Software Requirement Specification for Bulk Mail Blocking/Unblocking

Name	NAREN SHANKAR N
Roll no	7376221CS239
Seat no	251
Project ID	11
Problem Statement	Bulk Mail Blocking / Unblocking

1. Introduction

1.1. Purpose:

This paper aims to provide a thorough overview of the website for blocking and unblocking mail. The document will elucidate the objectives and characteristics of the system, as well as its interfaces, functionalities, operational limitations, and response to external stimuli.

1.2. Scope of Project:

- Rather than blocking and unblocking mail IDs for each individual student, this software solution will act as a portal for blocking and unblocking mail IDs of students in a single click. The dashboard will include information on blocked and unblocked mail addresses, along with the duration and number of block and unblock cycles.
- Students' mail IDs can be blocked or unblocked by administrators. If the student's email address is blocked, they can fill out an unblock request form

to get it unblocked.

2. System Overview:

2.1. Users:

1. **Students**:

They can view their dashboard displaying the status of mail id, number of times blocked and the reason for getting blocked.

2. **Admins:**

Block the mail id of the students , unblock the mail id of the students by getting the respective unblock form

2.2. Features:

1. **Admin Access:**

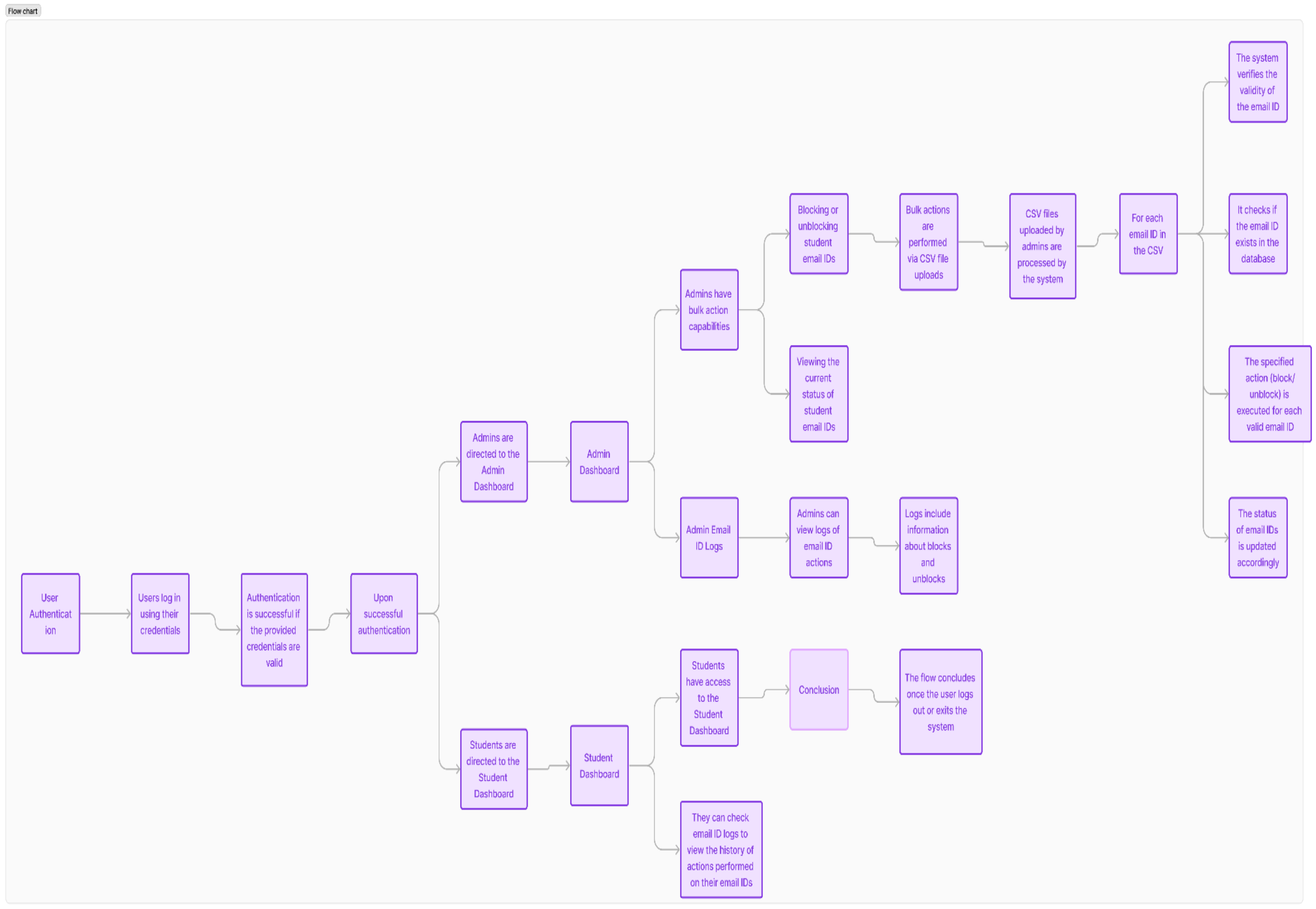
Only Admins can login to the webpage. Admin can view the status of the students mail id, number of times mail id blocked / unblocked, block and unblock mail id.

