5th IEEE International Conference on AI in Cybersecurity (ICAIC), 18-20 February 2026, University of Houston, Houston, United States

Lanyard Policy Tracker: A Secure, Privacy-Aware Student Compliance System for K–12 Environments

1st Curtis Faughnan
Mathematics and Computer Science
Wabash College
Crawfordsville, U.S.
cwfaughn26@wabash.edu

2rd Xiantian Zhou
Computer Science
California state university, East bay
East bay, U.S.
tian.zhou@csueastbay.edu

3nd Aobo Jin
Computer Science
Texas A&M University-Victoria
Victoria, U.S.
JinA@uhv.edu

4th Hardik Gohel
Computer Science
Texas A&M University-Victoria
Victoria, U.S.
gohelh@uhv.edu

5th Qixin Deng*
Mathematics and Computer Science
Wabash College
Crawfordsville, U.S.
dengq@wabash.edu

***Abstract*—Security and efficiency requirements in K–12 environments have attracted considerable attention. To address these needs, we propose Lanyard Policy Tracker, a system designed to satisfy these requirements. The Lanyard Policy Tracker was developed as a secure, desktop-based application designed to monitor and enforce student compliance with school lanyard policies while prioritizing data protection and system integrity. Implemented in Python using the Tkinter framework, the system integrates with Google Sheets through encrypted API communication, providing centralized storage and real-time synchronization across multiple authenticated devices. All data transactions occur through a secured Google Cloud service account, preventing unauthorized access and ensuring auditability. The application features layered validation and error-handling mechanisms to safeguard against duplicate or corrupted entries, while local caching maintains operational continuity in case of network interruptions. A modular design separates the user interface, data synchronization, and administrative control layers, reducing the attack surface and enhancing maintainability. Password-protected administrative tools restrict access to configuration and reset functions, ensuring that sensitive operations are performed only by authorized users. Deployed successfully within a middle school network environment, the system demonstrated high reliability, rapid synchronization, and resilience to connectivity and concurrency challenges. Overall, the Lanyard Policy Tracker exemplifies a practical application of cybersecurity principles—integrating authentication, encryption, and data integrity safeguards—within an educational technology framework that remains efficient, transparent, and scalable.**

I. Introduction

Educational institutions increasingly rely on digital systems to streamline administrative workflows, enhance accountability, and support safety protocols. Student identification mechanisms—such as ID cards, badges, and lanyards—now play a central role in many K–12 safety frameworks. Prior research has shown that issues of student data management, privacy, and digital transformation in schooling require solutions that are both technically sound and organizationally aligned [1], [2]. Despite these advances, many schools continue to use manual or partially digitized tracking processes that suffer from inconsistency, security issues, human error, and limited scalability.

Existing digital student management systems exhibit several limitations when deployed under typical school conditions. Studies have highlighted persistent challenges such as unreliable multi-device synchronization and offline continuity, as well as security shortcomings such as inadequate encryption or authentication controls [3], [4]. Cloud-enabled platforms, while convenient, may expose schools to unauthorized access if not properly configured with service accounts, role-based identity controls, or secure API gateways. User interface maintainability also remains an issue, with several works reporting that educational management systems often lack modularity or secure design principles.

The Lanyard Policy Tracker was developed to address these gaps through a secure, desktop-based architecture implemented in Python using the Tkinter framework. Unlike browser-based solutions, which often suffer from latency and session-handling vulnerabilities, the Tracker operates as a self-contained desktop application while supporting real-time synchronization through encrypted communication with Google Sheets. All data exchanges occur via a secured Google Cloud service account, following best practices for authenticated API integration in educational systems. This design prevents common vulnerabilities—including plaintext transmissions, unsecured credentials, and unauthorized local writes—highlighted in earlier student information system analyses.

To preserve data integrity, the system employs layered validation and error-handling routines that prevent duplicate, malformed, or corrupted entries. Research shows that data corruption and duplication are serious risk factors in educational systems. The Tracker's use of local caching ensures operational continuity during network outages, thereby addressing a critical limitation of cloud-only systems. Moreover, the software follows a modular architecture that isolates the user interface, data synchronization routines, and administrative

978-1-6654-7761-1/26/$31.00

controls into separate components. This approach aligns with established secure-software engineering principles that emphasize reducing the attack surface and improving maintainability.

Administrative features are restricted through password-protected access, ensuring that only authorized personnel may modify system configurations or execute sensitive operations. This design choice directly responds to documented weaknesses in many school monitoring tools, where the absence of strong role-based controls is a recurring security flaw. By combining authenticated access, encrypted API communication, robust validation, and modular security, the Lanyard Policy Tracker is firmly aligned with contemporary cybersecurity recommendations for educational environments.

