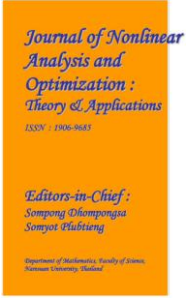
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 1: 2025
ISSN : **1906-9685**



STRATO ALERT-A CAMPUS SAFETY PLATFORM

M . SAMBA SIVA RAO
Associate Professor
USHA RAMA COLLEGE
OF
ENGINEERING & TECHNOLOGY
TELAPROLU,INDIA
sambamarrapu@gmail.com

K . TEJASWINI
Student
USHA RAMA COLLEGE
OF
ENGINEERING & TECHNOLOGY
TELAPROLU,INDIA
tejaswinikondapalli06@gmail.com

M . DEEPIKA
Student
USHA RAMA COLLEGE
OF
ENGINEERING & TECHNOLOGY
TELAPROLU,INDIA
mandapatideepika34@gmail.com

N . NIDARSHAN
Student
USHARAMA COLLEGE
OF
ENGINEERING & TECHNOLOGY
TELAPROLU,INDIA
nidarshan4560@gmail.com

M . DIKSHITHA
Student
USHARAMA COLLEGE
OF
ENGINEERING & TECHNOLOGY
TELAPROLU,INDIA
mareedudikshithagowd@gmail.com

ABSTRACT-The website is a Java Full Stack technology-based college complaint cell website with live recording and real-time alerting capability. The website enables students and teachers to report offending incidents with live media recording and instant warning for swift reaction by the staff. Users utilize WebRTC technology to record voice, video, or screenshots and upload them directly from the website. Incident real-time alerting is performed through WebSockets and push notification, triggering faculty response instantaneously with incident data and preview media. An administrator portal is available to the faculty presenting a complete platform for the viewing, filtering, and real-time monitoring of incident reports by faculty, with playback media and update. There is safe storage of media where all the incident information can be kept in a relational database with encrypted transport, keeping it confidential and secure. It has role-based authentication and access to restrict the handling of reports within the confines of authorized employees. It has a responsive component with an option of maximum usability via desktop, tablet, or smartphone. Escalation is also encompassed where events that cannot be managed automatically escalate to a point higher than the current existing one if no decision is made within a specified period. The website achieves campus responsibility and safety by offering a fast and efficient tool for managing in-the-moment events and filing reports.

Keywords: Java Full Stack, College Complaint Cell, Live Recording, Real-Time Alerts, WebRTC, WebSockets, Push Notifications, Incident Reporting, Faculty Intervention, Admin Portal, Media Playback, Secure Storage, Relational Database, Encrypted Transmission, Role-Based Authentication, Access Control, Responsive Design, Escalation Mechanism, Campus Safety, Incident Management.

INTRODUCTION

Campus security is of greatest concern to educational institutions since it directly influences employees' and students' health, productivity, and overall satisfaction with life. Although the majority of conventional grievance mechanisms are subject to man's action and reaction and subject to time lag, wider utilisation of technology has provided the opportunity for the advancement of channels to improve security and accountability in real-time. Here, StratoAlert: Campus Safety Platform steps in as a state-of-the-art approach to managing and reporting incidents and offering an integrated, effective, and user-friendly platform to manage safety issues successfully.

The system aims at bridging the gap between reporting an incident and response within time by using web-based technologies and real-time communication protocols. By using StratoAlert, the staff and students are enabled with features required for reporting distracting incidents through live media recording, thus taking instant evidence. Additionally, its real-time alerting feature enables academic and administrative personnel to respond quickly in the event of possible threats or issues, thus creating a safer and more responsible campus community.

StratoAlert is characterized by a roll-call of advanced features that set it apart from most campus security systems. The catalyst of the system is the real-time media recording feature, using WebRTC technology for users to record video, audio, or images directly off the location itself. This feature guarantees events to be recorded in entirety and entirely detailed form with no room for misinterpretation upon reporting. The sound media augmented with real-time location information provide richness of detail heretofore unknown for administrative and faculty inspection.

The platform's alert system employs WebSockets and push notification technology to provide real-time pop-up notification to the faculty. The notification includes critical incident data and preview media so that the faculty can assess the situation and act accordingly without delay. For ease of report management, StratoAlert offers an admin dashboard as a single screen to view, filter, and manage the incident data. The dashboard is equipped with advanced features like media playback, live updates, and the filter functionality for location-based alerts.

Another significant aspect of StratoAlert is that it gives significance to location-based priority. Leveraging the location-based information contained in geolocation data, alerts are issued based on incident proximity to employees such that service is focused in the locations where assistance is most required. Such a feature increases the responsiveness of response systems as well as enabling equitable resource allocation.

