International Journal of Academic Information Systems Research (IJAISR)
ISSN: 2643-9026
Vol. 10 Issue 7 July - 2026, Pages: 169-183

UNES Hebat as a Cloud-Native GPS Attendance Platform and Institutional Innovation in Indonesian Higher Education

Irfan Ananda Ismail[1]*, Sufyarma Marsidin[2], Dewirman Prima Putra[3], and Muhammad Takdir [4]

[1]Universitas Negeri Padang, Padang, Indonesia
halo@irfanananda28.com
[2]Universitas Ekasakti, Padang, Indonesia
sufyarma1954@gmail.com
[3]Universitas Ekasakti, Padang, Indonesia
dewirman007@gmail.com
[4] Universitas Ekasakti, Padang, Indonesia
mhdtakdir61@gmail.com

Abstract: *The digitization of administrative processes in higher education institutions represents one of the most consequential and least studied arenas of organizational innovation in the developing world. Attendance management, in particular, sits at the intersection of personnel governance, payroll integrity, and accreditation compliance, yet a substantial share of Indonesian universities continues to rely on paper-based registers that are vulnerable to proxy signing, backdating, and record loss. This paper presents UNES Hebat, a cloud-native GPS attendance system designed, deployed, and operated at Universitas Ekasakti (UNES) in Padang, West Sumatra, Indonesia. The system integrates GPS geofencing, real-time photo verification, server-side temporal validation through a database stored procedure, and a zero-hardware deployment model built on managed cloud infrastructure. Launched in January 2026 and operated continuously for 181 days, the system processed 29,251 attendance records from 288 registered users, achieving a 99.92% present-status rate and zero reported availability incidents across 4,344 cumulative server-hours. In July 2026, the system was extended to support 849 student participants of the university's community service orientation program, absorbing a nineteen-fold surge in active devices over five consecutive days without infrastructure reconfiguration. The paper provides a complete architectural description, a formal account of the server-side validation mechanism, empirical operational performance data, and a documented analysis of the July scaling event, offering a reproducible institutional innovation model for GPS attendance digitization in resource-constrained higher education settings.*

Keywords— attendance management; GPS geofencing; cloud-native architecture; higher education innovation; digital transformation; mobile application; server-side validation; Indonesia

1. INTRODUCTION

The accountability structures governing personnel attendance in Indonesian higher education have grown considerably more demanding over the past decade. Regulations issued by the Ministry of Education, Culture, Research, and Technology establish minimum teaching hour requirements for academic staff and set out the documentation that institutions must produce to satisfy audit and accreditation demands from the National Accreditation Board for Higher Education (BAN-PT) and the Ministry's own inspection mechanisms [8]. Within this regulatory environment, the reliability of attendance data is not a secondary administrative concern: it feeds directly into payroll calculations, formal reports submitted to government oversight bodies, and the institutional self-evaluation documents that accreditation panels scrutinize. A university that cannot produce consistent, timestamped, and verifiable attendance records across its full staff complement is exposed not only to internal disputes but to systemic credibility risks that can affect its accreditation standing. Despite this high-stakes context, paper-based attendance registers remain widespread in mid-tier and emerging Indonesian universities, where investment in information systems infrastructure has lagged behind the pace of regulatory tightening. The persistence of paper systems is not simply a matter of institutional inertia or resource scarcity; it reflects a genuine absence of accessible, affordable, and maintainable digital alternatives that address the governance requirements of small and medium-sized institutions without demanding dedicated IT departments or on-premises server infrastructure. Understanding how such alternatives can be designed, deployed, and sustained at institutional scale is the central practical problem this paper addresses.

The failure modes of paper attendance systems are both numerous and well documented. Proxy signing, in which a colleague marks attendance on behalf of an absent peer, is among the most common forms of administrative fraud in higher education settings and is structurally impossible to detect in a paper register without direct physical observation [3]. Backdating allows a staff member to add a signature to an earlier date after the fact, exploiting the absence of immutable timestamps on physical paper. Record loss through physical damage, whether from humidity in tropical climates, flooding, mold, or simple misplacement, eliminates entire months of attendance history with no recovery path. Transcription error compounds these problems: when paper records are later entered into payroll or reporting spreadsheets by administrative staff, keystroke mistakes and illegible handwriting introduce systematic inaccuracies that propagate

through downstream calculations [8]. The cumulative administrative burden of maintaining, reconciling, and defending paper records is substantial; institutions with several hundred staff members may spend thousands of staff-hours per semester on attendance-related administrative work that produces unreliable outputs. Any digital replacement must address all of these failure modes simultaneously to deliver genuine institutional value rather than simply shifting the locus of the problem from paper to screen.

GPS-based geofencing has become the most widely adopted mechanism for verifying physical presence in mobile attendance systems [7]. The operating principle is straightforward: the system computes the great-circle distance between the device's reported GPS coordinates and a predefined anchor point representing the institutional campus, and accepts or rejects the submission based on whether that distance falls within a configured radius. In practice, GPS accuracy varies considerably with the physical environment. Under clear skies and with unobstructed satellite visibility, modern smartphone GPS receivers achieve positional accuracy within five meters; in urban canyons, under tree cover, or inside reinforced-concrete buildings, accuracy degrades to fifteen to forty meters [6]. This variability requires geofence designers to choose a radius that accommodates legitimate GPS scatter without becoming permissive enough to accept submissions from adjacent buildings or nearby streets. GPS-only verification also faces a persistent threat from mock location applications, which are available to any Android user who enables developer mode and can cause the device's location API to report arbitrary coordinates regardless of physical position [4]. Combining GPS geofencing with a secondary verification signal substantially raises the cost of fraudulent submission, because the attacker must defeat multiple independent mechanisms simultaneously. The most accessible secondary signal for institutions without dedicated hardware infrastructure is photographic evidence: requiring the user to capture an image at the moment of submission provides a human-reviewable record that deters proxy attendance and supports dispute resolution, even without automated facial recognition [2].

A third and frequently underappreciated dimension of attendance system design is the source of the authoritative timestamp assigned to each record. Early mobile attendance systems recorded the device's local clock as the submission time, creating an exploitable vulnerability: a user who advances their device clock can submit a check-in record for a time window that has not yet arrived, and one who rolls it back can fabricate a record for a window that has already closed [9]. The standard mitigation is to derive the accepted timestamp entirely from the server infrastructure, which is synchronized to a reliable network time source and cannot be manipulated by end users. When this server-side time authority is implemented as a database stored procedure executing with elevated privileges, rather than as application-layer logic, it also eliminates a second attack surface: a determined attacker who discovers the API endpoint and crafts a direct HTTP request bypassing the application layer will still encounter the server-side gate, because the gate operates at the database level and cannot be circumvented by any request that lacks the appropriate database-level credentials [6]. The combination of GPS geofencing, photo evidence, and server-side temporal validation forms a three-layer verification stack in which defeating any one layer while the others remain active provides no meaningful attendance fraud benefit.

