent, and a record in the EmailRecord table (RecordID, EmailID, ActionType, ActionDate, Status).  
  
- \*\*Email Receiving Interface\*\*: The system displays incoming emails in the appropriate user’s inbox. Users can view the subject, body, and sender information.   
 - \*\*Interaction Method\*\*: Inbox refresh or notification-based display.  
 - \*\*Inputs\*\*: An incoming email with attributes (Subject, Body, SenderID, ReceiverID).  
 - \*\*Outputs\*\*: The email is displayed in the user’s inbox with attributes (InboxID, UserID, EmailID, ReadStatus, DateAdded), and a record is created in the EmailRecord table.  
  
- \*\*Email Deletion Interface\*\*: Users can delete emails from their inbox or other folders. The system prompts for confirmation before deletion and may offer an option to move the email to a trash folder.   
 - \*\*Interaction Method\*\*: Selection of an email or multiple emails followed by a delete action.  
 - \*\*Inputs\*\*: Selected email(s) to be deleted.  
 - \*\*Outputs\*\*: The email(s) are removed from the folder or inbox, and a record is created in the InboxRecord table (RecordID, InboxID, ActionType, ActionDate, Status).  
  
- \*\*Email Viewing Interface\*\*: Users can view the content of an email, including subject, body, and attachments. The system marks the email as "read" and logs the action.   
 - \*\*Interaction Method\*\*: Clicking on an email to open it for viewing.  
 - \*\*Inputs\*\*: A selected email from the inbox or another folder.  
 - \*\*Outputs\*\*: Display of the email content (Subject, Body, Attachments), marking the email as "read", and a record created in the InboxRecord table.  
  
- \*\*User Creation Interface\*\*: Administrators can create new user accounts by entering user details such as name, email address, password, and role. The system validates the input and securely stores the user information.   
 - \*\*Interaction Method\*\*: Form-based input for user details.  
 - \*\*Inputs\*\*: Name, email address, password, and role.  
 - \*\*Outputs\*\*: Confirmation of the new user being created with attributes (UserID, Name, EmailAddress, Password, Role) stored in the database.  
  
- \*\*User Registration Interface\*\*: Administrators can register new users by providing full name, email address, password, and role. The system checks for email uniqueness and password strength before registration.   
 - \*\*Interaction Method\*\*: Form-based input for user registration.  
 - \*\*Inputs\*\*: Full name, email address, password, and role.  
 - \*\*Outputs\*\*: Confirmation of the user being registered with attributes (UserID, Name, EmailAddress, Password, Role) stored in the database.  
  
- \*\*User Management Interface\*\*: Administrators can edit, disable, or delete existing user accounts. The system updates the user status and logs the action.   
 - \*\*Interaction Method\*\*: Selection of a user and action (edit, disable, delete).  
 - \*\*Inputs\*\*: Target user ID and the action to be performed.  
 - \*\*Outputs\*\*: Updated user object or confirmation of deletion, with a record created in the EmailRecord or InboxRecord table.  
  
- \*\*Administrator Creation Interface\*\*: Administrators can create new administrator accounts by entering name, email address, password, and permissions. The system validates the input and securely stores the administrator information.   
 - \*\*Interaction Method\*\*: Form-based input for administrator details.  
 - \*\*Inputs\*\*: Name, email address, password, and permissions.  
 - \*\*Outputs\*\*: Confirmation of the new administrator being created with attributes (AdminID, Name, EmailAddress, Password, Role) stored in the database.  
  
- \*\*Administrator Management Interface\*\*: Administrators can edit, disable, or delete existing administrator accounts. The system updates the administrator status and logs the action.   
 - \*\*Interaction Method\*\*: Selection of an administrator and action (edit, disable, delete).  
 - \*\*Inputs\*\*: Target administrator ID and the action to be performed.  
 - \*\*Outputs\*\*: Updated administrator object or confirmation of deletion, with a record created in the EmailRecord or InboxRecord table.  
  
- \*\*Contact Management Interface\*\*: Users and administrators can add, edit, or delete contacts in the user’s contact list. The system validates the contact information and logs the action.   
 - \*\*Interaction Method\*\*: Contact list view with options to add, edit, or delete contacts.  
 - \*\*Inputs\*\*: Contact details including name, email address, phone number, and address.  
 - \*\*Outputs\*\*: Updated contact list with a new, modified, or removed contact, and a record created in the EmailRecord or InboxRecord table.  
  
- \*\*Contact Addition Interface\*\*: Users can add a new contact to their contact list by providing details such as name, email address, phone number, and address. The system validates the email and ensures there are no duplicates.   
 - \*\*Interaction Method\*\*: Form-based input for contact details.  
 - \*\*Inputs\*\*: Name, email address, phone number, and address.  
 - \*\*Outputs\*\*: Confirmation of the new contact being added with attributes (ContactID, UserID, Name, EmailAddress, PhoneNumber, Address) stored in the database.  
  
- \*\*Contact Deletion Interface\*\*: Users can delete a contact from their contact list. The system prompts for confirmation and may offer an option to move the contact to an archive or trash folder.   
 - \*\*Interaction Method\*\*: Selection of a contact followed by a delete action.  
 - \*\*Inputs\*\*: Selected contact(s) to be deleted.  
 - \*\*Outputs\*\*: The contact(s) are removed from the contact list, and a record is created in the EmailRecord or InboxRecord table.  
  
## 2.2 Hardware Interface Output  
  
This system does not require direct interaction with external hardware devices. All operations are performed via standard computing devices such as desktops, laptops, or mobile devices. There are no specific hardware communication protocols or supported device types that need to be defined for the system to function.  
  