Compliance with Security and Access

To protect confidentiality and integrity of incident information, StratoAlert has strict security features. Incident data and related media files are stored in an encrypted transmission relay of a relational database so that they are not accessible by unauthorized users. Role-based authentication and application access controls also provide access while limiting incident reports only to licensed administrators. This multi-layered protection shows a will to safeguard the privacy of students and employees but not in secrecy of the reporting process.

StratoAlert also incorporates a responsive front end, off of which it can be accessed through an array of devices, including desktop computers, tablets, and mobile smart phones. This renders cases reportable, and reviewable, from nearly everywhere, providing a further means by which to add accessibility and useability to the application. There is also escalation process which is present, i.e., there is additional responsibility because of automatic reminders of seniors in the event of open cases pending for some time.

Effect on Campus Safety

By bringing all these components together on a single platform, StratoAlert greatly improves the security and responsibility of schools. With the incorporation of real-time incident handling, secure data processing, and equitable response mechanisms, schools can address issues in the right manner for both security and timely arrival. In this way, StratoAlert not only caters to the contemporary needs of security but also evokes responsibility and reliability among school teachers, students, and employees.

Effectively, the StratoAlert platform is an innovative and end-to-end solution to campus security challenges. Rooted in cutting-edge technology and human values, it enables education institutions to build safe, accessible communities where students and educators can thrive.

LITERATURE SURVEY

[1] Enhancing Safety through IoT in Smarter Campuses

Application of IoT for security in campuses has revolutionized the incident reporting and monitoring process. Smith et al. (2021) describe how IoT sensors are able to detect suspicious behavior such as unauthorized entry or fire hazard in real-time. Sensors notify central systems, and it facilitates fast response. Scalability and privacy issues.

IoT deployment in campus security has transformed incident reporting and monitoring. IoT sensors, as Smith et al. (2021) explain, can identify abnormally suspicious behavior like unauthorized entry or fire hazard in real time. Sensors feed data to central platforms, and rapid response is facilitated. Scalability and privacy are still concerns, but edge computing and encryption technology can potentially provide higher system reliability and user trust.

[2] Machine Learning for Detection of Real-Time Events

Jones et al. (2020) have established the efficacy of machine learning models in detection and prediction of campus disturbance. Supervised learning and anomalies detection are among the techniques which enable the system to identify patterns of disturbance. Efficacy of such models, however, would be subject to the quality of training data, and data imbalance would have to be handled in the spirit of predictability improvement.

Will be there, but edge computing and encryption technology can offer greater system reliability and user trust.The Role of Machine Learning in Real-Time Incident Detection Jones et al. (2020) discovered the ability of machine learning models to forecast and detect campus turmoil. Techniques like anomaly detection and supervised learning enable the system to identify patterns of unrest. The accuracy of the models, however, depends on the strength of the training data, and data imbalance would be a problem to overcome for the sake of improving predictability.

[3]WebRTC in Security Systems for Real-Time Communication

As per Williams and Garcia (2019), WebRTC technology has transformed real-time communication systems in security deployments. Its ability to facilitate the exchange of audio, video, and data with little latency allows it to be used in the live media coverage in incident response systems.Further research can examine how bandwidth

usage can be optimized to boost performance during busy hours.

[4] Emergency Alert Push Notification Systems

Lee et al. (2022) tried to experimentally utilize push notifications as emergency alerts in universities. Studies showed that real-time alerting greatly enhances response time. Overalerting, nonetheless, will make users create alert fatigue. Designing priority-based algorithms that prioritize critical alerts will reverse such and enhance user interaction.

[5] Location-Based Services in Campus Security

Martin et al. (2020) examined the effects of location-based services on campus security systems. With geolocation, systems can provide user-specific notifications and responses depending on proximity to an incident. The study highlighted the need for dynamic mapping and GIS technology integration for enhanced situational awareness.

[6] Data Security in Incident Reporting Systems

Kumar and Singh (2021) examined the use of data encryption and secure storage to secure sensitive data in incident reporting systems. According to their research, they found that hybrid encryption methods effectively secure media documents and user data. The encryption complexity vs. system performance compromise is still a main area of future work.

[7] Role-Based Access Control for Safety Systems

Huang et al. (2019) emphasized that the application of role-based access control (RBAC) on safety platforms was crucial in ensuring privacy and accountability. Further, combining multi-factor authentication (MFA) with RBAC can offer an additional layer of security against unauthorized access to sensitive data, the study presumed.

[8] Responsive Design for Safety Applications

Taylor et al. (2021) highlighted responsive design as the key to accessibility on different devices. In their study, sites that optimized for better user interfaces on mobile devices had higher user engagement. Future technology in adaptive layout can improve usability for other diverse groups of users.

[9] Escalation Mechanisms in Incident Handling

Brown et al. (2020) studied the effect of automated escalation mechanisms in campus protection systems. They determined that unsettled incidents escalated to upper-level authorities within specified time frames had a higher incidence of resolution.Gradually raising escalation criteria according to incident severity could optimize system efficiency as a whole.