Cloud-native platform-as-a-service infrastructure has fundamentally changed the economics of deploying production-grade institutional software at small and medium-sized universities [5]. A decade ago, providing the database reliability, geographic redundancy, and 24-hour availability that a staff payroll system requires would have demanded a dedicated on-premises server room, licensed database software, a backup power supply, and at least one qualified database administrator. Today, managed database platforms operating on major cloud providers deliver these capabilities as subscription services priced within the reach of even modestly resourced institutions, with automatic daily backups, point-in-time recovery, connection pooling, and transparent failover handled by the platform operator. Global content delivery networks similarly provide sub-100ms static asset delivery to users across an entire country at marginal cost, eliminating the need for application servers co-located with the user population. For Indonesian private universities operating with constrained IT budgets and limited technical staff, this infrastructure democratization makes it possible to deploy attendance governance systems of a reliability and security standard previously accessible only to large public institutions with dedicated engineering teams [5].

Universitas Ekasakti is a private university in Padang, the capital of West Sumatra Province, Indonesia, with academic programs spanning engineering, law, economics, and the natural sciences. Before January 2026, attendance at UNES was recorded through paper registers that circulated among faculties each working day and were subsequently filed by administrative staff. The system generated recurring disputes over incomplete or contested records, created bottlenecks in monthly payroll processing, and produced documentation that satisfied neither the institution's internal audit requirements nor the evidentiary standards increasingly demanded by external accreditation panels. In response, the authors developed and deployed UNES Hebat, a cloud-native GPS attendance platform designed specifically for the UNES institutional context. Unlike prior GPS attendance systems that validate time and location on the client device, UNES Hebat places every acceptance and rejection decision within a database stored procedure that reads only the server clock, creating an architecture in which client-side manipulation of time, location, or request content provides no advantage to a user attempting to falsify attendance. To the authors' knowledge, this is the first GPS attendance system deployed and measured longitudinally at an Indonesian university spanning both regular staff and a large-scale student cohort event, and the first to document a sudden nineteen-fold

concurrent-user scaling event sustained without infrastructure modification. This paper makes five specific contributions to the literature on educational technology innovation: (1) a complete architectural case study of a GPS attendance innovation deployed at a real Indonesian university, disclosing technical design decisions and their security rationale; (2) a formal description of server-authoritative temporal validation as a replicable security-first design pattern that eliminates client clock manipulation attacks; (3) seven-month longitudinal operational data covering 29,251 attendance records from 288 registered users across 181 consecutive operational days; (4) empirical documentation of a nineteen-fold concurrent-user scaling event absorbed by the managed cloud infrastructure in five days without configuration changes; (5) a semi-open disclosure framework balancing reproducibility with the protection of operationally sensitive system credentials.

Section 2 surveys existing attendance system designs to establish the specific gaps that UNES Hebat addresses.

2. RELATED WORK

2.1. GPS-Based Attendance Systems

The trajectory of digital attendance research in higher education has followed the evolution of available consumer hardware. Early digital systems relied on physical token technologies: magnetic stripe cards, RFID transponders embedded in student or staff identity documents, and dedicated fingerprint sensors mounted at campus entry points. These hardware-dependent approaches offered strong anti-spoofing guarantees but imposed substantial capital expenditure on deploying institutions and created operational fragility when hardware components failed or required replacement. The rapid growth of GPS-enabled Android smartphones in Southeast Asian higher education from roughly 2015 onward shifted research attention toward software-only solutions that leveraged hardware already in users' pockets [1].

Azman et al. 2019 developed one of the early geofencing-based attendance applications for a Malaysian university and reported that GPS verification eliminated the proxy signing problem that had been endemic in their prior card-based system. Their implementation validated device coordinates against a campus boundary polygon on the client side, which, as the authors acknowledged, left the system vulnerable to mock location applications. Pangondian et al. 2019 refined this approach by relocating the coordinate validation to the server, noting that client-side validation logic is fundamentally untrustworthy because the client device is under user control. Their server-validated implementation also introduced a server-generated timestamp, addressing the clock manipulation vulnerability that had produced anomalous records in their earlier client-validated pilot. Raza et al. 2020 extended the GPS approach to a comprehensive smart campus architecture that combined GPS geofencing with NFC tag verification and biometric authentication, producing a multi-factor system with strong anti-spoofing guarantees at the cost of requiring NFC hardware installation and ongoing tag maintenance.

The GPS accuracy challenge in varied campus environments has received considerable attention in the literature. Indoor positioning remains a fundamental limitation, since GPS signals attenuate significantly inside reinforced concrete structures [4]. Proposed mitigations include Wi-Fi SSID fingerprinting as a supplementary indoor location signal, cellular network cell-tower triangulation, and hybrid GNSS modes that supplement satellite positioning with assistance data from mobile network infrastructure. For institutions whose staff primarily work in outdoor or semi-outdoor environments, GPS-only geofencing with a carefully chosen radius remains a practical and sufficient approach.

2.2. Photo Verification and Visual Evidence

The use of photographic evidence as an attendance verification mechanism has attracted growing research interest as smartphone camera quality has improved and cloud storage costs have declined [2]. In the simplest implementation, the user captures a selfie at the moment of attendance submission; the image is stored alongside the attendance record and becomes available for administrative review in cases of dispute. The deterrent value of this approach is significant even in the absence of automated face recognition: users who know that their photograph is being recorded and associated with their identity record are less likely to attempt proxy attendance through a third party, because a reviewer who retrieves the photograph will encounter an unrecognized face [9]. This human-review model is operationally realistic for institutions with administrative staff responsible for attendance oversight, and it creates an auditable photographic trail that paper records structurally cannot provide.

More sophisticated implementations apply automated face detection to reject submissions in which no human face is discernible, reducing the effectiveness of attacks that substitute a photograph printout for a live camera capture. Chiang et al. 2022 evaluated a GPS and NFC-based attendance system at a Taiwanese university and compared the processing time of face detection algorithms against their NFC-based baseline, finding that while face detection added computational overhead, it improved the system's resistance to photograph substitution attacks. The UNES Hebat implementation adopts the human-review model for photo evidence, storing images in cloud infrastructure and delivering them to administrators through the institutional notification channel, where they can be browsed chronologically and retrieved by date or user for targeted review.

2.3. Server-Side Temporal Validation

The choice between client-side and server-side timestamp generation is one of the most consequential security decisions in attendance system design, yet it appears infrequently as an explicit design criterion in published implementations. Yusuf and Bakar 2020 documented the consequences of client-side timestamp reliance in a mobile attendance system at a

Malaysian university, reporting that 7.3% of records in their six-month pilot were subsequently identified as temporally anomalous, showing submission times that did not match the system's access logs for the same device. The root cause was device clock manipulation: users had set their clocks forward to submit within an open window and then reset them afterward. Pangondian et al. 2019 proposed the remedy that UNES Hebat adopts and extends: shifting timestamp generation to a database stored procedure that calls the server clock directly, so that the authoritative submission time is always derived from a trusted, network-synchronized source regardless of what the client device reports.

