Automation Mailing System using Templates for Bulk Mailing

Design Document

C-DAC, Bangalore

Version 01

May 2024

Prepared By:

Sagar Deware

1.0 Introduction

Introduction to the project.

1.1 Purpose

The primary objective of the project, ’Automation Mailing System Using Templates for Bulk Mailing,’ is to revolutionize the conventional mailing process by introducing a highly automated and streamlined system.

By harnessing the power of automation, this project aims to significantly enhance efficiency and productivity in handling bulk mail. The overarching goal is to cater to the diverse needs of users, including employees and Students, by providing them with a sophisticated yet user-friendly platform. This platform will empower users to effortlessly create, customize, and send bulk emails using pre-designed templates, thereby saving valuable time and resources.

Furthermore, the project seeks to mitigate common challenges associated with manual mailing processes, such as errors, delays, and inconsistency in communication. By leveraging automation, it promises to deliver swift and accurate mailings, ensuring timely dissemination of information and correspondence.

Ultimately, the Automation Mailing System endeavors to revolutionize how organizations and individuals manage their communication needs, offering a seamless and efficient solution that aligns with modern standards of productivity and convenience.

1.2 Intended Audience

This document is intended for top management, developers, project managers, marketing staff, the testing team, CREST team. This document explains the requirements of all the required modules. 1 Developers should go through all the sections in the given order.

1.3 References

- Functional Requirement Specifications (FRS)
- System Requirement Specification(SRS)

2.0 Acronyms, terms and definitions

FRS	Functional requirements specification
Concerned Person	User refers to department/employee/worker

3.0 Assumptions and constraints

- The Database should be stored in the System for testing(Before Deployment).

4.0 Basic Design approach

- The project follows the Minimalist Design approach for solution design.
- The Design will focus more on usability and ease of understanding.

5.0 Risks

- Using Outdated UnSecure Port can risk the communication.

6.0 System overview

The ’Automation Mailing System using templates for Bulk mailing’ project encompasses the development and implementation of a comprehensive software solution tailored specifically for use within CDAC (Centre for Development of Advanced Computing). This system will serve as an integral tool for every entity within CDAC, spanning across departments, teams, and individual users.

7.0 Architecture Design

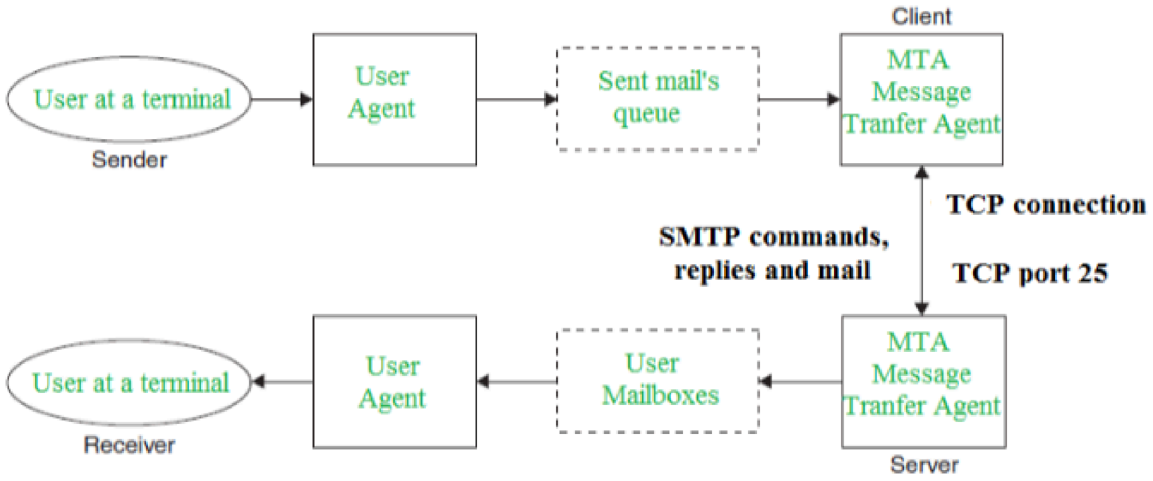
The Main Component of The project is Sending email with ease using Templates which is done through SMTP Connection.