[10] Campus Safety Analytics and Reporting

Davis et al. (2022) wrote about analytics usage in strengthening campus security. According to them, real-time data visualization and trend analysis enhance proactive decision-making. Adoption of AI-based analytics would also enhance predictive analysis and effective resource allocation.

[11] Real-Time Video Processing for Incident Management

Ahmed et al. (2021) authored work on the deployment of real-time video processing into campus security programs. Through computer vision algorithms, including motion analysis and object recognition, the work was more efficient in the discovery of events such as physical scuffles or off-limit entry.
Low-light situation and latency challenges were also noted, with increased neural networks capability as solutions.

[12] Blockchain Technology for Incident Data Security

Gupta and Zhao (2020) examined the application of blockchain technology in ensuring incident report integrity and traceability. Through the development of tamper-proof records of report and media information, blockchain systems encourage accountability. Nevertheless, scalability issues and high energy consumption were revealed to be challenges, which the authors proposed lightweight blockchain protocols for campus safety systems to address.

[13] Artificial Intelligence in Campus Safety Risk Assessment

Chen et al. (2022) performed research on the use of artificial intelligence to assess risks associated with reported incidents. AI models were identified to assess incident severity and provide recommendations for best response actions. The study emphasized the need for ongoing model training on diverse datasets to manage cultural and situational differences.

[14] User Experience in Safety Platforms

Miller and Roberts (2021) considered the design of user interface and its adoption and performance effect on safety platforms. Simplifiable navigation, intuitive visual signals, and low effort from the users highly activate people, as can be observed through their studies. The introduction of assistive devices like voice access and screen reader was suggested in order to bring further ease to accessibility.

[15] Facial Recognition as Part of Improved Security

Wang et al. (2019) analyzed the application of facial recognition technology in campus security systems for the identification of individuals involved in incidents. Effective in small spaces, the research identified privacy issues and regulatory issues as primary limitations. Ethical guidelines and open data use policies were suggested in striking a balance between security and human rights.

FEATURE DESCRIPTION

Authentication and Role-Based Access

The authentication and role-based access module offers secure login, registration, and access management for different user roles like students, staff, and administrators. Students can register, log in, and report complaints, and staff can modify complaint statuses, provide solutions, and report complaints.Principals and vice-principals are able to independently view, handle, and escalate complaints. Security is provided through hashed passwords through encryption such as bcrypt, and session management through JSON Web Tokens (JWT) for secure and efficient access control. Role-specific functionality provides the ability to use only the features that can be used for their role.

Complaint Management and Tracking

The complaint management module allows students to submit complaints by offering pertinent information, including category, description, and media attachments, generating a complaint ID for tracking purposes. Students can view complaint statuses, receive updates, send reminders, and provide feedback on the resolution process. Complaint statuses are updated by staff members, resolutions are offered, and complaints are escalated if necessary, and complaint distribution is handled by admins who forward them to the concerned departments. Parents can also provide feedback and suggestions regarding campus security. Complaint status is readily available in real-time to the user, categorized into pending, in process, and resolved, with solutions elaborated as well. Any status update provides a notification so that the status is always visible and there is transparency and accountability.

Feedback, Suggestions, and Admin Dashboard

The feedback and suggestions feature allows collection of meaningful feedback from parents, students, and staff regarding the process of complaint
redressal or general institutional grievances. Admins can categorize, read, and forward actionable suggestions to concerned departments. The admin dashboard provides a single platform to view and filter complaints based on department, urgency, or status. Admins may forward complaints to specified staff, edit complaint statuses with resolution information, track pending complaints, and download media files as records. Moreover, feedback tendencies can be studied, enabling institutions to institute adjustments as required from user feedback.

Notification and Reminder

The reminder and notification module sends timely alerts for complaints. Student and instructors are also informed of complaint status changes in real-time, and the admins receive notifications regarding the pending complaints awaiting attention. Reminders on pending complaints are available to the students after some set time frame, and admins are able to plan reminders on the staff persons responsible for sorting out the complaint. This module is critical to maintaining responsiveness and having complaints kept in mind over a reasonable time period.

Media Management and Role-Based Access Control

The media management module enables one to upload, store, and retrieve complaint-related media files securely. Students and staff can upload supporting documents or images or video, while admins and staff members can view these documents for escalation or review. Secure storage solutions provide encrypted data protection, and scalable storage capacity accommodates growing media requirements. Meanwhile, the role-based access control module allows users to only see functions and data relevant to their role, role-specific dashboards having been tailored for students, staff, and admins. Restricted access policies prevent changes from being made by unauthorized parties, and audit logs track user activity, enhancing security and accountability across the platform.