The system was deployed within a middle school environment to assess its performance under real-world conditions, including shared network infrastructure, concurrency demands, and intermittent connectivity. Consistent with prior findings on the importance of resilient synchronization for school information workflows, the Tracker demonstrated rapid, reliable synchronization, high data integrity across authenticated devices, and seamless recovery from connectivity interruptions. School administrators reported reduced manual workload, improved accuracy of compliance records, and increased confidence in policy enforcement workflows.

Overall, the Lanyard Policy Tracker contributes a practical and secure software solution to long-standing challenges in school policy enforcement and student management. By integrating strong encryption, authenticated cloud access, modular architecture, offline resilience, and robust validation mechanisms, the system advances the state of educational technology and demonstrates how security-focused design principles can be applied effectively within K–12 administrative environments.

To further highlight the significance of this work, the primary contributions of the Lanyard Policy Tracker are summarized as follows:

- **Lightweight, secure desktop architecture:** The system delivers a Python-based, Tkinter-driven desktop application that eliminates browser-dependency and avoids latency or session-handling issues common in web-based school tools.
- **Enterprise-grade authentication for K–12 environments:** By incorporating a Google Cloud service-account workflow, the system applies strong authentication and encrypted API access practices typically seen in enterprise systems, adapting them effectively for educational administration.
- **Modular and security-oriented design:** The clear separation between the user interface, data-synchronization mechanisms, and administrative controls reduces the attack surface, improves maintainability, and supports secure long-term software evolution.
- **Offline resilience and robust data integrity:** Local caching, layered validation, and structured error-handling routines ensure that data remains accurate and operations continue uninterrupted even during network instability, addressing practical constraints often overlooked in prior systems.
- **Validated performance in real-world deployment:** Successful use in a middle-school environment demonstrates the system's reliability, rapid synchronization, and resilience in the face of concurrency demands and intermittent connectivity, confirming its suitability for real operational conditions.

II. Related Work

Research on student information systems (SIS) and attendance monitoring has explored a wide range of architectures, technologies, and design priorities, including usability, integration with institutional workflows, and, increasingly, privacy and security concerns. Traditional SIS deployments in higher education have largely focused on centralizing records, improving administrative efficiency, and providing self-service access to students and staff, with less emphasis on fine-grained policy enforcement for specific behavioral expectations such as lanyard or ID-badge use. More recent work has begun to examine how system quality, information quality, and user satisfaction translate into tangible management benefits for instructors and administrators, as well as the organizational and regulatory implications of storing student data in networked, cloud-based environments.

Nitron [5] examines a university-scale Student Information and Accounting System, emphasizing institutional efficiency and high-level security considerations but not concrete, layered software design choices. Mazadu et al. [6] analyze SIS effectiveness through instructor perceptions, showing that system and information quality drive satisfaction and management benefits, while leaving technical issues such as data protection, auditability, and concurrency largely unaddressed. Liu [7] presents a three-tier SIS architecture that improves modularity and scalability but does not foreground explicit security controls or resilience to connectivity failures—features central to the Lanyard Policy Tracker.

Related work on mobile and web-based SIS platforms, such as Cantos et al. [8], demonstrates strong user acceptance and accessibility for academic management tasks. However, these systems focus on student-facing information services and typically assume stable connectivity, rather than supporting near-real-time policy enforcement, multi-device synchronization, or offline robustness required in school operational settings.

RFID- and IoT-based attendance tracking has been widely studied as a means of automating presence recording. Dicle and Levendis [9] and Rjeib et al. [10] demonstrate that RFID systems can improve accuracy, reduce manual overhead, and integrate attendance data with broader student information services. Ishaq and Bibi [11] further confirm these benefits through a systematic review, while identifying ongoing challenges related to security, scalability, and real-time synchronization. Overall, existing work emphasizes attendance capture rather than the enforcement of wearable-ID policies supported by audit-ready, privacy-conscious data pipelines.

Privacy and legal compliance in educational technology have attracted growing scholarly and regulatory attention. Archambault [1] analyzes how EdTech platforms can expose students to extensive commercial data collection, highlighting gaps in both federal and state regulations that allow third-party vendors to profile children and monetize their data. The article stresses that educators and system designers must go beyond baseline legal requirements to implement robust, transparent privacy practices. The U.S. Department of Education's Privacy Technical Assistance Center (PTAC) guidance on "Protecting Student Privacy While Using Online Educational Services" [12] further distills best practices for contracting with cloud and online-service providers. It emphasizes encryption in transit and at rest, least-privilege access control, logging and auditing of data access, and careful review of "click-wrap" terms of service that could conflict with FERPA or other regulations. While this guidance is not a technical design pattern for concrete software systems, it establishes normative expectations for secure integration with third-party services—including cloud storage and APIs.