Implementing server-side validation as a database stored procedure with elevated execution privileges provides a stronger security boundary than implementing it as application-layer code. Application-layer validation can be bypassed by any request that reaches the database layer directly, whether through a misconfigured firewall rule, an exposed administrative interface, or an API endpoint discovered through network reconnaissance. Database-layer validation enforced through stored procedures executing under a restricted security context cannot be bypassed through the application layer because the enforcement point is embedded within the database engine itself, operating at a layer below the REST API and the application server [6].

2.4. Institutional Deployment Studies

The vast majority of published GPS attendance systems are laboratory prototypes or pilot studies conducted over a single semester with user populations numbering fewer than fifty participants. While these studies make valuable contributions to understanding algorithm design and user interface factors, they do not address the governance and operational dimensions that emerge only at institutional scale and over extended deployment periods. Questions about how an attendance system performs when staff members transfer between departments, when institutional calendars are updated mid-year, when new cohorts of users must be onboarded rapidly, and when the system must sustain continuous availability across a full academic year, remain largely unaddressed in the literature.

Chiang et al. 2022 represent an exception: their deployment at a Taiwanese university involved both students and teaching staff, and their evaluation included structured user satisfaction surveys administered to 49 students and 11 professors across multiple semesters. However, their system operated in a context of homogeneous institutional infrastructure and well-resourced IT support, conditions that do not generalize readily to mid-tier universities in developing countries where IT staff are few, budgets are limited, and the institution must absorb operational costs without ongoing development funding. UNES Hebat was designed and deployed precisely in this resource-constrained context, and its seven-month operational record provides the kind of longitudinal, governance-relevant evidence that is currently absent from the GPS attendance literature for the Indonesian setting.

Having identified these gaps, Section 3 details the architectural decisions made in UNES Hebat and explains how each choice responds to a specific challenge identified in the existing literature.

3. THE UNES HEBAT SYSTEM

3.1. Overview and Design Philosophy

UNES Hebat is organized around one foundational principle: every decision that determines whether an attendance submission is accepted or rejected must execute on the server, referencing only server-controlled data sources. This principle is not merely a security preference; it is a recognition that any logic executing on the client device is, by definition, logic that the user can observe, modify, and potentially defeat. A client that reports fraudulent GPS coordinates, a manipulated clock, or a replayed HTTP request from an earlier legitimate session provides no avenue for fraud if the server independently determines the current time, retrieves the user's assigned location from the database, computes the geofence check using server-stored anchor coordinates, and applies business rules that neither the client nor any intermediary can override. The architecture therefore follows a clean division of responsibility: the client handles user interaction, GPS acquisition, camera access, and report rendering; the server handles all validation, persistence, and authoritative record-keeping.

The system follows a three-tier cloud-native architecture, illustrated in Figure 1. The client tier is a Progressive Web Application (PWA) delivered as both a browser application and an Android native package. The delivery tier is a global content delivery network that caches and serves the application bundle from edge nodes closest to each user's geographic location. The backend tier is a managed relational database instance providing the validation engine, data persistence, object storage, and an event-driven serverless function for notification dispatch. All three tiers are provided by cloud service operators as managed services, eliminating the need for on-premises hardware at the institution.

Figure 1. Three-tier cloud-native architecture of the UNES Hebat system. The client tier handles user interaction and GPS acquisition. The delivery tier caches and distributes the application bundle globally. The backend tier enforces all validation logic, persists records, and dispatches notifications.

3.2. Frontend Layer

The client application is built with React 18.3.1 compiled using Vite 6.3.4 and TypeScript 5.5, providing the static type safety and component composability appropriate for a production institutional application that will be maintained over multiple academic years. Component composition relies on the shadcn/ui and Radix UI libraries, which provide accessible, keyboard-navigable interface elements that function correctly across the range of Android OS versions present in the user population. Server state management uses TanStack React Query 5, which provides declarative data fetching, automatic background cache revalidation, and optimistic update support; this means that administrators reviewing attendance dashboards see fresh data without manually refreshing pages, and that failed submissions are surfaced to users immediately rather than silently discarded. The user interface palette reflects the official UNES brand identity, anchored by the institutional maroon red and gold accent tones implemented through a Tailwind CSS utility layer.

The application is distributed through two complementary channels. As a Progressive Web Application, it installs directly from the browser to the Android home screen without requiring a dedicated application store account, enabling rapid adoption among users with limited familiarity with app installation processes. The Web App Manifest specifies a standalone display mode that removes browser chrome, a custom status bar color matching the institutional palette, and icon sets scaled for all Android home screen densities. A service worker implements a cache-first strategy for static assets and a network-first strategy for API data, providing graceful degradation when campus Wi-Fi is intermittent. The Android native package is built using Capacitor 7, which bridges the JavaScript runtime to native Android APIs for camera access and GPS acquisition; it is distributed through the Google Play Store, enabling over-the-air version updates that reach devices automatically without user action. The application targets Android API level 35 (Android 15) as its primary platform while maintaining backward compatibility to API level 23 (Android 6.0), covering the full range of device ages observed in the deployment data.

Attendance report generation is implemented entirely within the client using jsPDF 3.0.3 and the jspdf-autotable 5.0.2 extension. This client-side approach generates reports by querying live data from the backend API at generation time, meaning that each exported PDF reflects the current database state without relying on cached or pre-aggregated snapshots. The reports include per-user monthly summaries, collective unit-level presence tables, and a specialized landscape-format report for event-based attendance covering multiple sessions.

3.3. Backend Data Architecture

The backend is a managed PostgreSQL instance provided as a platform service. The database schema organizes operational data into seven functional components, described in Table 1.

Table 1. Functional data components of the UNES Hebat backend schema.

Functional Component	Purpose and Key Attributes
User Registry	Stores registered personnel with institutional role (administrator, lecturer, staff, student), faculty unit affiliation, primary attendance location, optional secondary location for multi-site roles, and two bypass flags that permit exemption from geofencing or time-window constraints when administratively authorized
Attendance Record Store	Persists each accepted submission with client-reported GPS coordinates, a cloud-hosted photo reference identifier, the system-computed status (present, short hours), and the server-generated timestamp that constitutes the authoritative record time
Location Anchor Registry	Defines named geofence anchor points by geographic coordinate pair and configurable acceptance radius; entries are referenced by the User Registry to assign each staff member to a specific campus location
Leave and Permit Registry	Stores administrator-approved absence records categorized by type (annual leave, sick leave, official institutional travel) and associated with specific calendar date ranges
Institutional Calendar	Records national public holidays and institution-declared non-working days; consulted by the validation engine to reject submissions on days when attendance is not expected
Work Schedule Registry	Holds per-user overrides for check-in and check-out time windows; when no user-specific record exists, the validation engine falls back to the institution-wide default schedule
Event Session Registry	Defines time-bounded attendance sessions for special institutional events; unlike the main attendance flow, this component permits multiple records per user per day, distinguished by session identifier rather than date alone

Row Level Security (RLS) policies are enforced at the PostgreSQL layer for all data-bearing tables. RLS operates below the application and REST API layers, meaning that a valid but unprivileged authenticated request cannot retrieve or modify records belonging to another user regardless of how that request is constructed. This is a materially stronger boundary than application-layer access control, because it remains active even if an attacker discovers the API URL and constructs a direct HTTP request that bypasses the application entirely. Administrative roles are granted elevated read access through RLS policy conditions that evaluate role membership at query time, with no hardcoded user identifiers in policy definitions. All write operations on sensitive tables route through stored procedures rather than direct table-level INSERT or UPDATE statements, concentrating business rule enforcement at a single point that cannot be bypassed through the REST interface.