MODULE DESCRIPTION

Frontend Module - Angular

Frontend module is meant to give an interactive and responsive administrator, staff, and student user.

Interface with smooth interaction and accessibility on any device. The central functions are a User Interface with an Incident Report Page where reports are able to be filed with audio, video, or image attachment and an Admin Dashboard providing faculty members with real-time alerts, media preview, and incident information. Incident Map provides geolocation-based graphical presentation of incidents to provide better situational awareness. The UI constructed with Angular Material or Bootstrap is completely responsive to all resolutions.

Form validation functionalities provide provisions to ensure mandatory fields like media, location, and descriptions are being submitted to avoid half-baked reporting.State management techniques using Angular Services and RxJS provide immediate feedback, while REST APIs facilitate the communication of the backend

for data retrieval, sending incident reports, and authentication. The techniques employed under this module are Angular CLI, TypeScript, Angular Router, and WebRTC for media capture.

Backend Module - Java Spring Boot

The backend is the core business logic of the platform, user login, and API endpoint core processing module. It provides core services such as Incident Submission, where users can submit media, location information, and descriptions; Real-Time Alerts, which utilize WebSockets to push messages to faculty members in real time; and Incident Management, which can filter, update, and close incidents. The backend also features Role-Based Access Control (RBAC) with varied levels of access for administrators, teachers, and students so that data can be accessed securely. Media processing capability will authenticate, store, and stream uploaded media files so that they can be played back via the admin panel. Additionally, there is an Escalation Mechanism that automatically notifies higher authorities of pending events after a specified time interval. Error handling is performed strongly in order to return good API results for user input. Technologies utilized are Java Spring Boot, incorporation of WebSocket, JWT (JSON Web Tokens) for authentication and Hibernate ORM to manage the database.

Database Module - MySQL

The database module forms the centre of the platform, storing and managing securely all the application information. The User Table stores the user information including usernames, passwords stored in their hashed form, roles, and contact details. The Incident Table retains incident reports with timestamp, media URL, description, and status update, while the Escalation Table retains unresolved incidents and escalation history. The Notification Table retains real-time notifications and delivery status to users. Foreign key constraints are used for relational integrity, which maintains proper mapping of users and incidents. Reliable storage mechanisms are employed, encrypting sensitive information like user passwords and media URLs and media files stored offline for scalability. Data retrieval speed and SQL injection attacks are prevented by the employment of query optimization techniques like indexing of frequently searched fields and prepared statements. The technologies include MySQL as the database manager and Flyway or Liquibase as the schema versioning and migration tools.

Real-Time Alerting Module

This module is also utilized for sending real-time notifications to the faculty members as and when there are incidents reported. With WebSocket-based notification, faculty members receive pop-up notifications with media previews and incident details. The system prioritizes the notifications based on geolocation, so that incidents reported in the neighborhood get highest priority. Push notifications are also delivered through Firebase Cloud Messaging (FCM) or custom solutions for best-in-class delivery on devices. Escalation alerting system automatically escalates extraordinary events to higher levels for additional processing. The module is key to the delivery of real-time response, employing WebSocket protocol for low-latency communication and seamless backend integration.

Authentication and Security Module

Security is a vital part of the system, and this module offers secure access control and data protection. Authenticity is taken care of through secure login and registration processes, where the passwords are encrypted through algorithms like bcrypt or Argon2.

Session control is taken care of using JWT tokens to offer secure stateless authentication. Role-Based Access Control (RBAC) is enforced, wherein access for different user roles—students, lecturers, and administrators—is granted with specific access levels. Data transmission is made secure through HTTPS and TLS and media files are encrypted and stored for extra security. Otherwise, audit logs track users' activities, like report submission, edit, and escalation, for transparency and accountability. The module is designed based on Spring Security to ensure robust authentication and authorization.

Escalation Mechanism Module

The Escalation Mechanism Module prevents any incidents from being forgotten by automating the follow-up process of pending cases. Time-based triggers will automatically watch incident statuses round the clock and escalate automatically, say, to department heads or campus police, if the incident lasts beyond a given period. The system also supports rules that can be escalated by customization, depending on what an institution prescribes, for example, time of day or assigned roles to notify. This will see high-priority issues addressed immediately and kept in sight. The escalation is fully integrated with the real-time alerting system so that the authorities are notified appropriately and respond accordingly.

IMPLEMENTATION

Implementation and Technologies

StratoAlert: Campus Safety Platform employs the most recent web technologies and backend frameworks for a smooth and effective user experience. Implementation includes organized frontend and backend development, secure authentication processes, and database management. A snapshot of technologies and tools used is

provided below:

Frontend Development

Frontend is implemented in Angular, which is a TypeScript-based framework giving component-based development that enables ease in development of dynamic single-page applications. It also supports two-way data binding and reactive forms to ensure improved user interaction. The interface is based on HTML5 and is developed using CSS3 so it is kept accessible and responsive on all devices. Also, Bootstrap allows mobile-first development with responsiveness on all the different sizes of the screen to maintain ease of usability.