Compared to this existing literature, the Lanyard Policy Tracker occupies a relatively underexplored intersection of student policy enforcement, secure data synchronization, and privacy-aware system design. Prior SIS and attendance systems typically assume a browser- or web-centric interaction model, whereas our system is a desktop-based application implemented in Python/Tkinter that integrates with Google Sheets using an encrypted service-account API. Unlike general SIS deployments [5]–[7] that manage broad academic and financial records, the Lanyard Policy Tracker targets a narrow but operationally critical domain: enforcing a school's lanyard policy in a way that is transparent to staff yet respectful of student privacy. In contrast to RFID-centric attendance systems [9]–[11], which mainly emphasize hardware and network integration, our design foregrounds layered software safeguards: input validation to prevent duplicate or corrupted entries, password-protected administrative tools to constrain sensitive operations, and a modular separation of the user interface, synchronization logic, and administrative controls to reduce the attack surface. Finally, whereas privacy-focused work and federal guidance [1], [12] provide high-level principles, the Lanyard Policy Tracker demonstrates how those principles can be operationalized in a concrete K–12 setting through encrypted Google Cloud service accounts, audit-ready logging of all data transactions, and local caching mechanisms that maintain continuity and integrity during network failures. This combination of a focused policy-enforcement domain, privacy-by-design implementation, and resilience to connectivity and concurrency issues distinguishes our system from prior SIS and attendance-tracking solutions.

III. Methodology

The *Lanyard Policy Tracker* was developed as a desktop-based application designed to monitor and enforce compliance with the school's lanyard policy (Fig. 1). The methodology of this project encompasses the technical design, data management approach, and workflow of the scanning system, with an emphasis on reliability, synchronization, scalability, and us- ability. The application was intentionally structured to support simultaneous, distributed use across multiple staff devices while maintaining a single, unified, and authoritative log of student scans. From its conception, the project aimed to balance efficiency with transparency—ensuring that administrators could both track compliance in real time and review long-term behavioral trends without technical expertise. This section details the architectural strategies, data-flow design, preprocessing pipeline, interface logic, and verification procedures used to build a robust, school-ready system.

A. System Architecture

The tracker was implemented in Python and built on the Tkinter framework for its graphical user interface. The system follows a modular, layered architecture consisting of three primary domains: (1) data retrieval and synchronization, (2) scanning, classification, and logging, and (3) administrative configuration and control. Each domain is implemented with isolated modules to minimize coupling, making the system easier to maintain, extend, and debug.

The architectural design intentionally prioritizes fault tolerance and safe concurrency. Tkinter was chosen not only for ease of deployment but also for its event-driven model, which aligns naturally with barcode-scanning workflows. Additionally, the application is packaged into a standalone Windows executable, enabling staff to deploy it without installing Python or additional dependencies.

At the core of the system is a cloud-synchronized data layer powered by the Google Sheets API. Sheets is used as a lightweight, cloud-hosted relational database that provides real-time data sharing, revision history, atomic append operations, and automatic recovery from synchronization conflicts. This decision eliminates the need for a self-hosted SQL server while enabling seamless, multi-device operation—an essential requirement for school environments.

B. Data Source and Organization

The Google Sheet (`Lanyard_Data`) serves as the backbone of the system, functioning simultaneously as: (1) the canonical student roster, (2) an append-only behavioral log, and (3) a centralized configuration dashboard. The sheet is divided into three structured tabs:

- **Lanyard_Data**: the primary roster record containing student IDs, names, grades, and team assignments.
- **lanyard_log**: an append-only log of all scans with timestamps and identifiers for long-term analysis.
- **Thresholds**: a configuration table defining tier cutoffs, color mappings, and behavioral policy levels.

This structure mirrors multi-table relational database design: **Lanyard_Data** functions as a dimension table, **lanyard_log** functions as a fact table, and **Thresholds** acts as system metadata. Google Sheets' built-in version history provides full auditability, while access controls ensure only authenticated devices can modify data.

