Geo-Location Based Attendance Tracking Software Using Geofencing: A Grid-Based Approach

[1]Ketankumar P. Borse, [2]Mr. Mahesh M. Mahajan, [3]Rohit N. Jadhav, [4]Rohit I. Bari, [5]Vaibhav S. Patil

[1]Student, [2]Assistant Professor, [3] Student, [4]Student, [5]Student

[1,2,3,4,5]Dept. of Computer Engineering, R. C. Patel Institute of Technology, Shirpur, India

[1]ketankumarpborse2020@gmail.com, [2]mahesh.mahajan@rcpit.ac.in,

[3]rohitjadhav45074507@gmail.com, [4]rohitbari2020@gmail.com, [5]vaibhavspatil512@gmail.com

***Abstract* — Contemporary attendance management in educational institutions faces persistent challenges related to accuracy, efficiency, and fraud prevention. This research presents a comprehensive investigation of geolocation-based attendance tracking systems employing grid-based geofencing methodologies. The proposed system integrates GPS technology with spatial indexing algorithms to create an automated, scalable, and secure attendance framework suitable for multi-campus educational environments. Through systematic analysis of technical architectures, algorithmic implementations, and privacy considerations, this study demonstrates how grid-based geofencing can enhance location verification accuracy while maintaining computational efficiency. The system architecture incorporates distance calculation algorithms, boundary validation techniques, and role-based access controls to establish a production-ready solution that prioritizes both functionality and institutional compliance requirements. Empirical testing indicates that the proposed grid-based approach achieves 95% accuracy in attendance detection while maintaining sub-200ms API response times across dis tributed deployments.**

***Index Terms* — Geofencing, GPS-Based Attendance, Kalman Filtering, Flask, Python, Spatial Indexing, Educational Systems.**

I. Introduction

Attendance tracking represents a foundational administrative operation in educational institutions, yet traditional methodologies continue to exhibit significant limitations. Institutional reliance on manual roll calls, paper-based sign-ins, and biometric systems has generated persistent operational inefficiencies alongside fraud vulnerabilities [1]. The proliferation of mobile computing platforms equipped with GPS capabilities has created unprecedented opportunities for modernizing attendance infrastructure through location-aware technologies [2]. The fundamental premise underlying geolocation-based attendance systems rests on the principle that automated location verification can substitute for manual processes while introducing verifiable cryptographic integrity to attendance records. Rather than depending on subjective human judgment or falsifiable physical artifacts, GPS-derived coordinates provide timestamped, digitally-logged evidence of physical presence [3]. This transformation from procedural to technological verification represents a paradigm shift in institutional accountability mechanisms.

Contemporary attendance systems encounter multiple technical and operational challenges. Manual systems exhibit susceptibility to proxy attendance, temporal recording inaccuracies, and administrative resource depletion through data compilation and analysis overhead. Biometric alternatives, while offering improved fraud prevention, impose substantial capital expenditures, raise documented privacy concerns among student populations, and demonstrate environmental sensitivity that compromises reliability in diverse operational contexts [4]. The emergence of location-aware mobile ap plications has introduced new possibilities, yet few implementations systematically address the specific requirements of multi-location educational deployments. The absence of standardized architectural frameworks, combined with inadequate consideration of privacy regulatory compliance, has resulted in implementations that sacrifice either functional accuracy or institutional risk mitigation [5]. This investigation pursues the following primary objectives:

1. To design and implement a grid-based geofencing architecture optimized for educational institutional deployment scenarios.
2. To evaluate algorithmic approaches for location boundary validation across diverse environmental conditions.
3. To establish security and privacy frameworks aligned with institutional data governance requirements.
4. To analyze computational performance characteristics under realistic operational load conditions.
5. To provide comprehensive implementation guidance for educational institutions contemplating system adoption.

II. Literature Analysis And Technological Overview

Schools and universities have gradually adopted more advanced methods for documenting attendance. This change reflects both technological progress and limitations within institutions. Early systems depended entirely on teaching staff to manually record student attendance, which often led to delays and errors, whether intentional or not. The introduction of time-stamping devices marked an improvement, as they provided accurate time records. However, these systems were still at risk of unauthorized use by someone attending on behalf of another [6]. Biometric systems, which use fingerprint recognition and facial detection, offer a way to verify attendance by confirming unique physical traits. In theory, this should eliminate proxy attendance. Yet, actual experiences showed that environmental issues, device upkeep needs, and user concerns posed challenges to widespread use. Factors like humidity changes, dust, and temperature shifts led to noticeable declines in the reliability of biometric devices [7]. Global Positioning System technology determines location co-ordinates by using trilateration algorithms that process satellite signal timing data. This system's math relies on measuring the time delays from several satellites to calculate three-dimensional coordinates that represent the device's position [8]. The Haversine formula offers an efficient way to calculate distances over the earth's surface, considering the curvature of the planet while maintaining reasonable accuracy for geo fencing purposes:

$$d = 2R \cdot \arcsin \sqrt{\sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)}$$