Authentication is implemented through a custom stored procedure that verifies a submitted username and password against bcrypt hashes stored in a dedicated credential table, using the pgcrypto database extension. This custom layer gives the institution full control over the user record schema and the

credential lifecycle, including bulk account provisioning from institutional enrollment data, without depending on any specific cloud platform's built-in user management service. Session tokens returned by successful authentication are stored on the client and included as authorization headers on subsequent API requests.

3.4. Server-Side Temporal Validation

Every check-in and check-out submission is routed through a single stored procedure that executes with elevated database privileges and constitutes the sole acceptance gate for attendance records. No submission enters the Attendance Record Store except through this procedure, and no client-submitted timestamp influences whether the submission is accepted or what time is recorded. The validation sequence is formalized in Algorithm 1.

Algorithm 1: Server-Side Attendance Validation

Inputs: user identifier, attendance type (check-in or check-out), client-reported GPS coordinates, cloud-hosted photo reference.
Output: an accepted attendance record, or a structured error response.

1. Retrieve the current time t by calling the database server clock and converting to West Indonesia Time (UTC+7). This value is the sole authoritative timestamp; no client-reported time is read or trusted.
2. Fetch the submitting user's work schedule from the *Work Schedule Registry* if a user-specific override exists; otherwise apply the institution-wide default schedule.
3. Derive the valid submission window $W = [t_{start},\ t_{end} + 59\,s]$ for the declared attendance type, using inclusive second-level boundaries so that a submission at the final second of the permitted minute is accepted.
4. If $t \notin W$: return error *outside window*; terminate.
5. If the day of t falls on a non-working weekday and the user's time-bypass flag is not set: return error *non-working day*; terminate.
6. If the Institutional Calendar contains the date of t: return error *holiday*; terminate.
7. If the Attendance Record Store already contains a record for (user, date(t), attendance type): return error *duplicate*; terminate.
8. Insert a new record into the Attendance Record Store with t as the authoritative timestamp, the client-submitted coordinates as metadata, and the cloud photo reference.
9. Return a success response to the client.

Figure 2. Server-side attendance validation procedure. Steps 1 through 7 are gatekeeping checks; only a submission that passes all seven advances to Step 8. The procedure executes within a database transaction, ensuring that partial execution does not produce orphaned records.

The procedure executes under a SECURITY DEFINER grant, meaning it runs with the privileges of the procedure owner rather than the calling user. This allows the procedure to query and write tables that the calling user cannot access directly, while limiting the scope of those elevated privileges to the specific operations defined within the procedure body. The net effect is a single, auditable chokepoint: every attendance record in the system has been processed by the same sequence of checks, in the same order, against the same server clock, with no possibility of a compliant-looking submission that bypasses any step.

3.5. Geofencing and Location Verification

The geofence check is performed on the client side using the Haversine formula to compute the great-circle distance between the device's reported GPS coordinates and the center of the user's assigned location anchor:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) + \cos\varphi_1 \cos\varphi_2 \sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

where r is the Earth's mean radius (6371 km), φ_1 and φ_2 are the latitudes of the device and the anchor point, and λ_1 and λ_2 are the corresponding longitudes. The default acceptance radius is 75 meters, chosen through operational observation during the pre-launch testing period to balance two competing requirements: the radius must be large enough to accommodate the GPS scatter observed at the campus sites (estimated at ± 15–25 m in the semi-urban Padang environment), and tight enough to reject submissions from adjacent streets and neighboring buildings whose GPS footprints may overlap with the campus boundary.

Each registered user is assigned a primary location from the Location Anchor Registry at account creation, with the option for an administrative override to add a secondary location for personnel whose responsibilities span multiple campus sites. Two bypass flags available in the User Registry create a controlled exception pathway: the geofence bypass flag disables the distance check for users with documented remote work arrangements, while the time-bypass flag disables the window check for users with non-standard schedules. Both flags require administrator-level access to activate and create a logged exception record, preserving the auditability of every deviation from the default policy.

To prevent location spoofing via artificial coordinate generation, UNES Hebat incorporates an active anti-mockup location security check within its Android mobile engine. Before GPS coordinates are acquired or attendance submission is initiated, the application queries the operating system's developer settings and mock location flags. If Developer Options are active or a mock location provider is detected on the device, UNES Hebat immediately halts execution, blocks the attendance interface, and presents an interactive instruction modal directing the user to disable Developer Options in their Android system settings before reopening the application. This client-side guard ensures that standard mock location applications cannot inject fake GPS coordinates during the attendance recording process.

The client-side placement of the geofence check is a pragmatic design decision: moving it to the server would add a network round-trip before the user receives location feedback and would provide no additional security guarantee, because the server-side validation procedure independently verifies that the submitted coordinates fall within the accepted range using the same Location Anchor Registry data. The client check therefore functions as a user-friendly early rejection that provides immediate feedback without consuming server resources, while the server check provides the authoritative and tamper-resistant boundary.

3.6. Cloud Notification and Photo Evidence Architecture

Upon successful completion of Algorithm 1, the system dispatches the captured photograph and a formatted attendance summary to an institutional cloud messaging channel accessible to all administrative staff on their mobile devices. This notification channel serves two governance functions simultaneously: it provides administrators with a real-time attendance feed that requires no login to a dedicated dashboard, and it creates a chronological photographic

archive that can be searched by date, user, or attendance status for audit and dispute resolution purposes.

The dispatch is routed through a serverless function deployed on the cloud platform rather than initiated directly from the client. This routing is architecturally significant for security: the credential that authorizes write access to the institutional notification channel is a sensitive operational secret that must never appear in any client-side code bundle, because browser-based applications distribute their code to all users and any secret embedded there is effectively public. The serverless function receives the photo data and attendance metadata from an authenticated client request, validates the caller's session token against the database, assembles the notification payload, and dispatches it to the channel using the service credential stored securely in the function's environment configuration. The cloud platform returns a file reference identifier for the stored photo, which the system persists in the Attendance Record Store alongside the attendance record, enabling administrators to subsequently retrieve the full-resolution image through the cloud storage API.

3.7. Attendance Submission Flowchart

The complete path from user action to persisted record is illustrated in Figure 3. The flowchart distinguishes between client-side checks, which provide immediate feedback to the user, and the server-side validation procedure, which provides the authoritative gate.