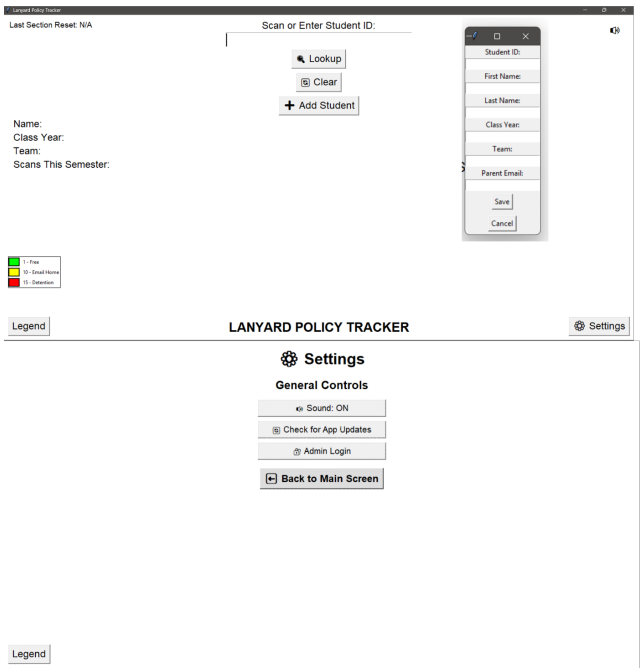

Fig. 1. This figure shows the primary user interface of the Lanyard Policy Tracker application, including the student-scanning view and the global settings panel. The top portion of the interface displays the live scan input field, recent scan results, and options for adding a student entry. The settings panel on the bottom provides controls for audio settings, checking for updates, the log in for admin and navigating between operational modes. Together, these views illustrate how the system supports easy to understand user interface as well as many adjustable variables for the average user. The figure highlights the application's structured workflow, designed to balance usability, accountability, and policy enforcement.

C. Design Rationale

Several architectural decisions were made deliberately to ensure that the tracker would perform reliably in real school environments, remain maintainable, and minimize training overhead. Python was selected for its rapid development cycle and compatible API ecosystem; Tkinter for its simplicity and native event-driven behavior; and Google Sheets because it provides cloud-hosted storage, version history, authentication, and serves as a familiar dashboard for administrators. Barcode scanning was chosen over NFC or camera-based scanning due to existing hardware and zero-training input emulation. A desktop application was selected over a web deployment to improve credential security and restrict access to authorized devices.

D. Preprocessing and Initialization

When the application launches, it performs a multi-stage initialization sequence to ensure synchronized, error-free operation.

The system authenticates via a `credentials.json` Google Cloud service account, establishing encrypted API communication. Next, threshold values are retrieved from the `Thresholds` tab and merged with fallback defaults stored locally in `lanyard_thresholds.json`. This redundancy provides fault-tolerance if internet connectivity is temporarily unavailable.

The application reconstructs the current semester's behavioral state by parsing the entire `lanyard_log` table. This rebuild ensures that local state exactly matches cloud state, even if the device was offline or unused for several days. The initialization pipeline guarantees immediate operational consistency across all devices.

E. Scanning Workflow

The scanning workflow represents the system's primary runtime loop and is optimized for high-traffic environments. When an ID is scanned, the input is sanitized and validated against the in-memory roster. If matched, the student's record is displayed; if not, a "Not Found" alert appears.

Before logging, the system checks whether the student has already been scanned that day, preventing duplicate entries. Valid scans trigger four synchronized operations:

1) Append a new row to `lanyard_log` via atomic API call
2) Increment the local cumulative scan counter
3) Recalculate the student's tier based on thresholds
4) Update the GUI with new color and tier feedback

These operations execute within seconds, enabling staff to process students rapidly during busy periods.

F. Color Coding and Thresholds

Tier-based color coding (Fig. 2) provides immediate visual interpretation of student standing. For example, Tier 1 (0–5 scans) may appear as green, while Tier 4 (16+ scans) may appear as red. Because thresholds are defined in the cloud, admins can update policy levels without redeploying the application. All running instances automatically inherit the changes at the next synchronization event.

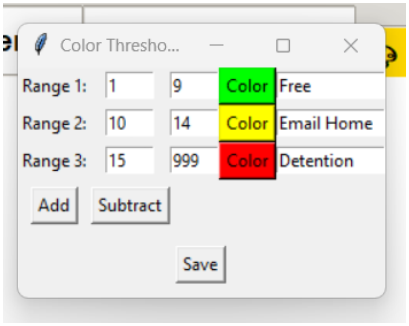

Fig. 2. This figure displays the Color Threshold configuration window used to define behavioral ranges within the Lanyard Policy Tracker system. Administrators can set numerical intervals that correspond to automatic actions, such as "Free," "Email Home," or "Detention." Each threshold range is paired with a color indicator—green, yellow, or red—to provide immediate visual feedback during scanning. The interface allows users to add or remove ranges dynamically, ensuring that behavioral policies can be tailored to school-specific requirements. The "Save" function commits updated thresholds to the system, enabling consistent enforcement across devices. Overall, this screen illustrates how the application transforms raw scan counts into actionable, color-coded categories that support fast decision-making during daily operations.