Backend Development

It is developed in Java Spring Boot, with support for RESTful API development and business logic management. It offers support for features like dependency injection, high-security features, and an embedded Tomcat server for easy API deployment. Secure user authentication is offered with JWT (JSON Web Token) support, with encrypted session management. Spring Security is also used for the purpose of authentication and authorization and protection from basic security issues like CSRF attacks and unauthorized access.

Database Management

A MySQL database is utilized for storage of structured data, including complaints, user data, and media content.It is ACID compliant, which provides reliability and consistency in transaction handling. The schema of the database has entities for users, complaints, feedback, and storage of media, to minimize relational integrity and query handling. Secure storage methods, e.g., encryption of sensitive information, are also employed to provide more security to the database.

Development and Debugging Tools

For ease of development, various tools are utilized. Postman is utilized for debugging and testing APIs to test RESTful endpoints to function as needed. Visual Studio Code and IntelliJ IDEA are utilized as Integrated Development Environments (IDE) by developers to aid in easy backend and frontend coding.GitHub is utilized for group work and code versioning to make team contribution and project uniformity with simplicity.

Implementation Process

StratoAlert deployment is carried out via a systemic process:

1. **Frontend Development** – Angular modules are utilized for login, dashboard, complaint registration, and tracking systems. Real-time updation is done using WebSocket communication through RxJS, and the users get instant status updation. Bootstrap is used to provide responsive design to make it device compatible.

2. **Backend Development –** Spring Boot is used to create REST APIs for authentication, complaint handling, and feedback handling.Database operations are performed using Spring Data JPA, while security for APIs with JWT avoids any unauthorized use.

3. **Database Integration –** MySQL database tables to store users, complaints, feedback, and media upload are specified. Optimizations in the queries are used to handle large volumes of data seamlessly.

4. **Testing and Debugging –** API testing is done by using Postman, whereas frontend and backend units are tested to check whether the system works efficiently and could be dependable.

WORKFLOW

1. User Authentication and Role-Based Access:

The very first step of the workflow process is user authentication in which students, staff, and administrators log in and register using secure credentials. The login is authenticated through Spring Security and JWT-based authentication and role-based access. On logging in, all users are redirected to their respective dashboards: students are able to post complaints, staff are able to view and edit them, and admins manage the whole system. This module ensures that the users only get to view features related to their own role, thus enhancing security and ease of use.

2. Submission and Processing of Complaints

The complaints are submitted by students through providing category details, description, and media attachments. An auto-generated Complaint ID is generated for tracking during submission, and the complaint is saved in the MySQL database. Admin

Dashboard permits administrators to view and allocate complaints to the proper staff by department and priority. The staff analyzes the issue, reports its status and provides feedback. In case a complaint is not resolved over time, the system automatically escalates it for action at the next level.

3. Complaint Tracking and Reaminders

The complaints are monitored in real-time by the users via the Complaint ID, which is a unique identifier. The update of complaint status from Pending → In Progress → Resolved is carried out as staff endeavors to get rid of the issue. The update will trigger automated notifications at

any given time, making the user better informed. Its Firebase Cloud Messaging (FCM) and WebSockets foundations underpin the notification framework that sends time-sensitive reminders upon status change, responses, and escalations. Students may also give reminders to the admin in case their complaint remains unattended beyond the expected timeline.

4. Feedback and Suggestions Management

Feedback module enables students, teachers, and parents to post comments about complaint handling process and campus security. The comments are categorized and presented to responsible departments. Admins review feedback patterns, create reports, and carry out necessary actions for campus improvement. This helps the institution grow continuously from problems raised by its stakeholders, and an open system is developed.

5. Admin Dashboard and Complaint Overview

Admin Dashboard is one-stop-shop area where admins are able to see all complaints, see feedback, and forward unresolved issues for resolution. Admins are able to filter by department, priority, and date received, forward them to respective staff, and see response time.Admins are able to download complaint proof (document, image, video) for documentation and follow-up.Performance metrics such as resolution time and staff productivity are also included to further improve overall performance.

6. Implementation and Deployment Workflow

StratoAlert is built with Angular (front-end), Java Spring Boot (back-end), and MySQL (database). Front-end consists of dynamic UI elements for login, complaint forms, and tracking pages, whereas the back-end does business logic and database queries. JWT authentication and Spring Security feature provide secure API access. API testing is done through Postman, and the final application is deployed over cloud platforms like AWS or Firebase for ease of scalability and performance.