2. **Admin's Analytical Dashboard:**

Admin can view the number of students they have blocked and unblocked with the exact time and can view the blocked and unblocked mail id of the particular student.

3. **Student Access:**

The student can login to the webpage using their official mail id. They can view their dashboard displaying their mail id status, number of times it got blocked and the reason for getting blocked.


Flow chart
User Authentication
Users log in using their credentials
Authentication is successful if the provided credentials are valid
Upon successful authentication
Admins are directed to the Admin Dashboard
Admin Dashboard
Admins have bulk action capabilities
Blocking or unblocking student email IDs
Bulk actions are performed via CSV file uploads
CSV files uploaded by admins are processed by the system
For each email ID in the CSV
The system verifies the validity of the email ID
It checks if the email ID exists in the database
The specified action (block/ unblock) is executed for each valid email ID
The status of email IDs is updated accordingly
Viewing the current status of student email IDs
Admin Email ID Logs
Admins can view logs of email ID actions
Logs include information about blocks and unblocks
Students are directed to the Student Dashboard
Student Dashboard
Students have access to the Student Dashboard
Conclusion
The flow concludes once the user logs out or exits the system
They can check email ID logs to view the history of actions performed on their email IDs

3. System Requirements Specification:

3.1 Functional Requirements:

- **User Management:**
  - Administrators have access control over specialized features and an analytics dashboard. Students have no access to this particular portal.

- **Admin Dashboard:**
  - Admins can view a list of all students mail id status
  - Admins can view details of each student mail id.

  - Admins can block or unblock student’s mail id

- **Analytics Dashboard:**
  - Admin can view the number of blocked and unblocked mail id’s with certain time.

3.2. Non-Functional Requirements:

- **Performance**: The system must do the function of blocking or unblocking within a few seconds and the students should be able to have access to their respective mail id.
- **Security**:

  1. Authorization and Authentication: Strong authorization and role-based authorization provide safe user access.
  2. Encryption: Make use of industry-standard protocols to encrypt sensitive data both in transit and at rest.
  3. Limit access to administrative functions to individuals who are permitted.
  4. Logging and Monitoring: Put in place logging systems to keep tabs on user activity and identify questionable conduct.
  5. To ensure encrypted connection between clients and servers, it is recommended to enforce HTTPS.

- **Usability**:
  1. Clear Instructions: Provide clear instructions and guidance for users to understand how to block or unblock emails effectively.
  2. Responsiveness: Ensure the website is responsive and works seamlessly across different devices and screen sizes.
  3. Error Handling: Implement user-friendly error messages and feedback to guide users in case of mistakes or issues.

4. Feedback Mechanisms: Incorporate feedback mechanisms to gather user input and improve the system's usability over time.

- **Reliability**:
  1. Stable Servers: Employ robust infrastructure for minimal downtime.
  2. Error Management: Handle errors effectively to maintain stability.
  3. Redundancy Measures: Utilize backups and failover systems.
  4. Continuous Monitoring: Proactively detect and address issues.
  5. Regular Maintenance: Ensure system health with scheduled updates.

- **Scalability**: In scalability, prioritize flexible infrastructure and modular architecture to accommodate dynamic growth. Utilize load balancing for even traffic distribution and implement database sharding for scalable data management. Continuous monitoring enables automated resource adjustments, ensuring seamless performance as demand fluctuates.

Stack:

Front End	HTML, CSS, JS
Backend	Python with Django
Data Base	MongoDB