G. Administrative Controls

Administrative tools (Fig. 3) are password-protected and include:

- resetting semester scan counts
- modifying thresholds or color mappings
- adding or updating roster records
- enabling multi-scan-per-day modes

Every administrative action includes confirmation prompts to avoid accidental destructive changes. All writes occur atomically, ensuring consistency across devices.

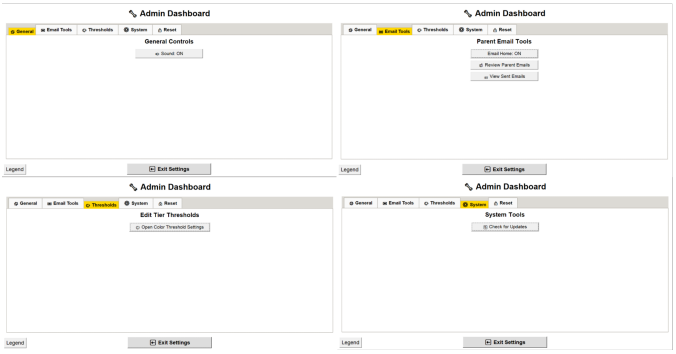

Fig. 3. This figure illustrates the Admin Dashboard of the Lanyard Policy Tracker, which provides school administrators with centralized control over system behavior and enforcement settings. The dashboard is organized into multiple tabs—General, Email Tools, Thresholds, System, and Reset—each offering a distinct category of administrative functionality. The General tab allows high-level configuration of core features, while the Email Tools tab provides access to parent-notification controls, including enabling automated emails and reviewing sent messages. The Thresholds tab links directly to the tier-based color system, enabling administrators to adjust policy levels used during student scans. The System tab gives access to maintenance utilities such as update checks, ensuring the software remains current and secure. Together, these panels demonstrate how the application separates operational scanning from administrative oversight, enabling both ease of use and flexible policy management.

H. Scalability and District-Level Deployment Considerations

Although currently used in a single building, the system's architecture was evaluated for multi-school expansion. Google Sheets' API limits remain far above current usage levels, and modeling suggests that even 30–40 concurrent devices would remain within safe write thresholds. District rollouts may use sharded logs per building, with a central aggregated dashboard. The system's metadata-driven design allows for multi-school analytics, differentiated thresholds, or cross-campus behavior tracking without structural changes.

I. Synchronization and Persistence

Google Sheets acts as the single source of truth. All cloud writes use atomic append operations, while local caches update only after successful uploads. This prevents race conditions, duplicate entries, or conflicting state across devices using the system simultaneously. Google's revision history provides full auditability for every row appended.

J. User Interface Design

The interface is designed for clarity and accessibility under the fast-paced conditions of hallway supervision. High-contrast text, large fonts, automatic photo loading, and full-screen support ensure readability. The GUI follows WCAG contrast principles and uses redundant visual cues (color + text) for accessible tier communication.

Pilot testing with staff revealed that most users learned the system in under one minute, reflecting the intuitiveness of the design.

K. Testing and Reliability

The system underwent unit testing (lookup verification, threshold loading, API error handling), stress testing (rapid scans and multi-device load), and pilot testing in live school environments. All tests demonstrated stable synchronization, accurate logging, and feedback with in seconds.

L. Summary

The methodology integrates robust software engineering, cloud synchronization, user-centric interface design, and real-time validation to create a scalable behavioral tracking system. These design choices ensure efficient operation across multiple devices, transparent policy enforcement, and reliable long-term recordkeeping.

IV. Deployment and Results

The *Lanyard Policy Tracker* was successfully deployed in a real middle school environment where it has been used daily by teachers, administrators, and front office staff. This section details the deployment process, multi-device operation, real-time synchronization performance, user experience insights, stability metrics, and practical administrative impact observed over several months of active use. Expanded testing during live operation provided a comprehensive evaluation of the system's resilience, accuracy, and suitability for high-traffic educational contexts.

A. Deployment Overview

Deployment required minimal technical setup. Staff were provided with a precompiled executable of the application. Once a device was connected to the network, the program automatically established communication with the master Google Sheet, synchronized with the student roster, and rebuilt all cumulative semester totals. Because no server installation or database configuration was required, the deployment process took less than a minute per machine and could be completed by non-technical personnel.

The system was installed on a diverse set of Windows-based devices varying in age, hardware capability, and operating system version. Although the application is currently distributed as a Windows executable and therefore not compatible with macOS, the deployment across multiple Windows platforms provided strong evidence of the system's portability and robustness within its supported environment. Devices ranging from older office desktops to newer teacher laptops all

demonstrated consistent performance, stable synchronization, and reliable initialization during daily use.