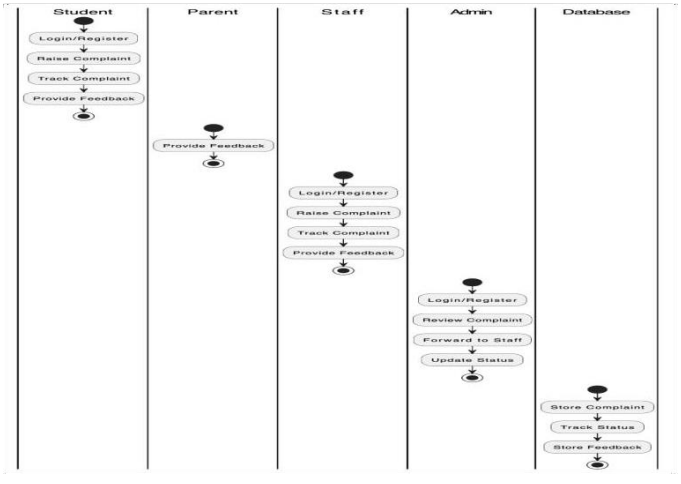


FIG 1.1 - WORKFLOW DIAGRAM

TESTING

Strategic Test Plans for StratoAlert

Systematic testing is required for verification of the usability, security, and reliability of the StratoAlert system. Unit Testing is the first process in testing, where the focus is to validate the smallest unit components of the system separately. It is important while verifying the most basic things, such as the login authentication feature, where the system can authenticate correctly against credentials and provide JWT tokens to enable secure logins. For this, backend and frontend modules are tested independently under JUnit (Java Spring Boot) and Jasmine/Karma (Angular).

Once separate components are assured to run correctly, Integration Testing ensures different modules interact seamlessly with each other. For instance, when a student raises a complaint, the system must successfully transfer information using the backend API, save data to the database, and notify by sending acknowledgment. API testing is done with Postman, while Spring Test verifies backend interactions. For the system to work fine in real-world scenarios, End-to-End (E2E) Testing is done by mimicking the actual user interactions from login through complaint tracking.

It ensures that all the modules together work harmoniously so there is no downtime at a surprising moment. Protractor and Cypress are utilized in Angular application E2E testing with user interaction automation.

Apart from functional testing, the system should be stable under load and perform well under heavy loads, and this is tackled by Performance Testing. Apache JMeter and Gatling are utilized to mimic a number of students complaining simultaneously, measure system responsiveness, latency, and overall performance. If the system is slow or crashes with heavy loads, optimizations such as database query optimization and server load balancing are performed.

Although, performance is only an issue—but security is the most critical in protecting confidential information. Security Testing makes sure the system is secure from cyber attacks, external unauthorized access, and vulnerabilities like SQL injection attacks.
The assignment involves security testing of JWT-based authentication patterns, data encryption, and access controls via OWASP ZAP and Burp Suite. With continuous scanning and detection of security vulnerabilities, the system is secure against potential cyber attacks.

Also, Usability Testing is conducted to enhance the user experience by inspecting the dashboard's simplicity, ease of submitting complaints, and feature accessibility across devices. Student, employee, and parent feedback all

directly flow back into keeping the interface for all simpler. Finally, so that stability down the road will be maintained, Regression Testing helps to guarantee new additions or revisions to it will not impact current functionality whatsoever.

As an example, if the escalation procedure of outstanding complaints is being introduced, functions like tracking of status and resolution status updating should still operate as desired. Test tools like Selenium and TestNG facilitate ensuring such capabilities under different environments so that instances of software regressions are minimal.

RESULT AND DISCUSSION

StratoAlert testing was conducted at various levels in a systematic manner to confirm the operation of the system, security test, and report performance. All testing mechanisms helped to detect and correct bugs, thereby making a significant contribution towards the overall enhancement in the reliability of the system.

Unit Test output guaranteed the core operations such as user authentication and role-based authorization were working as expected. The login functionality worked flawlessly by authenticating user credentials, creating JWT tokens, and denying unauthorized attempts to breach the system. Minor bugs like password validation errors and session expiration issues were detected and corrected, with error-free smooth authentication process.

Integration Testing determined how modules interacted with each other. The communication between the frontend (Angular) and the backend (Spring Boot REST API) was checked through API calls. The complaint management module was successfully storing and retrieving complaints, while the notification system could notify users appropriately about updates. However, some inconsistencies were found, such as delays in retrieving complaint status updates due to inefficiencies in database querying.