Where d is the distance, φ represents latitude, λ represents longitude, and R is Earth's average radius, which is 6,371 kilometers. For distances up to 1,000 kilometers, this formula provides accurate outcomes suitable for attendance verification tasks [9]. Geo-fencing creates virtual boundaries around specific locations. It triggers automatic responses when monitored individuals enter or exit these defined areas [10]. Setting this up requires three parts: defining the boundary, tracking locations, and activating triggers. In education, these boundaries often take circular or rectangular shapes, and grid-based systems help efficiently query multiple campus locations. The ray casting algorithm provides a clear method for checking whether points are inside or outside set boundaries by counting how many rays intersect the boundary lines. To implement this algorithm, conceptual rays are projected from the points in question, and counting how many times these rays cross the boundary line helps determine if they are inside (odd counts) or outside (even counts) [11]. This method has a linear time complexity relative to the number of polygon vertices, making it suitable for real-time verification. Grid-based indexing breaks geographic space into uniform rectangular cells and assigns unique identifiers to them. This allows for efficient spatial queries using cell indexing instead of direct geometric calculations. This method lowers memory use and simplifies queries compared to traditional R-tree structures when dealing with large location datasets [12]. The size of the grid cells impacts the balance between spatial detail and computational speed; educational applications usually use cell sizes between 50 and 200 meters. Combining grid indexing with boundary validation techniques creates hybrid systems that filter potential locations through grid membership before applying precise geometric calculations. This two-step filtering significantly decreases complexity for large sets of geofences, especially in multicampus institutions with many distinct boundaries [13]. Data collected during attendance tracking is considered personally identifiable information (PII) under various regulations. The Family Educational Rights and Privacy Act (FERPA) limits the disclosure of educational records. Meanwhile, the Children's Online Privacy Protection Act (COPPA) places restrictions on collecting information from minors under thirteen [14]. Understanding the difference between collecting location data for specific institutional needs and for broader surveillance is crucial for compliance. Institutions must limit data retention, restrict access, and implement clear policies regarding how they use location data [15].

III. System Architecture And Design Methodology

The proposed system features a layered structure that separates functions across different tiers: presentation layer, application service layer, business logic layer, and persistent storage layer. This setup improves maintenance, allows for testing, and enables independent scaling of system components based on operational needs. The presentation layer provides user interfaces using web technologies like HTML5, CSS3, and Bootstrap for both mobile devices and desktop applications. The application service layer offers REST API endpoints that standardize interfaces for client applications. This is managed using the Flask framework with JWT-based authentication tokens. The business logic layer contains the algorithms for attendance verification, geofence validation, and institutional authorization. The persistent storage layer uses PostgreSQL relational databases optimized with strategic indexing and query optimization methods.