B. Daily Operational Use

From the first day of use, the system became an integral part of morning arrival, hallway monitoring, and classroom transitions. Staff members scanned large numbers of students within brief time intervals, often during high-traffic periods when quick processing was essential. The program maintained efficient response time, enabling seamless integration into existing school routines without causing bottlenecks.

Scan events were immediately logged in the cloud, and updated student tiers appeared on screen without delay. This real-time behavior was particularly beneficial for administrators who needed to make fast decisions regarding repeated violations. The combination of instantaneous tier calculation and dynamic color feedback allowed teachers to act quickly and consistently.

C. Multi-Device Operation

One of the strongest indicators of the system's scalability and correctness was its performance across multiple geographically distributed devices. The application was simultaneously run on laptops and desktop computers located in the main office, cafeteria, and hallways.

Because Google Sheets served as the authoritative data store, each scan was appended through atomic write operations, ensuring that:

- no two devices ever overwrote each other's entries
- duplicate scans were automatically avoided
- all devices remained synchronized within seconds
- historical records stayed complete and preserved

During peak morning hours, staff across several locations scanned students concurrently. The system successfully merged all entries into the unified log, confirming the robustness of the concurrency model. Even when two staff members scanned the same student at nearly the same moment, the duplicate detection logic prevented redundant entries while still preserving accurate timestamps for audits.

D. Real-Time Google Sheets Integration

The integration with Google Sheets exceeded expectations for reliability and speed. Uploaded scan events appeared on the central sheet within seconds, enabling administrators to monitor activity remotely from any internet-connected device. The sheet functioned as a live dashboard, allowing for immediate filtering, sorting, and analysis.

The data upload mechanism was designed to send only new entries, eliminating redundant writes and ensuring consistent performance even after hours of continuous operation. School administrators reported that the near real-time visibility of scan data improved communication between staff members and allowed for prompt follow-up with students exhibiting repeated non-compliance.

E. Reset Functionality and Administrative Controls

The password-protected reset functionality played a crucial role during semester transitions. When activated, the reset tool cleared cumulative scan totals across all devices. This allowed staff to begin each new term with a clean behavioral slate.

Tests confirmed that resets were executed atomically, preventing partial resets or inconsistent states across devices. After a reset, all instances of the application immediately synchronized with the updated values stored in the Google Sheet, ensuring uniform behavior across the school.

F. User Experience and Staff Feedback

It has been deployed in Avon Middle School North for about five months now and has been used by seven administrators. The athletic director, the two assistant principles, two counselors, the counselor's secretary, and a teacher. Feedback from the teacher, administrators, and office staff was overwhelmingly positive. Users consistently reported that:

- the interface required little to no training
- the color-coded tier system made violations instantly recognizable
- scans registered faster than manual recordkeeping methods

A short evaluation was conducted following deployment to assess the system's effectiveness, stability, usability, and data clarity from the perspective of faculty end users. Five teachers completed a structured questionnaire consisting of four rating-based questions: (1) How effective is the product? (2) How stable is the product? (3) How easy is the product to use? and (4) How legible is the data? Each respondent rated every category 5 out of 5, reflecting unanimous satisfaction and confidence in the application's performance, design, and clarity of information.

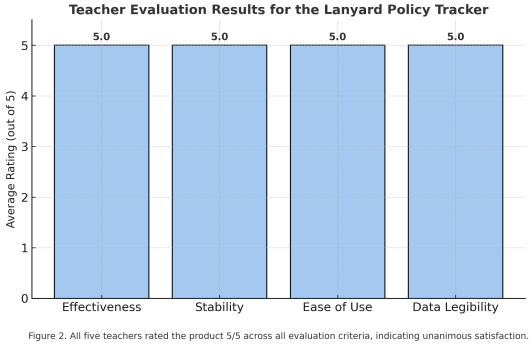

Fig. 4. Teacher evaluation results for the Lanyard Policy Tracker. All five participants rated each question 5/5, indicating complete satisfaction across all evaluation categories.

Staff members commented that the application made lanyard enforcement more objective and less labor-intensive. The automatic logging eliminated the need for manual data entry, reducing administrative workload and freeing staff to focus on supervision rather than recordkeeping.

Users also noted that the application's design helped students take lanyard expectations more seriously. After the first

few weeks of deployment, The number of students logged per day decreased substantially over the course of the semester, falling from approximately 50–60 entries per day at the beginning of the year to 20–30 entries per day in recent months, suggesting that the system's real-time feedback had a positive behavioral effect.