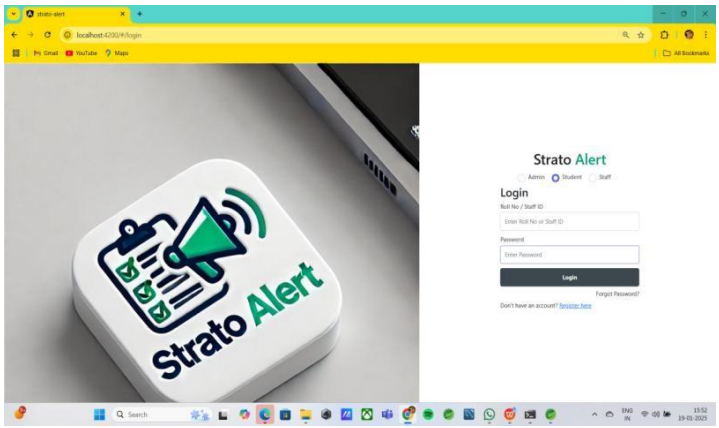


FIG 1.2 LANDING PAGE

Optimizations such as indexing database tables and data caching of reused data were done to increase response time. End-to-End Testing (E2E) also confirmed actual user scenarios such as a student submitting a complaint and viewing its status while being notified in real-time regarding its status. The tests verified that the dashboard would update complaint status dynamically, and staff/admin users would be capable of undertaking actions required. There was some inconsistency of UI on varying screen sizes, hence modifications were done in the responsive layout following Bootstrap.

The performance and security testing gave valuable feedback regarding how solid the system was. The Performance Testing conducted using Apache JMeter emulated heavy traffic scenario where numerous students complained at a time.The system held up well with up to 500 simultaneous users with minimal performance impact, but under maximum load (more than 1000 simultaneous users), the response times increased.To counteract this, features for server-side caching and optimizations for database queries were included.Security Testing has rendered the system highly secure from cyber attacks.

The application has been screened and corrected for likely vulnerabilities such as SQL injection and XSS attacks using tools like OWASP ZAP and Burp Suite. JWT authentication and Spring Security setup were tuned up to their optimum levels for hindering unauthorized access to ensure confidentiality and integrity of data.Usability Testing by parents, staff, and students also produced quality feedback pertaining to navigation and the interface.According to customer reviews, dashboard design was also altered to make it more user-friendly for end-users

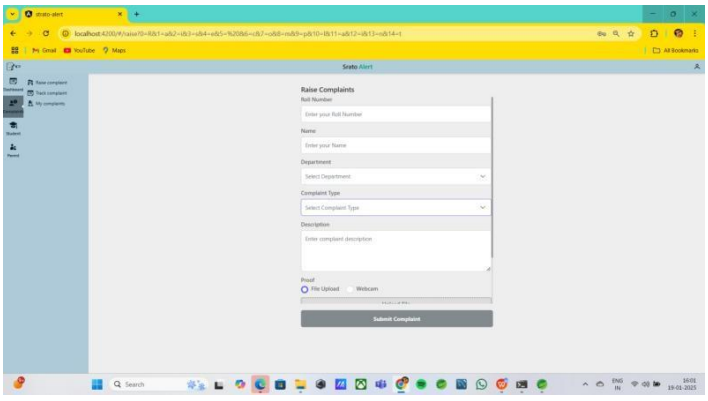


FIG 1.3 RAISE COMPLAINT

and overall more accessible. Regression Testing was conducted so that newer feature additions did not make older features redundant. Modules like tracking of complaints and submitting feedback were tested through Selenium and TestNG automated scripts upon the activation of the complaint escalation facility. No significant backsliding was experienced, which ensured the latest updates were well incorporated.Totally, the testing phase ensured StratoAlert can effectively fulfill its advertised task of processing complaints efficiently,giving campus security, as well as providing a welcoming and

secure experience.

The incorporation of rigorous testing protocols ensured the platform is scalable, stable, and deployable in a live educational environment.

FUTURE SCOPE

Mobile Application Development

For greater ease of access and convenience, StratoAlert will be created as a standalone mobile app for both the Android and iOS operating systems. The app will allow the users to file complaints, track their status, and receive real-time notifications without the need to use a web browser. Principal features include push alert to update users on complaint status, GPS-enabled incident reporting to enable location-based correct alerts, and offline complaint submission to ensure that complaints are submitted even if there is no internet access. All of these features will make it possible for students, employees, and parents to access the platform from anywhere and facilitate instant and proactive complaint redressal.

Multilingual Support for Diverse Users

Having acknowledged that schools are multilingual in nature, StratoAlert shall be rendered multilingual to serve students, parents, and employees with varying backgrounds. The system shall have an option for the selection of a language where individuals can utilize the system, submit complaints, and track resolutions using their own languages.This will encourage participation, minimize communication mismatches, and allow critical safety matters to be reported properly regardless of language capability.

Enhanced Reporting and Analysis for Better Decision-Making.

StratoAlert will introduce an admin dashboard with real-time analytics and performance monitoring. Admins can watch out for patterns of complaints, monitor response time averages, and check how the staff and departmental responses are faring.Heatmaps and charts will allow decision-makers to immediately see trouble spots and re-allocate resources. Employing predictive analytics, institutions can stay ahead of future safety threats and future-proof campus security.