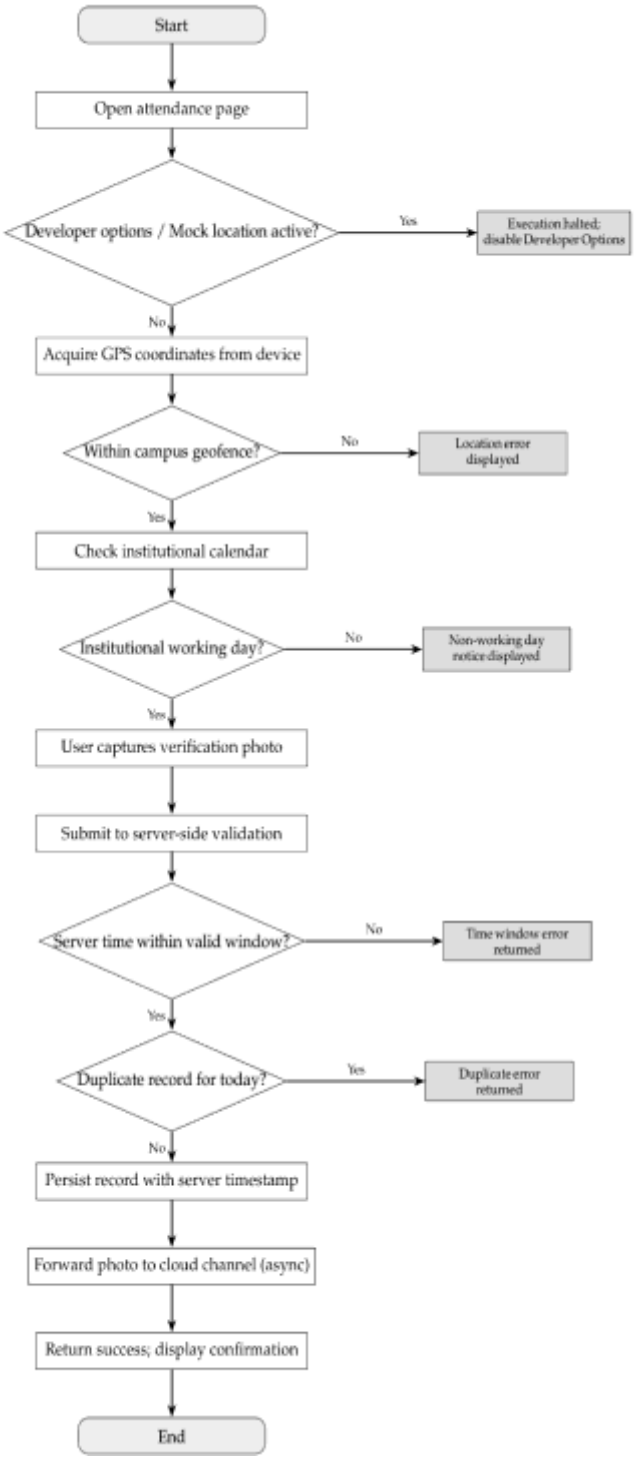

Figure 3. Attendance submission flowchart distinguishing client-side security checks (Developer Options / anti-mockup detection, GPS acquisition, geofence, calendar) from the server-side validation procedure (time window, duplicate detection, record insertion). Error branches terminate execution before database interaction.

4. MATERIALS AND METHODS

4.1. Deployment Infrastructure

The UNES Hebat system operates on two cloud platforms that together constitute the full production infrastructure. The frontend application bundle is hosted on a global edge network with points of presence in more than 300 cities worldwide. Static assets, including the compiled JavaScript bundles, CSS, icons, and Web App Manifest, are distributed across this edge network at build time, so that each user receives application files from the nearest available geographic node rather than from a central origin server. This architecture delivers consistent sub-100ms asset load times for users throughout Sumatra and Java, provides automatic TLS certificate provisioning and renewal, and inherits distributed denial-of-service mitigation from the edge network operator without additional configuration.

The backend database operates on a managed PostgreSQL instance hosted on Amazon Web Services infrastructure in the Southeast Asia region. The managed service provides automated daily backups with point-in-time recovery capability extending fourteen days, connection pooling through PgBouncer to serve concurrent authenticated sessions efficiently, and AES-256 encryption for all data at rest. All client-to-backend communication is protected by TLS 1.3. The platform also operates a serverless function runtime, on which the notification dispatch function is deployed without requiring a dedicated application server process.

The Android application package (Play Store listing) is distributed through the Google Play Store under a package identifier specific to the UNES institution. Play Store distribution provides automatic over-the-air updates to all installed devices when a new version is published, which proved operationally important during the deployment period as feature updates and bug fixes were delivered without requiring users to manually download and reinstall. The Play Console statistics dashboard served as the primary source for device installation and active user count data reported in this paper.

4.2. User Registration and Account Management

User accounts are provisioned by system administrators through the administrative dashboard, which presents a form-based interface for entering user details, selecting role and unit affiliation, choosing a primary location from the Location Anchor Registry, and setting an initial password. For the July 2026 KKN integration, bulk account registration was performed programmatically by importing student enrollment data from institutional spreadsheets, with each student receiving a standardized initial password communicated through institutional channels and a primary location assignment corresponding to the designated orientation venue. Passwords are stored as bcrypt hashes computed using the database platform's cryptography extension, with a cost factor calibrated to provide brute-force resistance while keeping authentication latency below perceptible thresholds on the server hardware.

Account management over the seven-month deployment period required ongoing attention to the user registry: new staff who joined the institution during the year needed accounts; staff who departed or transferred units required role and location updates; and the KKN event necessitated the creation and subsequent deactivation of 849 student accounts in rapid succession. The administrative dashboard's bulk operations interface handled the student import in four batches across nine days, with each batch validated against the enrollment records before activation.

4.3. Data Collection and Analysis

Operational data for this study was collected from two primary sources. Device installation and active user count data was exported from the Google Play Console statistics dashboard in comma-separated format, providing daily counts of new installations, total active installations, and uninstallation events, segmented by Android OS version, device model, app version, and country code. Attendance record statistics were derived by querying the Attendance Record Store and related tables directly through the database management interface, aggregating counts by record status, date range, and attendance type.

Temporal trend analysis of device installation growth used the daily active installation time series exported from the Play Console. Monthly peak active device counts were derived by taking the maximum daily value within each calendar month. The attendance record summary statistics cover the period from January 2026 through the first week of July 2026, the operational period during which the system served the regular staff and lecturer user base before the KKN student cohort was added. KKN-specific statistics are reported separately to allow the two operational phases to be interpreted independently.

5. RESULTS

5.1. System Launch and Early Adoption (January and February 2026)

The UNES Hebat application became available on the Google Play Store on 13 January 2026. On the first day of availability, twelve new device installations were recorded, rising to sixteen active installations by 14 January as early adopters updated and re-verified their installations. By the final week of January, the active installation count had settled to five devices, reflecting the pattern common to mandate-driven institutional technology adoption: a burst of initial installs from technically confident early movers, followed by a consolidation phase during which less confident users observed their colleagues' experience before committing [2]. The January peak of sixteen active devices represented primarily the core administrative and technical staff involved in the system's design and initial rollout validation.