The Architecture Design for SMTP Connection:

1. **Composition of Mail:** A user sends an e-mail by composing an electronic mail message using a Mail User Agent (MUA). Mail User Agent is a program which is used to send and receive mail. The message contains two parts: body and header. The body is the main part of the message while the header includes information such as the sender and recipient address. The header also includes descriptive information such as the subject of the message. In this case, the message body is like a letter and the header is like an envelope that contains the recipient's address.
2. **Submission of Mail:** After composing an email, the mail client then submits the completed email to the SMTP server by using SMTP on TCP port 25.
3. **Delivery of Mail:** E-mail addresses contain two parts: username of the recipient and domain name. For example, vivek@gmail.com, where "vivek" is the username of the recipient and "gmail.com" is the domain name.

   If the domain name of the recipient's email address is different from the sender's domain name, then MSA will send the mail to the Mail Transfer Agent (MTA). To relay the email, the MTA will find the target domain. It checks the MX record from the Domain Name System to obtain the target domain. The MX record contains the domain name and IP address of the recipient's domain. Once the record is located, MTA connects to the exchange server to relay the message.

4. **Receipt and Processing of Mail:** Once the incoming message is received, the exchange server delivers it to the incoming server (Mail Delivery Agent) which stores the email where it waits for the user to retrieve it.
5. **Access and Retrieval of Mail:** The stored email in MDA can be retrieved by using MUA (Mail User Agent). MUA can be accessed by using login and password.



Architecture Overview:
The automated mailing system will follow a client-server architecture model. It will consist of several components responsible for different tasks such as managing templates, recipient lists, sending emails, and logging activities. The architecture will be designed to be scalable, reliable, and secure.

Components:

- Client Interface:

1. This component provides a user-friendly interface for users to interact with the system. It allows users to create and customize email templates, manage recipient lists, schedule mailings, and view mailing logs.

2. The client interface can be a web application accessible via a browser or a desktop application.

- Server Backend:

1. The server backend handles the core functionalities of the system. It consists of several modules responsible for different tasks.
2. Modules include:

A. Template Management: Handles creation, storage, and retrieval of email templates. Supports various template formats such as HTML, plain text, and rich media.

B. Mailing Engine: Responsible for sending emails based on templates and recipient lists. Implements throttling mechanisms to prevent server overload and monitors email delivery status.

C. Security: Implements authentication and authorization mechanisms to control user access. Ensures data encryption during transmission and storage

- Database:

1. The database stores persistent data such as email templates, recipient lists, mailing logs, and user information.
2. It utilizes a relational database management system (RDBMS) like MySQL or PostgreSQL to ensure data integrity and scalability.
3. The database schema is designed to support efficient retrieval and manipulation of data.

- Mail Transfer Agent (MTA):

1. The MTA is responsible for transmitting emails from the server to the recipients' mail servers.
2. It utilizes standard email protocols such as SMTP (Simple Mail Transfer Protocol) for sending emails and IMAP (Internet Message Access Protocol) for receiving bounce notifications and tracking email delivery status.

8.0 Data Design

The initial data table is only for development purpose, it will be updated after Completion of basic Development of project.

The data table :

Table name : data

Columns: ID , Name , email, file name , file path , Dynamic data (data replacing from template).

This table is for sending email for providing Certificates to the students. Where Certificate will be attached to email which is going to be fetched using file name and path.

Idx (Primary Key)	Field Name	Data Type	Description
ID	ID	int	Student id for student filtration
	Name	varchar	to replace it in template
	Email	varchar	email will be sent to this email after fetching.
	File name	varchar	to add filename in attachment
	File Path	varchar	to attach file in email
Indexes (Foreign Keys)			

9.0 Component Design

9.1.1 Authentication Workflow:

User Authentication:

A. When a user accesses the client interface (web application or desktop application), they are prompted to log in.
B. The login credentials (username and password) are submitted to the server backend for authentication.

Server-side Authentication:

1. The server backend receives the login request and validates the user credentials against the stored user database.
2. If the credentials are valid, the server generates a session token for the user.

Session Management:

1. The session token is a unique identifier that represents the authenticated user's session.
2. This token is securely stored on the client-side (e.g., as a browser cookie or in local storage) and sent with each subsequent request to the server.

3. The server maintains a session store mapping session tokens to user identities.

Authorization:

1. Once authenticated, the server backend verifies the user's authorization level to determine the actions they can perform.
2. Authorization can be role-based, where each user is assigned a role (e.g., administrator, content creator) with specific permissions.
3. Access to sensitive operations such as template management, recipient management, and sending mailings is restricted based on the user's role.