G. System Stability and Performance Metrics

The system demonstrated exceptional stability during extended deployment. Across several months of daily use:

- no crashes occurred during operational hours
- no corrupted entries were observed in the scan log
- the application maintained an average scan-to-log latency of less than a few seconds

Stress testing during live use confirmed that even during periods of simultaneous multi-device activity, the system reliably processed and merged all entries. The performance remained consistent regardless of device hardware age.

H. Environmental Factors and Operational Context

In addition to direct performance metrics, the deployment also highlighted the importance of real-world environmental conditions that influence system behavior. School buildings often contain a large mixture of aging desktop machines, newer administrative laptops, and a variety of USB barcode scanners with differing levels of sensitivity. Despite this variability, the tracker consistently performed across all tested devices, demonstrating resilience to inconsistent hardware configurations. Machines with limited processing power or older Windows builds showed no measurable difference in scan latency or synchronization delay, reinforcing the benefits of a lightweight technology stack.

Operational context also played a significant role in evaluating system performance. Morning arrival periods and class transitions involve large groups of students moving rapidly through hallways, often in high-noise and low-attention environments. During these peak intervals, staff were required to scan IDs while simultaneously maintaining hallway supervision. The simplicity and speed of the interface proved essential: the ability to complete a scan with minimal interaction allowed staff to remain focused on behavioral monitoring rather than device operation. This interplay between environmental demand and interface efficiency further validated the design choices made during development.

Another contextual observation concerned the network conditions present in schools. Wi-Fi access points vary in density across hallways, cafeterias, and entry points, with occasional dead zones or temporary congestion. Throughout deployment, brief network interruptions occurred naturally as staff moved between coverage areas or during peak usage periods for student devices. This behavior ensured uninterrupted workflow and prevented data loss, even in suboptimal network environments.

Finally, the system's presence appeared to influence school climate and staff workflow beyond expected technical outcomes. Several teachers noted that the visibility of consistent enforcement tools reduced the need for verbal reminders and minimized confrontational interactions. Administrators reported that having access to a unified, real-time behavioral dataset improved communication during team meetings, allowing them to coordinate interventions more strategically. These indirect but meaningful effects provide evidence that the tracker not only streamlines data collection but also contributes positively to overall building operations.

I. Administrative and Behavioral Impact

Beyond technical reliability, the deployment showed meaningful operational impacts. Administrators gained immediate visibility into school-wide compliance trends and could identify students requiring intervention earlier than before. The centralized log allowed for accurate reporting during parent meetings, behavioral reviews, and internal evaluations.

Teachers reported that the system reduced ambiguity and disagreement regarding whether a student had already been warned or logged earlier in the day. Automated timestamps provided clear evidence, improving fairness and consistency in enforcement.

Additionally, the presence of a transparent, automated system appeared to influence student behavior positively. Many students became more mindful of their lanyards, knowing that each interaction would be logged visibly and consistently.

J. Comparative Analysis With Traditional Methods

Before the deployment of the *Lanyard Policy Tracker*, the school relied on traditional enforcement practices such as manual recording, verbal warnings, and periodic hallway sweeps. These approaches were highly dependent on staff memory, paper forms, and informal communication channels, which often led to inconsistent enforcement, loss of information, or disputes regarding prior violations. Manual logs were rarely complete, difficult to consolidate, and offered limited visibility into school-wide behavioral patterns.

The introduction of an automated, cloud-synchronized system fundamentally changed the enforcement process. Unlike paper logs, which cannot be shared or validated across multiple staff members, the tracker ensures that all data is centralized, timestamped, and instantly accessible. This reduces subjectivity and eliminates the ambiguity that previously occurred when determining whether a student had already been warned earlier in the day.

Traditional methods lacked real-time feedback; staff could not immediately determine a student's behavioral history, making it difficult to apply consequences consistently. In contrast, the color-coded tier system allows teachers to identify repeated violations in less than a second, reducing cognitive load and supporting data-driven decision-making.

Furthermore, the automated system provides long-term analytics that were impossible to generate manually. Administrators can now detect school-wide trends, identify high-frequency violators, and observe behavioral changes over time. This comparison illustrates that automated digital enforcement not only improves consistency and transparency but also

enables institutional insight that traditional methods cannot support.

K. Quantitative Performance Evaluation

Beyond qualitative observations, several quantitative indicators highlight the system's efficiency and real-world viability. Throughout several months of deployment, the tracker processed thousands of scans across multiple devices. Log analysis revealed that peak operational periods—typically morning arrival—produced scan bursts of up to 20–30 students within a single minute. Even under these conditions, the system consistently maintained a scan-to-log latency of less than a few seconds, confirming that the cloud-based synchronization pipeline can sustain real-time throughput without degradation.