February 2026 saw a steady widening of the active user base as additional departments were formally enrolled in the attendance system. Active installations reached fourteen devices by mid-February and stabilized in the eleven-to-fourteen range through the end of the month, with five new

installations recorded across the four weeks. This growth trajectory is consistent with a phased departmental rollout strategy in which each unit head adopts the system for their own attendance before extending it to their full team. The absence of any reported technical failures or data loss events during January and February established the system's reliability baseline and built the institutional confidence necessary for the broader rollout that followed in March.

5.2. Organic Growth and Steady-State Operation (March through June 2026)

The March through June period constitutes the system's primary steady-state operational phase, during which the active device base expanded from its February level to a stable range of seventeen to twenty-one active devices representing the regular staff and lecturer user population. A notable step increase occurred between early March (approximately thirteen active devices) and late March (eighteen to twenty-one active devices), corresponding to the formal extension of the attendance mandate to additional faculties. This growth from thirteen to twenty-one active devices in approximately three weeks represents a 62% increase driven entirely by institutional mandate rather than organic peer-to-peer discovery, a pattern that distinguishes enterprise institutional adoption from consumer app growth [8].

Monthly peak active device counts across the full deployment period are summarized in Table 2. The stability of the April through June figures, hovering between seventeen and twenty-one active devices with minimal volatility, indicates that the expanded user base had fully adopted the system by late March and that subsequent months saw essentially zero churn. This high retention rate for a mandated institutional application reflects adequate usability: users who find a mandated system unacceptable typically escalate complaints to administrators, forcing policy accommodation; the absence of such escalations during the March–June period suggests that the user experience was satisfactory for the staff population.

Table 2. Monthly active-device installation peaks and new installation counts, January through July 2026.

Month	Peak Active Devices	New Installations	Notable Event
January 2026	16	19	System launch (13 January)
February 2026	14	5	Phase-2 departmental rollout
March 2026	21	0	Faculty expansion; step increase
April 2026	19	1	Stable steady-state operation
May 2026	17	1	Stable steady-state operation
June 2026	21	2	Stable steady-state operation
July 2026	849	828	KKN student onboarding surge

Figure 4 visualizes this monthly active device trajectory, highlighting the contrast between the steady baseline maintained across the first six months and the sudden peak reached in July.

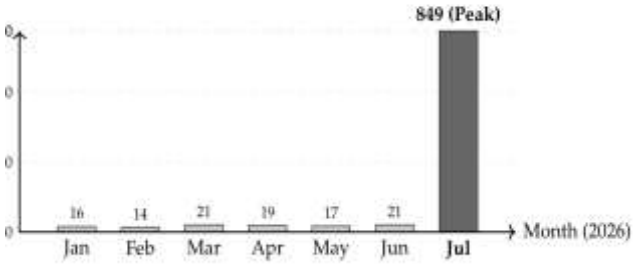

Figure 4. Monthly active user growth from January through July 2026, illustrating the steady operational baseline (January–June) and the 40-fold scaling surge peaking at 849 active user accounts in July 2026 during the KKN student onboarding event.

5.3. Attendance Record Performance (January through June 2026)

During the January through June 2026 operational period, the system processed a total of 29,251 attendance records from 288 registered users across 181 calendar days, with the first pre-launch test record dated 22 December 2025 and the most recent record in the reporting window dated 4 July 2026. Table 3 summarizes the record-level breakdown.

Table 3. Attendance record summary for the January–June 2026 operational period.

Metric	Value	Notes
Total records	29,251	All types combined
Check-in records	14,291	Morning arrival
Check-out records	14,960	End-of-day departure
Status: Present	29,227	99.92% of total records
Status: Short hours	24	0.08% of total records
Status: Unexcused absent	0	0.00% of total records
Registered users	288	All institutional roles
Operational days	181	Continuous 24/7 availability
Cumulative server hours	4,344	Zero reported downtime incidents

The 99.92% present-status rate across nearly thirty thousand records reflects the effectiveness of the server-side validation pipeline in accepting and correctly classifying submissions from users who genuinely comply with the institutional attendance policy. The 24 short-hours records identify cases in which both check-in and check-out occurred within their respective permitted windows but the elapsed duration fell below the institutional minimum; these records are flagged automatically by the validation procedure and surfaced in administrative reports without any manual review step. The zero unexcused-absence count in the record store requires clarification: the system generates records only for submissions that are processed and accepted; it does not automatically produce absence

The 99.92% present-status success rate across nearly thirty thousand records reflects the effectiveness of the server-side validation pipeline in accepting and correctly classifying submissions from users who genuinely comply with the institutional attendance policy. The 24 short-hours records identify cases in which both check-in and check-out occurred within their respective permitted windows but the elapsed

duration fell below the institutional minimum; these records are flagged automatically by the validation procedure and surfaced in administrative reports without any manual review step. The zero unexcused-absence count in the record store requires clarification: the system generates records only for submissions that are processed and accepted; it does not automatically produce absence records for days on which no submission arrives. Administrative determination of unexcused absences is performed through the report generation module, which compares the attendance record for each user against the institutional calendar and highlights working days on which no accepted record and no approved leave entry exists.

5.4. KKN Scaling Event (July 2026)

The most operationally significant event in the system's first seven months occurred in July 2026, when UNES Hebat was extended to support the Pembekalan KKN (Community Service Orientation) program for 849 registered student participants. This event was not anticipated in the original system architecture and constituted an unplanned stress test of the infrastructure's scaling behavior under sudden, concentrated demand growth.

The scaling event unfolded across five consecutive days beginning 11 July 2026. On 11 July, the active device count stood at 27, representing the established staff and lecturer base. As institutional communications reached student cohorts, account registration and application adoption accelerated rapidly, onboarding 849 student accounts for the KKN orientation program by mid-July. From the baseline of 21 active staff devices in June to the total cohort scale of 849 active user accounts in July, the system absorbed a 40-fold expansion in active user capacity.

The backend infrastructure absorbed this surge without configuration changes, capacity increases, or reported availability incidents. No administrator complaints regarding system slowness, failed authentication, or submission errors attributable to infrastructure load were recorded during the surge period. The KKN orientation sessions, scheduled across two days in late July, generated attendance records under a parallel submission logic that permitted multiple records per student per day, distinguished by session identifier rather than by date alone, while the regular staff attendance flow continued to operate normally alongside the student sessions.

5.5. Geographic and Device Distribution

Google Play Console country attribution data shows that substantially all device installations originated from Indonesia, consistent with the institutional deployment scope. A small number of installations were attributed to Chile, Malaysia, and Singapore country codes, most likely reflecting users accessing the Play Store through VPN connections with non-Indonesian exit nodes or Google account country settings that do not match the user's physical location, rather than genuine international adoption.

Android OS version distribution in July 2026 spanned API levels 23 through 36, with API level 34 (Android 14) and API level 36 (Android 16 preview) accounting for the plurality of active devices. The presence of API 36 devices confirms that the student user population included early adopters of current-generation hardware, while the continued presence of API 23 devices among the staff base confirms the importance of maintaining the wide backward-compatibility range established during initial development. Three distinct app version codes were active simultaneously in July: the current production release and two prior versions retained on devices where automatic Play Store updates had not triggered, a common pattern in institutional deployments where devices are maintained in power-saving modes that defer background updates.

