

# Mail Simulation Application Development

ELF / HTD Training Programs  
Java Full Stack - Use Cases

Table of Contents

1. INTRODUCTION..... 2

    1.1 Setup Checklist for Project ..... Error! Bookmark not defined.

    1.2 Instructions..... Error! Bookmark not defined.

2. PROBLEM STATEMENT..... 2

    2.1 Objective..... 2

3. IMPLEMENTATION..... 2

    3.1 Functional Requirements..... 2

    3.2. System implementation..... 3

### 1. INTRODUCTION

This document outlines the Project. The project is to develop a Mail Simulation for an organization. This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules.

### 2. PROBLEM STATEMENT

People use to communicate through letter and telegram, which is not faster. So mailing system got introduced to overcome this problem. Mail Simulation Application is an application where user can interact with each other through mail. This project doesn't deal with any SMTP or IMAP, it's just a simulation and created by just database and java code.

#### 2.1 Objective

To simulate mailing system so users can communicate easily. This application will have users and those users can communicate through mail. This will reduce time and increase efficiency.

#### 2.2 Abstract of the Project

This project is aimed to simulate mailing system. User can register to the application. User can login to the application. User can send mail to other user using mail id. And user can see the mail what he/she received.

### 3. IMPLEMENTATION

#### 3.1 Functional Requirements

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate, you can make appropriate assumptions and proceed.

There is just one type of user, that is normal user.

- The user can register to the application by providing all the data needed by the application
- If the user enters existing email id while registering, it should show proper error message
- The user can login using his/her credentials
- If the user gives invalid credentials, then it should show proper error message in the same page
- If the user forgot the password, he/she can raise a request to generate new password
  - User has to provide answers to the security questions
  - If the answers are correct, then user should able to change the password
  - If the answers are wrong, then application should show proper error message in the same page
- Once the user logged in the user can perform
  - Compose and send a mail with to-address, subject and message to an existing user in this application (if that user doesn't exist then he/she has to get a mail with subject as "Mail Delivery Failed")

- View inbox and view all mails the user received in it
- View sent items where it contains all the mail the user sent to other user
- View delete items where it contains all the mail which is deleted by the user
- View draft mails where it contains all stored draft mails which can be edited later
- Logout hyperlink to logout from the session

### 3.2. System implementation

- Login page contains following information
  - Text box to enter email id
  - Text box to enter password
  - Login button to login into the application
  - Hyperlink to request for forgot password
  - Hyperlink to register to the application
- Register page contains following information
  - Text boxes to enter required data
  - In that email id and password are mandatory
  - Register button to register as a user
  - Hyperlink to go to login page
- Change password page
  - Text box to accept the password
  - Text box to accept the conform password
  - Change Password button to change the password
- Home page
  - Hyperlinks to open received mails
  - Hyperlink to go to “inbox” (once login by default it will open inbox only)
  - Hyperlink to go to “sent items”
  - Hyperlink to go to “deleted items”
  - Hyperlink to “compose mail”
  - Hyperlink to “logout”
- Inbox page
  - All mails displayed as hyperlinks with from-address and subject (if user click on this it will open the mail)
  - Hyperlink in front of each mails to delete the mail
  - Hyperlink to go to “inbox”
  - Hyperlink to go to “sent items”
  - Hyperlink to go to “deleted items”
  - Hyperlink to “compose mail”
  - Hyperlink to “logout”
- Sent item page
  - All mails displayed as hyperlinks with to-address and subject (if user click on this it will open the mail)

- Hyperlink in front of each mails to delete the mail
- Hyperlink to go to “inbox”
- Hyperlink to go to “sent items”
- Hyperlink to go to “deleted items”
- Hyperlink to “compose mail”
- Hyperlink to “logout”
- Deleted item page
  - All mails displayed as hyperlinks with from/to address and subject (if user click on this it will open the mail)
  - Hyperlink in front of each mails to delete the mail
  - Hyperlink to go to “inbox”
  - Hyperlink to go to “sent items”
  - Hyperlink to go to “deleted items”
  - Hyperlink to “compose mail”
  - Hyperlink to “logout”
- Compose mail page
  - Text box to enter to-address
  - Text box to enter subject
  - Text area to enter message
  - Send button to send the mail to the respective user
- Mail page (this page is displayed when user click on any mail in the application)
  - From/to address
  - Subject of the mail
  - Message content of the mail
  - Hyperlink in front of each mails to delete the mail
  - Hyperlink to go to “inbox”
  - Hyperlink to go to “sent items”
  - Hyperlink to go to “deleted items”
  - Hyperlink to “compose mail”
  - Hyperlink to “logout”
- Draft page
  - All mails displayed as hyperlinks which is composed and not sent
  - Hyperlink in front of each mails to delete the mail
  - Hyperlink to go to “inbox”
  - Hyperlink to go to “sent items”
  - Hyperlink to go to “deleted items”
  - Hyperlink to “compose mail”
  - Hyperlink to “logout”

## 4. Database Specification

### User\_info

Id	int (primary key) (auto generate)
User_name	Varchar(255)
Email	Varchar(255) (unique)
Password	Varchar(255)

### Mail\_info

Id	int (primary key) (auto generate)
From_id	int (foreign key)
To_id	int (foreign key)
Subject	Varchar(255)
Message	Varchar(255)
Status	Varchar(255)