## 2.3 Software Interface Output  
  
The system interacts with the following external software components and databases:  
  
- \*\*Email Database\*\*:   
 - \*\*Description\*\*: A database where all email objects are stored. Each email has attributes such as EmailID, Subject, Body, SenderID, ReceiverID, SendDate, and Status.   
 - \*\*Interaction Method\*\*: CRUD (Create, Read, Update, Delete) operations to manage email data.   
 - \*\*Data Format\*\*: Structured data stored in tables.   
 - \*\*Related Functional Requirements\*\*: FR-01, FR-02, FR-03, FR-04, FR-05  
  
- \*\*EmailRecord Database\*\*:   
 - \*\*Description\*\*: A database that logs email-related actions such as sending, receiving, and managing emails. Each record has attributes such as RecordID, EmailID, ActionType, ActionDate, and Status.   
 - \*\*Interaction Method\*\*: Insert operations to log actions and retrieve records for auditing or tracking purposes.   
 - \*\*Data Format\*\*: Structured data stored in tables.   
 - \*\*Related Functional Requirements\*\*: FR-02, FR-08, FR-10, FR-11, FR-13  
  
- \*\*Inbox Database\*\*:   
 - \*\*Description\*\*: A database where incoming emails are stored in a user's inbox. Each inbox entry has attributes such as InboxID, UserID, EmailID, ReadStatus, and DateAdded.   
 - \*\*Interaction Method\*\*: Insert and update operations to store and manage inbox entries.   
 - \*\*Data Format\*\*: Structured data stored in tables.   
 - \*\*Related Functional Requirements\*\*: FR-03, FR-04, FR-05  
  
- \*\*InboxRecord Database\*\*:   
 - \*\*Description\*\*: A database that logs actions related to inbox management, such as viewing or deleting emails. Each record has attributes such as RecordID, InboxID, ActionType, ActionDate, and Status.   
 - \*\*Interaction Method\*\*: Insert operations to log actions and retrieve records for auditing or tracking purposes.   
 - \*\*Data Format\*\*: Structured data stored in tables.   
 - \*\*Related Functional Requirements\*\*: FR-04, FR-05, FR-11, FR-13  
  
- \*\*User Database\*\*:   
 - \*\*Description\*\*: A database that stores user account information. Each user has attributes such as UserID, Name, EmailAddress, Password, and Role.   
 - \*\*Interaction Method\*\*: CRUD operations to manage user data.   
 - \*\*Data Format\*\*: Structured data stored in tables.   
 - \*\*Related Functional Requirements\*\*: FR-06, FR-07, FR-08  
  
- \*\*Administrator Database\*\*:   
 - \*\*Description\*\*: A database that stores administrator account information. Each administrator has attributes such as AdminID, Name, EmailAddress, Password, and Role.   
 - \*\*Interaction Method\*\*: CRUD operations to manage administrator data.   
 - \*\*Data Format\*\*: Structured data stored in tables.   
 - \*\*Related Functional Requirements\*\*: FR-09, FR-10  
  
- \*\*Contact Database\*\*:   
 - \*\*Description\*\*: A database that stores user contacts. Each contact has attributes such as ContactID, UserID, Name, EmailAddress, PhoneNumber, and Address.   
 - \*\*Interaction Method\*\*: CRUD operations to manage contact data.   
 - \*\*Data Format\*\*: Structured data stored in tables.   
 - \*\*Related Functional Requirements\*\*: FR-11, FR-12, FR-13  
  
## 2.4 Communication Interface Output  
  
The system utilizes the following communication interfaces to interact with external systems or users:  
  
- \*\*Email Server Communication Interface\*\*:   
 - \*\*Description\*\*: The system communicates with an external email server to send and receive emails. This includes verifying the sender, authenticating the recipient, and ensuring secure transmission.   
 - \*\*Interaction Method\*\*: SMTP (Simple Mail Transfer Protocol) for sending emails and IMAP/POP for receiving emails.   
 - \*\*Inputs\*\*: Composed email object (Subject, Body, SenderID, ReceiverID, etc.).   
 - \*\*Outputs\*\*: Confirmation of successful sending or receipt of an email.  
  
- \*\*Web Browsing Interface\*\*:   
 - \*\*Description\*\*: The system is accessible through a web browser, allowing users and administrators to interact with it using a standard HTTP/HTTPS protocol.   
 - \*\*Interaction Method\*\*: Web-based GUI accessible via a URL.   
 - \*\*Inputs\*\*: User or administrator login credentials, form inputs for email and user management.   
 - \*\*Outputs\*\*: Web page rendering with dynamic content updates based on user actions.  
  
- \*\*Notification Interface\*\*:   
 - \*\*Description\*\*: The system may send notifications to users via email or in-app alerts to inform them of new emails, system updates, or administrative actions.   
 - \*\*Interaction Method\*\*: Email or internal messaging system.   
 - \*\*Inputs\*\*: Event triggers such as a new email arrival or user action.   
 - \*\*Outputs\*\*: Notification message to the user with relevant details.  
  
- \*\*Inter-system Communication Interface\*\*:   
 - \*\*Description\*\*: The system may need to interact with other internal or external systems for user verification, email routing, or data synchronization.   
 - \*\*Interaction Method\*\*: RESTful API or database-level queries.   
 - \*\*Inputs\*\*: User IDs, email IDs, or system event data.   
 - \*\*Outputs\*\*: Responses or data updates from the external system.  
  
This section ensures that all external data sources and interfaces referenced in the functional requirements are clearly defined and categorized. Developers can use this information to understand the necessary integration points and data flows within the system.