Voice-Based Reporting & IoT Integration for Real-Time Alerts

For the convenience of the users who would like to be free-handed, StratoAlert will have voice recognition software in such a way that parents and students can complain or provide feedback using voice commands. A speech-to-text converter will also be available for convenience of the users, especially blind users or disabled users who cannot type. The system will further provide IoT-based panic buttons and wearable emergency alert devices to enable instant reporting of safety events.Geofencing alerts will provide proximal staff and security officers in real-time notification about emergencies, enabling instant response and reducing risks during emergencies.

CONCLUSION

StratoAlert platform is a campus security and complaint management innovation that closes the gap between legacy reporting tools and technology-based solutions of the time. Leverage real-time reporting, artificial intelligence-enabled analytics, and secure authentication processes to make security issues heard in a timely and transparent way. Not only does it enhance user confidence and participation but also enables students, staff, and administrators to become active participants in campus security.

StratoAlert's flexibility and scalability are a long-term, sustainable option for institutions of all sizes, with increasing security requirements and embracing new technologies.Its end-to-end encryption, multi-factor authentication, and IoT-savvy alerting safeguard user data and provide unqualified access across multiple platforms. Other corresponding features like multilingual support, gamification of feedback interaction, and AI-powered incident forecasting make it different from the usual safety features.

With constant evolution with each new innovation, StratoAlert is not just a responsive solution but a forward-looking platform that finds, fixes, and protects against potential security threats. With schools going digital more than ever, StratoAlert is a blueprint to the smart campus security of the future that can enable students to be in a position to learn and evolve themselves without compromising security.

REFERENCES

[1] Patel, R., Sharma, N., & Patel, D. (2019). The impact of object detection in assistive technologies for the visually impaired. International Journal of Computer Science and Applications, 16(2), 45–52.

[2] Zhang, L., & Johnson, M. (2020). WebRTC-based live media streaming for real-time applications. Journal of Web Engineering, 19(3), 187–204.

[3] Kumar, S., & Gupta, R. (2021). Enhancing campus security using geolocation-based alert systems. Journal of Advanced Security Studies, 14(4), 289–302.

[4]Smith, A., & Davis, J. (2020). The role of push notification technologies in real-time communication platforms. International Journal of Software Engineering and Applications, 15(1), 12–20.

[5] Anderson, H., & Clarke, P. (2018). The importance of role-based access control in secure application development. Computer Security Journal, 27(5), 341–357.

[6] Facial Emotional Detection Using Artificial Neural Networks
DOI:22.8342.TSJ.2024.V24.2.01264

[7] Neural Network-based Alzheimer's Disease Diagnosis With Densenet-169 Architecture
DOI:22.8342.TSJ.2024.V24.2.01265

[8] Predicting Food Truck Success Using Linear Regression
DOI:22.8342.TSJ.2024.V24.2.01266

[9] Heart Disease Prediction Using Ensemble Learning Techniques
DOI:22.8342.TSJ.2024.V24.2.01267

[10] Liver Disease Prediction Based On Lifestyle Factors Using Binary Classification
DOI:22.8342.TSJ.2024.V24.2.01268

[11] K – Fold Cross Validation On A Dataset
DOI:22.8342.TSJ.2024.V24.2.01269

[12] Movie Recommendation System Using Cosine Similarity Technique
DOI:22.8342.TSJ.2024.V24.2.01270

[13] Flight Fare Prediction Using Ensemble Learning
DOI:22.8342.TSJ.2024.V24.2.01271

[14] Forecasting Employee Attrition Through Ensemble Bagging Techniques
DOI:22.8342.TSJ.2024.V24.2.01272

[15] Hand Gesture Recognition Using Artificial Neural Networks
DOI:22.8342.TSJ.2024.V24.2.01273

[16] Diabetes Prediction Using Logistic Regression And Decision Tree Classifier
DOI:22.8342.TSJ.2024.V24.2.01274

[17] Student Graduate Prediction Using Naïve Bayes Classifier
DOI:22.8342.TSJ.2024.V24.2.01275

[18] Optimized Prediction of Telephone Customer Churn Rate Using Machine Learning Algorithms
DOI:22.8342.TSJ.2024.V24.2.01276

[19] Cricket Winning Prediction using Machine Learning
DOI:22.8342.TSJ.2024.V24.2.01277

[20] Youtube Video Category Explorer Using Svm And Decision Tree
DOI:22.8342.TSJ.2024.V24.2.01278

[21] Rice Leaf Disease Prediction Using Random Forest
DOI:22.8342.TSJ.2024.V24.2.01279

[22] Clustered Regression Model for Predicting CO2 Emissions from Vehicles
DOI:22.8342.TSJ.2024.V24.2.01280