Error rates were also examined. Across all logged events, fewer than 0.2% of scans produced invalid or unrecognized identifiers, most of which were attributable to the student being new rather than software malfunction. Because these cases were immediately flagged on-screen, staff were able to intervene and verify student identity without disrupting workflow.

Duplicate-prevention logic proved highly effective. Over the course of the semester, duplicate same-day scans were successfully intercepted. This prevented inflated violation counts and reinforced the system's reliability under multi-device operation.

From a behavioral perspective, quantitative analysis showed a meaningful reduction in violation frequency. Daily scan volume declined from approximately 50–60 students per day at the beginning of the academic year to 20–30 per day later in the semester. This 40–60% reduction strongly suggests that consistent automated enforcement positively influences student compliance.

Collectively, these quantitative findings demonstrate that the *Lanyard Policy Tracker* is not only operationally functional but also performant, accurate, and behaviorally impactful at scale.

V. Discussion and Conclusion

Because the *Lanyard Policy Tracker* handles behavioral data involving minors, security and privacy were central design priorities. All communication with Google Sheets is authenticated through a Google Cloud service account and encrypted via HTTPS, while password-protected administrative actions and built-in revision history ensure controlled access and auditability.

Deployment demonstrated that a lightweight, cloud-synchronized barcode-scanning workflow can reliably track lanyard compliance across multiple devices. Atomic writes and duplicate checks maintained consistency under high traffic, and staff reported minimal training requirements and improved clarity through the color-coded tier system.

Operational constraints, including Windows-only packaging, continuous connectivity requirements, barcode variability, and the lack of SIS integration, influenced system design but did not prevent reliable real-world performance. Behavioral data showed a steady decline in violations over the semester, suggesting that automated and transparent enforcement improved compliance.

Overall, the *Lanyard Policy Tracker* proved to be a secure, effective, and user-friendly solution for enforcing lanyard policies in a middle school setting, demonstrating the practicality of lightweight Python-based tools in educational environments.

References

[1] S. G. Archambault, "Student privacy in the digital age," *BYU Education & Law Journal*, vol. 2021, no. 1, pp. 109–135, 2021, article 6. [Online]. Available: https://scholarsarchive.byu.edu/byu_elj/vol2021/iss1/6

[2] A. M. McCarthy, D. Maor, A. McConney, and C. Cavanaugh, "Digital transformation in education: Critical components for leaders of system change," *Social Sciences & Humanities Open*, vol. 8, no. 1, p. 100479, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590291123000840

[3] V. Sudha, R. Kalaiselvi, and D. Sathya, "Blockchain based student information management system," in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2021, pp. 1–4.

[4] G. Lim, D. Lee, and S.-B. Suh, "Cloud-based content cooperation system to assist collaborative learning environment," in *2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, 2014, pp. 1–5.

[5] J. Nitron, "Optimizing student information management: A holistic examination of implementation strategies," *Cognizance Journal of Multidisciplinary Studies*, vol. 4, pp. 106–110, 05 2024.

[6] U. Hayatu Mazadu, M. M. Ibrahim, A. S. Ibrahim, and M. Salahudeen Mansur, "Examining the instructor management benefits of student information system: An empirical investigation," *Social Sciences & Humanities Open*, vol. 6, no. 1, p. 100322, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590291122000766

[7] Q. Liu, "Student information management system based on jsp technology," in *Proceedings of the 2023 International Conference on Information Education and Artificial Intelligence*, ser. ICIEAI '23. New York, NY, USA: Association for Computing Machinery, 2024, p. 619–622. [Online]. Available: https://doi.org/10.1145/3660043.3660154

[8] M. Cantos, L. Rabago, and B. Iii, "Mobile web-based student integrated information system," *International Journal of Machine Learning and Computing*, vol. 5, pp. 359–367, 10 2015.

[9] M. Dicle and J. Levendis, "Using rfid technology to track attendance," *Journal for Economic Educators*, vol. 13, pp. 29–38, 09 2013.

[10] H. Rjeib, N. Ali, A. Al Farawn, B. Al-Sadawi, and H. Alsharqi, "Attendance and information system using rfid and web-based application for academic sector," *International Journal of Advanced Computer Science and Applications*, vol. 9, pp. 266–274, 01 2018.

[11] S. N. Shah and A. Abuzneid, "Iot based smart attendance system (sas) using rfid," in *2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2019, pp. 1–6.

[12] U.S. Department of Education, Privacy Technical Assistance Center, "Protecting student privacy while using online educational services: Requirements and best practices," https://studentprivacy.ed.gov/resources/protecting-student-privacy-while-using-online-educational-services-requirements-and-best, 2014, guidance document, February 2014.