BANNARI AMMAN INSTITUTE OF TECHNOLOGY
An Autonomous Institution Affiliated to Anna University - Chennai, Accredited by NAAC with A+ Grade
Sathyamangalam - 638401 Erode District, Tamil Nadu, India

Student Name: SANKARANARAYANAN M

Seat No: 169

Project ID: 10

Project title: Bulk Mail Blocking/Unblocking

Technical Components

Component	Tech Stack
Backend	Spring Boot
Frontend	React Js
Database	MySQL
API	RESTful services

PROBLEM STATEMENT:

Build a portal system to block/unblock bulk Bitsathy email ID of the students.

Introduction:

Purpose:

The purpose of automating email ID blocking and unblocking is to streamline the process and mitigate challenges associated with manual *9deliver a user-friendly solution that enhances efficiency, accuracy, and security in email ID management, ultimately improving organizational productivity and user experience

System Overview:

Users: Admin-Staff manages Bitsathy mail id include blocking/Unblocking

Dependencies:

- Integration with Google OAuth for user authentication.
- Consistent performance and availability of the existing email server.

Key Features:

User Authentication:

- Users log in using their credentials.
- Authentication is successful if the provided credentials are valid.

Dashboard Differentiation:

- Upon successful authentication:
- Admins are directed to the Admin Dashboard.

Admin Dashboard:

- Admins have bulk action capabilities, including:
- Blocking or unblocking student email IDs.
- Viewing the current status of student email IDs.
- Bulk actions are performed via CSV file uploads.

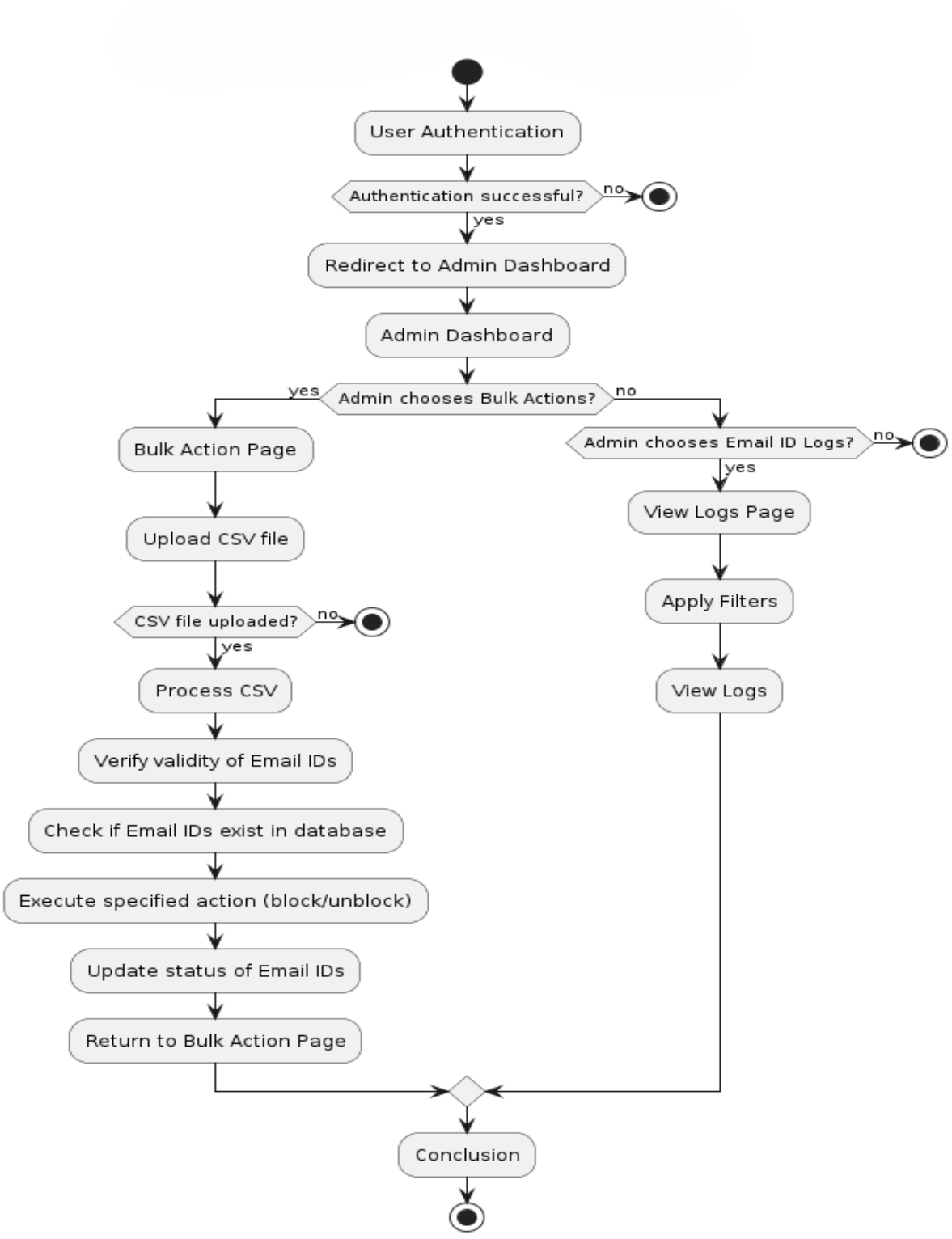
Bulk Action Processing:

- 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.

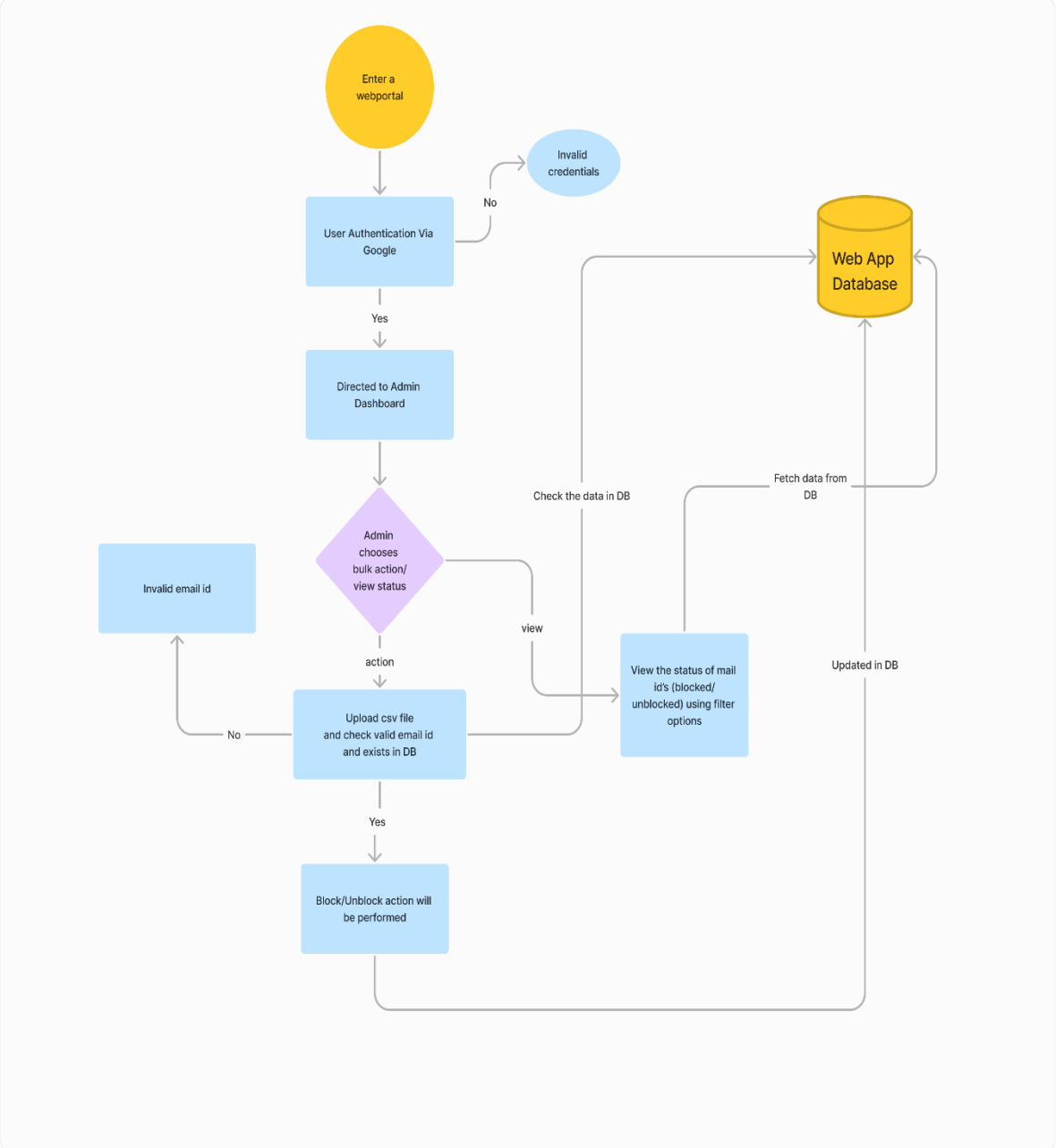
Admin Email ID Logs:

- Admins can view logs of email ID actions using filter options:
- Logs include information about blocks and unblocks.

Flow Chart:



Work Flow Diagram:



Work Flow:

User Authentication:

- Implement secure user authentication via username and password.
- Utilize industry-standard encryption protocols to safeguard user credentials.
- Provide multi-factor authentication options for enhanced security.

Admin Dashboard:

- Design an intuitive and visually appealing admin dashboard interface.
- Enable admins to perform bulk actions seamlessly, such as blocking or unblocking email IDs.
- Include interactive data visualization components to display email ID status and trends effectively.

Bulk Action Processing:

- Support bulk action processing via CSV file uploads with error handling for invalid data.
- Implement asynchronous processing to handle large datasets efficiently without impacting system performance.
- Ensure transactional integrity to maintain data consistency during bulk actions.

Admin Email ID Logs:

- Develop comprehensive logging functionality to record all email ID actions performed by admins.
- Enable advanced filtering and search capabilities to facilitate easy retrieval of log data.
- Implement log rotation and archival strategies for efficient storage management.

User Interface:

- Design a responsive and accessible user interface with a consistent layout and navigation structure.
- Incorporate intuitive feedback mechanisms to guide users through authentication and action processes.
- Prioritize accessibility standards to accommodate users with diverse needs and disabilities.

Security:

- Employ robust security measures, including HTTPS encryption and secure token-based authentication.
- Implement role-based access control to restrict admin functionalities based on user roles and permissions.
- Conduct regular security audits and vulnerability assessments to identify and mitigate potential risks.

Scalability and Performance:

- Architect the system for horizontal scalability to accommodate future growth and increased user demand.
- Optimize database queries and data access patterns for efficient performance under high load conditions.

- Monitor system performance metrics and implement auto-scaling mechanisms to ensure optimal resource utilization.

Error Handling:

- Implement comprehensive error handling mechanisms with detailed error messages for users.
- Log errors and exceptions centrally for monitoring and troubleshooting purposes.
- Provide user-friendly error resolution suggestions to guide users through problem resolution steps.

Session Management:

- Manage user sessions securely with session timeouts and secure session storage mechanisms.
- Implement CSRF protection and session fixation prevention techniques to enhance session security.
- Support session persistence across multiple devices for seamless user experience.

Data Integrity:

- Enforce data validation rules to ensure the integrity of input data and prevent data corruption.
- Implement database constraints and referential integrity checks to maintain data consistency.
- Utilize database transactions to ensure atomicity and isolation of bulk action operations.