Token Expiry and Renewal:

1. To enhance security, session tokens have a limited validity period (e.g., 30 minutes).
2. Before the token expires, the client interface can request a token renewal by sending a refresh token along with the expired token.
3. The server backend validates the refresh token and issues a new session token if the user is still authenticated.

Logout:

1. When a user chooses to log out of the system, the client interface sends a logout request to the server.
2. The server invalidates the session token associated with the user's session, effectively logging them out.
3. Additionally, the client interface clears the session token stored locally, ensuring the user is fully logged out.

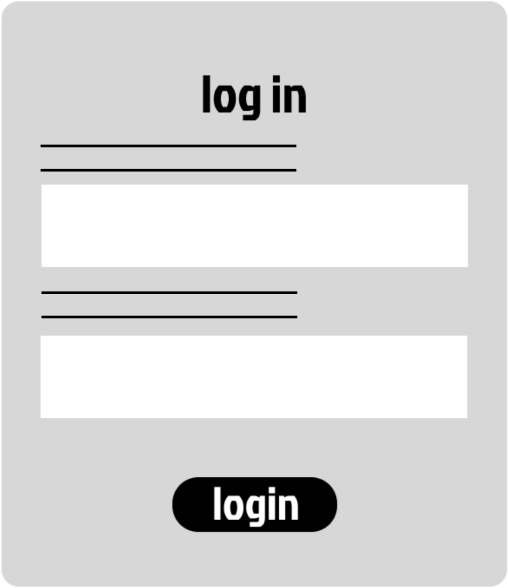
9.1.2 Template management

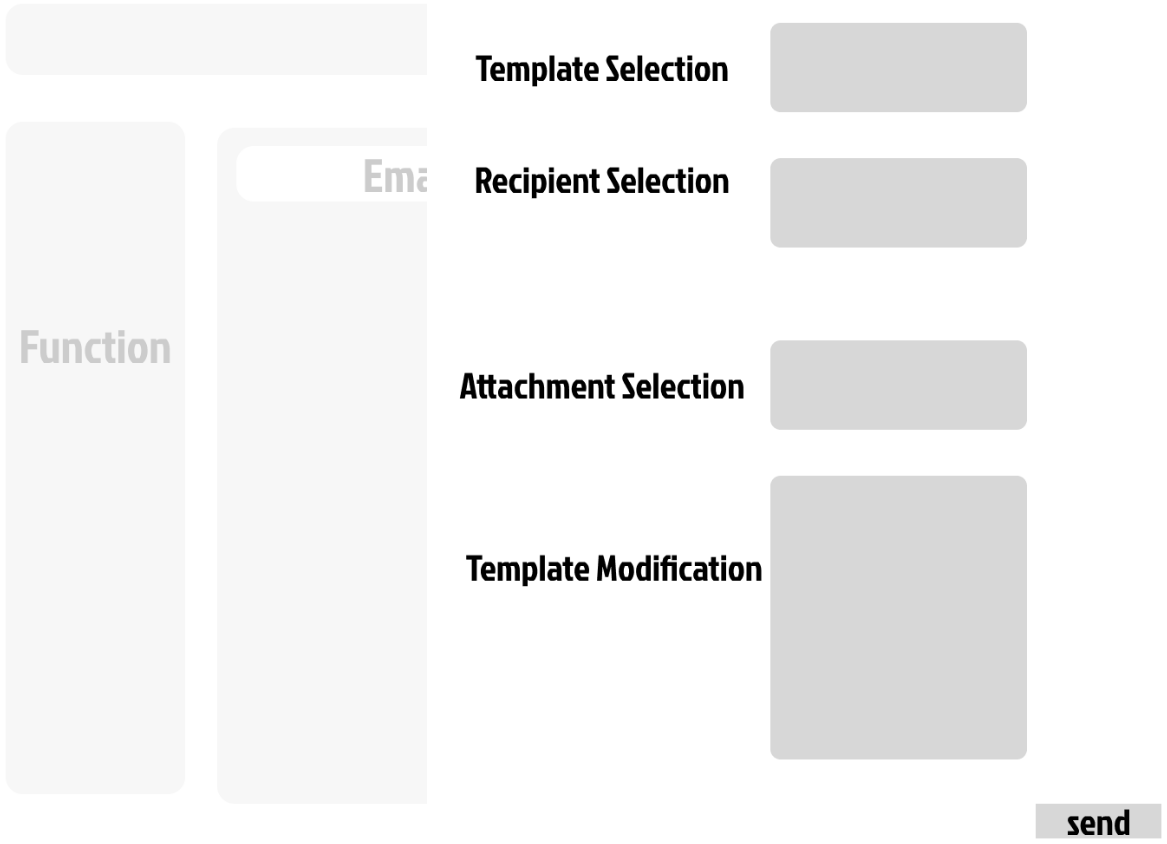
In this Component , an admin or concerned person can add , remove , modify the template. The admin can Archive the template as it pleases. the admin can give access to templates to a specific person.