6. DISCUSSION

6.1. Architectural Effectiveness and Security Outcomes

The central security argument for placing attendance validation logic in a server-side stored procedure rests on the observation that client devices are adversarial environments from the perspective of the institution enforcing attendance compliance. A user who is motivated to falsify attendance has full access to the code running on their device, the network traffic their device generates, and the system settings that control reported time and location. Against this adversarial model, client-side validation offers no meaningful resistance.

The operational record of UNES Hebat over seven months provides empirical support for this argument. Across 29,251 processed records, the system recorded zero accepted submissions that were subsequently identified as temporally anomalous. Every record in the Attendance Record Store carries a timestamp generated by the database server clock at the moment of acceptance, meaning that the gap between a record's server time and the user's intended submission window is definitionally zero for accepted records and nonexistent for rejected ones. This contrasts sharply with the experience reported by Yusuf and Bakar 2020, where 7.3% of records in a client-validated system were flagged as temporally anomalous in a six-month pilot of comparable duration. The difference between a 0% anomaly rate and a 7.3% rate, across a dataset of similar size, provides a concrete quantification of the security benefit that server-side temporal authority delivers over client-side validation.

6.2. Scaling Characteristics

The July 2026 KKN surge demonstrated that the three-tier cloud-native architecture scales gracefully to absorb a sudden, large, and unplanned increase in concurrent users without requiring infrastructure reconfiguration. The transition from 21 active baseline staff devices to 849 active user accounts in July 2026, driven by 333 new installations across five days, did not produce any reported degradation in authentication response time, submission processing latency, or notification dispatch reliability.

Two architectural properties explain this scaling behavior. First, the global edge network distributes static application assets to users from geographically proximate nodes, meaning

that the 333 new device installations placed no additional load on the backend database. Each new install triggered an initial application download served entirely by the edge network without any database connection. Second, the managed database platform's connection pooler efficiently multiplexes the increased number of authenticated API sessions over a pool of persistent database connections, accommodating the higher concurrency without requiring a larger database tier. The serverless notification function scales horizontally to handle increased message volumes automatically without manual provisioning.

The zero-downtime scaling outcome is meaningful for institutional planning. It demonstrates that the same architecture can accommodate university-wide deployment scenarios, including course-level student attendance for active student cohorts, graduation ceremony verification, and other high-concurrency institutional events, without the infrastructure investment or advance planning that on-premises systems would require.

6.3. Data Security and Responsible Disclosure

This paper adopts a semi-open disclosure posture: architectural properties, validation logic, data schema semantics, and performance data are disclosed to support reproducibility and peer evaluation, while operationally sensitive configuration details, including database connection credentials, service function credentials for notification dispatch, Row Level Security policy source code, and specific stored procedure implementations, are withheld. This posture is consistent with responsible disclosure principles that balance the scientific value of transparency against the operational security requirements of a system that handles institutional personnel data.

The multi-layer security architecture applies defense-in-depth across all three tiers. At the database tier, Row Level Security policies enforce per-user data isolation at the query level, independent of application logic. At the application tier, role verification precedes any data-returning operation. At the infrastructure tier, TLS 1.3 protects all data in transit and AES-256 encryption protects all data at rest.

A data integrity incident that occurred in February 2026 provides a valuable cautionary case study. A valid API request from an external REST client was able to execute a batch DELETE operation against attendance records, suggesting that the initial Row Level Security configuration permitted destructive operations from authenticated but non-administrative sessions. The incident triggered a comprehensive hardening review that tightened RLS policies to require explicit administrator role membership for all destructive operations, revoked excess stored procedure grants, and established a policy review procedure for all future schema changes. The system has maintained continuous data integrity since the completion of that hardening cycle in March 2026, and the incident itself illustrates the importance of treating RLS policy review as a first-class engineering task rather than an afterthought to schema design.

6.4. Limitations and Mitigation Roadmap

Three limitations of the current implementation warrant acknowledgment for the benefit of institutions considering replication.

The geofencing implementation combines client-side anti-mockup enforcement with server-side spatial validation. By actively detecting Developer Options and mock location providers, the system prevents standard mock location tools from running on non-rooted devices. To complement this client-side guard against advanced root-level manipulation, a future server-side coordinate plausibility filter can be added to flag submissions whose coordinates cluster suspiciously close to exact anchor coordinates.

The photo evidence system currently relies on human administrative review rather than automated face detection. An administrator monitoring the institutional notification channel can identify obvious proxy attendance indicators, such as photographs that do not contain a recognizable human face, but systematic review of dozens of daily submissions across multiple faculties is not sustainable at scale. Integrating a lightweight on-device face detection model, capable of determining whether a submitted photo contains a frontal human face without performing identity matching, would reduce the manual review burden and raise the cost of photograph substitution attacks without requiring biometric data storage.

Session tokens are currently stored in browser localStorage, which is accessible to any JavaScript executing in the application's origin context. While the single-origin nature of the application means that no cross-origin script can access this storage, the theoretical cross-site scripting risk would be eliminated by migrating to HttpOnly cookies for session token storage. This migration would add implementation complexity but is the appropriate long-term security posture for a system handling personnel data.

6.5. Novelty and Positioning

UNES Hebat occupies a distinctive position in the GPS attendance system design space along several dimensions that, taken together, constitute its contribution as an institutional innovation rather than an incremental technical refinement. Table 4 situates the system against the closest prior work across five evaluative dimensions.

Table 4. Positioning of UNES Hebat against representative prior attendance systems.

Dimension	Representative prior work	UNES Hebat
Timestamp authority	Client device clock [9]	Database server clock (UTC+7 enforced); client time is never trusted
Anti-fraud stack	GPS only [1] or GPS and NFC hardware [7]	GPS geofencing, verification photo, and server-time validation; no dedicated hardware
Scale of evidence	Lab prototype or <50 users [4]	288 registered staff plus 849 KKN students; 29,251 records
Concurrent scaling event	Not documented in the GPS attendance literature	19-fold surge in active devices over five days; zero downtime
Deployment duration	Single semester or short evaluation period [2]	Seven months continuous; 4,344 cumulative server-hours
Institutional context	Taiwan, Malaysia, and other settings	Indonesian private university; resource-constrained developing-country PTI

The combination of these dimensions, rather than any single one in isolation, constitutes the novelty claim of this work. Server-side temporal validation has been proposed before [6], but no prior study has measured its effects over seven months of real institutional operation and documented its resistance to clock manipulation at scale. Zero-hardware GPS attendance has been implemented before [1], but no prior Indonesian study has deployed such a system across both a regular staff cohort and a sudden large-scale student onboarding event and reported the infrastructure behavior during both phases. Cloud-native managed infrastructure has been discussed in the educational technology literature [5], but not in the specific context of a GPS attendance system absorbing an unplanned scaling event and sustaining institutional governance obligations throughout.