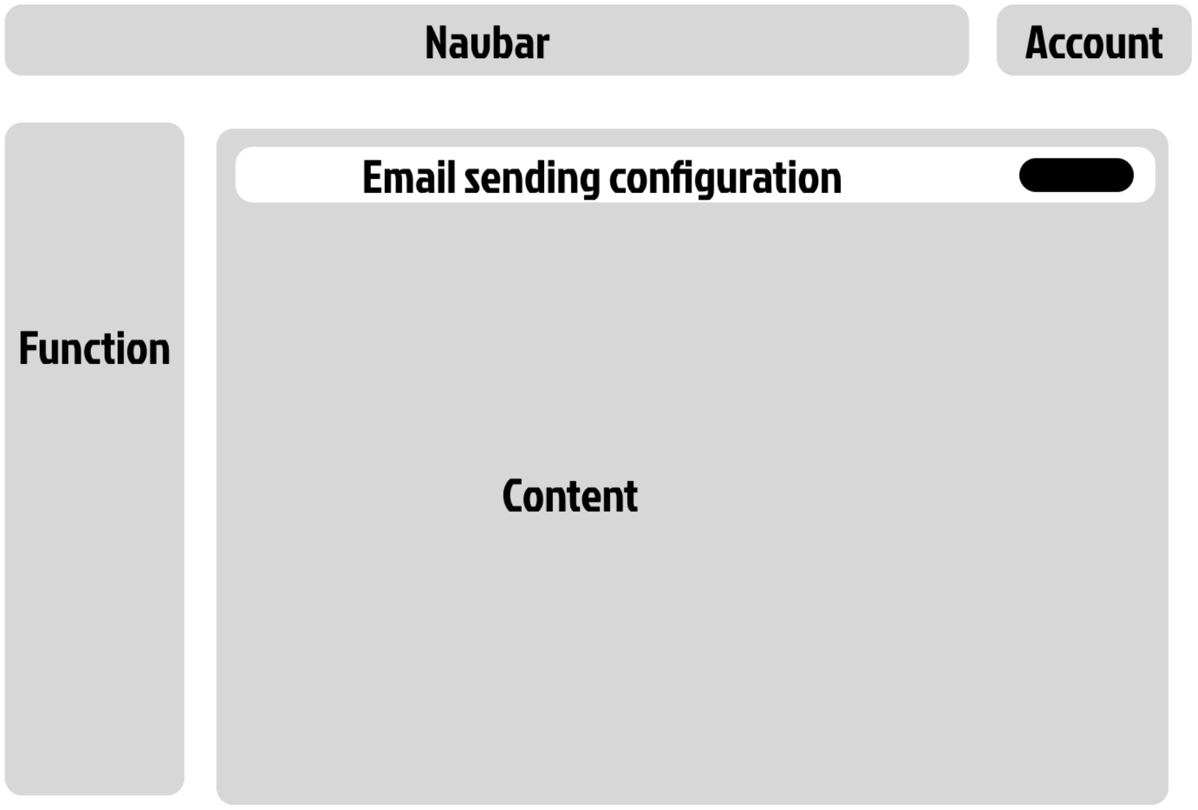
9.1.3 Submission of Email

In the System, Different module where Email sending is required, this component gives the functionality to direct send to multiple recipients with multiple formats. for explanation, the module where Admin manages the hostel allotment. if the admin wants to send an email to all students about the allotment details. He chose the attachment and template for that specific task. The system will automatically fetch all required data from the database and send it to the recipients.

10.0 Interface Design




Function
Ema
Template Selection
Recipient Selection
Attachment Selection
Template Modification
send


Naubar
Account
Function
Email sending configuration
Content

Appendix A/B/C – Appendix Name

Details of appendix