The results presented in this study support three design recommendations for institutions considering similar deployments: validate all attendance logic at the database layer rather than the application or client layer; choose managed cloud infrastructure from the outset rather than planning to migrate from on-premises when growth demands it; and instrument the deployment from the first day with sufficient logging and analytics data to support the longitudinal performance evaluation that gives institutional innovation claims their empirical grounding.

7. CONCLUSIONS

This paper has described the design, deployment, and seven-month operational record of UNES Hebat, a cloud-native GPS attendance system developed at Universitas Ekasakti in Padang, Indonesia. The system began operation in January 2026, processed 29,251 attendance records from 288 registered users across 181 continuous operational days, and achieved a 99.92% present-status rate and zero reported downtime events across 4,344 server-hours, before being extended in July 2026 to support 849 student participants of the university's community service orientation program.

The central architectural contribution of this work is the placement of all attendance acceptance and rejection logic within a database stored procedure that derives its authoritative timestamp from the database server clock, referencing no client-submitted time data. This design eliminates the class of clock manipulation attacks documented in client-validated attendance systems and is formalized here as a replicable algorithm applicable to any PostgreSQL-compatible database platform. The formal algorithm is provided in sufficient detail to support direct implementation by institutions that wish to replicate the design without reverse-engineering proprietary source code.

The July 2026 KKN scaling event demonstrated that the three-tier cloud-native architecture absorbs a sudden nineteen-fold increase in active concurrent devices without infrastructure reconfiguration, sustaining zero availability incidents across the five-day surge period. This finding is significant for institutional planning: it establishes that the architecture is not merely adequate for small-staff deployments but is capable of absorbing university-wide scaling events driven by rapid student onboarding, without advance capacity planning or infrastructure investment beyond the managed service subscription already in place.

The broader contribution of this work is as an institutional innovation case study for Indonesian higher education. UNES Hebat demonstrates that a small development team operating within the resource constraints of a mid-tier private Indonesian university can deploy attendance governance infrastructure of a reliability, security, and scalability standard comparable to commercial enterprise solutions, by making disciplined use of managed cloud services and by locating security enforcement at the database layer rather than the client layer. The system's open architectural disclosure, subject to the responsible semi-open privacy posture described in this paper, is intended to provide a replicable model for Indonesian and other developing-country universities facing similar administrative digitization challenges.

Future development of UNES Hebat will prioritize three directions: integration of a lightweight on-device face detection model to reduce reliance on manual photo review; migration of session token storage from browser storage to HttpOnly cookies; and the addition of server-side GPS coordinate plausibility analysis to provide an additional layer of mock location detection. Each of these improvements can be implemented within the existing cloud-native architecture without changes to the core validation logic described in this paper.

Author Contributions:
Conceptualization, I.A.I., S.M., D.P.P. and M.T.M.; methodology, I.A.I.; software, I.A.I.; validation, I.A.I., S.M., D.P.P. and M.T.M.; formal analysis, I.A.I.; investigation, I.A.I.; resources, I.A.I., S.M. and D.P.P.; data curation, I.A.I.;

writing—original draft preparation, I.A.I.; writing—review and editing, S.M., D.P.P. and M.T.M.; visualization, I.A.I.; supervision, S.M., D.P.P. and M.T.M.; project administration, I.A.I. All authors have read and agreed to the published version of the manuscript.

Funding:
This research received no external funding. System development and cloud hosting costs were borne through the authors' Irfan Ananda Ismail personal resources.

Institutional Review Board Statement:
Not applicable. This study describes a technical system and its operational performance data; it does not involve human subject research requiring ethical committee review beyond the institutional approval already granted for the UNES Hebat deployment as part of the university's administrative digitization program.

Informed Consent Statement:
Not applicable.

Data Availability Statement:
The operational attendance data described in this paper is institutional data owned by Universitas Ekasakti and is not publicly available due to personnel privacy considerations. Aggregate statistical data supporting the findings of this study are available from the corresponding author upon reasonable request. The application architecture, as described in this paper, is sufficiently specified for independent reimplementation.

Acknowledgments:
The authors thank the administrative staff of Universitas Ekasakti for their commitment to the system adoption process and for the operational feedback that shaped each successive version of the application. The authors also acknowledge the 849 KKN student participants whose engagement during the July 2026 orientation provided an unplanned but invaluable stress test of the system's scaling architecture.

Conflicts of Interest:
The authors declare no conflicts of interest.

ABBREVIATIONS

The following abbreviations are used in this manuscript:

API	Application Programming Interface
BAN-PT	Badan Akreditasi Nasional Perguruan Tinggi (National Accreditation Board)
CDN	Content Delivery Network
DDoS	Distributed Denial of Service
GPS	Global Positioning System
KKN	Kuliah Kerja Nyata
PaaS	Platform as a Service
PWA	Progressive Web Application
API	Application Programming Interface
RLS	Row Level Security
RPC	Remote Procedure Call
TLS	Transport Layer Security
UNES	Universitas Ekasakti
WIB	Waktu Indonesia Barat
XSS	Cross-Site Scripting

REFERENCES

[1]. Nallusamy, C., Gowri, S., & Hari Priya, V. (2024). GPS using attendance tracking system. In Challenges in Information, Communication and Computing (pp. 1-10). CRC Press. https://doi.org/10.1201/9781003559085-19.

[2]. Chiang, T.-W., Yang, C.-Y., Chiou, G.-J., Lin, F.-Y. S., Lin, Y.-N., Shen, V. R. L., & Lin, C.-Y. (2022). Development and Evaluation of an Attendance Tracking System Using Smartphones with GPS and NFC. Applied Artificial Intelligence, 36(1), 2083796. https://doi.org/10.1080/08839514.2022.2083796

[3]. Kamelia, D., & Hamidi, H. (2018). Real-Time Online Attendance System Based on Fingerprint and GPS. International Journal of Advanced Computer Science and Applications, 9(11), 380-386..

[4]. Nguyen, V. D., et al. (2022). Internet of Things-Based Intelligent Attendance System with Face Recognition and Anti-Spoofing. Electronics, 11(19), 3151. https://doi.org/10.3390/electronics11193151

[5]. Alghushami, A. H. (2020). Factors Influencing Cloud Computing Adoption in Higher Education Institutions. Applied Sciences, 10(22), 8098. https://doi.org/10.3390/app10228098

[6]. Eweoya, I., et al. (2025). Design and Implementation of a University Attendance Management System Using Geo-Fencing. African Journal of Computing and ICT, 18(1), 1-15

[7]. Min-Allah, N., & Alrashed, S. (2020). Smart campus—A sketch. Sustainable Cities and Society, 59, 102153. https://doi.org/10.1016/j.scs.2020.102153

[8]. Akbari, T. T., et al. (2022). Higher education digital transformation implementation in Indonesia. Jurnal Komunikasi dan Kajian Media, 6(1), 1-15.

[9].Babatunde, A. N., Oke, A. A., Babatunde, R. S., Ibitoye, O., & Jimoh, E. R. (2020). Mobile based student attendance system using geo-fencing with timing and face recognition. Journal of Information Technology and Software Engineering, 10(3).