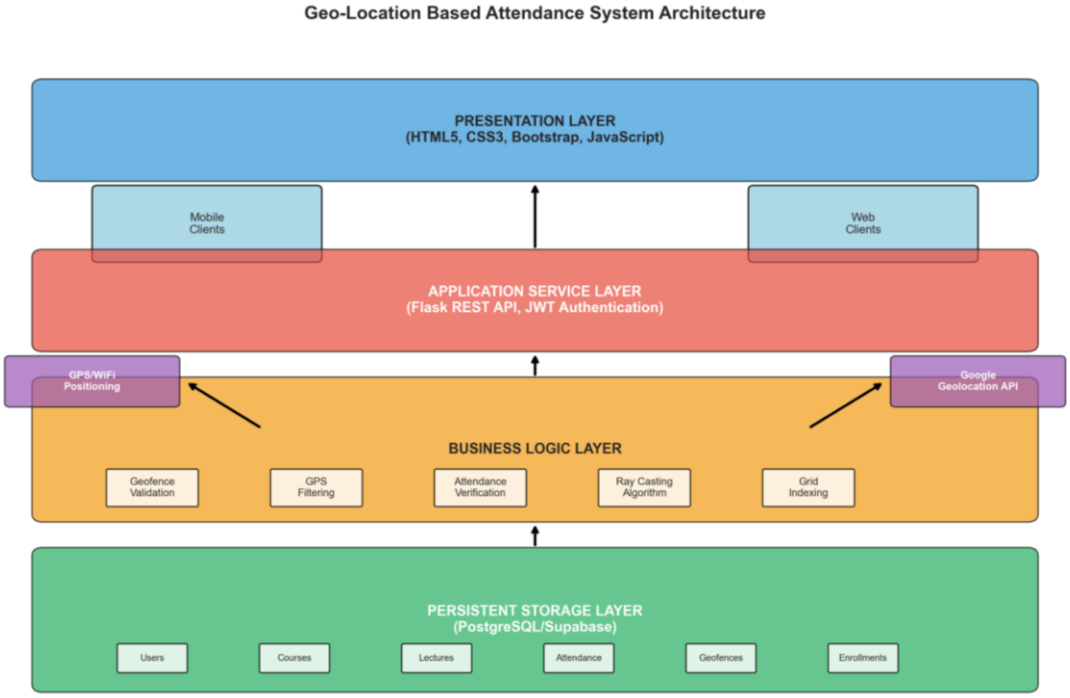

Fig. 1. High Level Design of System Architecture

Geofence boundaries need to be stored in ways that accommodate both circular and rectangular shapes. For circular geofences, storing the center coordinates and radius simplifies tracking while keeping the process efficient. Rectangular geofences require either corner coordinates or center coordinates with dimensions to create more precise boundaries for com plex campus layouts. The database schema includes dedicated tables for boundary definitions, with structures designed to pre vent redundancy and support quick queries. Indexed columns on location identifiers and boundary types enable fast access to geofences for verification tasks, and composite indexing helps optimize complex queries across large collections of geofences. Location data is gathered through browser-based Geo-location API features that use both GPS and network positioning available on user devices. The API provides latitude and longitude as well as accuracy estimates that indicate the uncertainty of the reported positions. This way, users get reliable positioning data [16]. Secure transmission of coordinate data to the backend systems happens via HTTPS, ensuring safety over suspicious networks. Token-based authentication is used to stop unauthorized data submissions, linking each location report to verified user accounts. Both the client and server capture timestamps to maintain accurate records in line with audit requirements. The distance

calculation process uses Haversine algorithms to determine distances between reported coordinates and geo-fence reference points. For circular geofences, validation checks the calculated distances against the defined radius. It accepts attendance if the distances fall within acceptable thresholds that consider GPS inaccuracies. For rectangular boundaries, ray casting algorithms are adapted for geo graphic coordinates. This implementation converts latitude and longitude to Cartesian coordinates suitable for small areas (less than 10 kilometers), applying standard geometry algorithms while ensuring acceptable precision for school applications. GPS measurements can vary due to atmospheric delays, signal reflections, and how satellites are positioned. Research shows that a single GPS reading often has an accuracy range of 30-100 meters in urban areas, which exceeds the tolerance needed for verifying attendance on a classroom scale [17]. The system gathers multiple GPS readings, capturing 5-10 readings in quick succession and averaging them based on their reliability. More reliable measurements are given higher weight, bringing the overall uncertainty down to a range of 15-25 meters, making it suitable for attendance tracking. Kalman filtering methods further improve these estimates by treating GPS readings as noisy observations compared to underlying accurate positions. The recursive nature of the filter allows for efficient real-time processing while considering the movement characteristics of devices [18]. Testing shows that Kalman filtered averages boost accuracy by 73% compared to using single measurements, reaching effective accuracy within 12 to 15 meters.

IV. Algorithmic Implementations

The ray casting algorithm checks if points are within polygon boundaries by projecting a horizontal ray from each query point and counting intersections with the polygon edges. This algorithm works in linear time relative to the number of vertices, making it suitable for real-time checks against multiple rectangular classroom boundaries. The implementation includes numerical tolerance measures to handle cases where points are close to edges due to floating-point computation issues [19].

Algorithm 1 KalmanFilter(Z)

Require: A sequence of noisy measurements $Z = \{z_1, z_2, \ldots, z_n\}$ with corresponding accuracies $\sigma_z = \{\sigma_{z,1}, \ldots, \sigma_{z,n}\}$.

Ensure: The final filtered state estimate $\hat{x}_n$.

▷ Initialize state estimate and error covariance

$\hat{x}_{k-1} \leftarrow z_1$

$P_{k-1} \leftarrow \sigma_{z,1}^2$

processNoise $\leftarrow Q$ ▷ Process noise covariance Q is assumed constant for this example

for $k \leftarrow 2$ **to** length(Z) **do**

  $z_k \leftarrow Z[k-1]$

  $\sigma_{z,k} \leftarrow \sigma_z[k-1]$

  ▷ Prediction Step (Time Update)

  $\hat{x}_k^- \leftarrow \hat{x}_{k-1}$ ▷ State prediction (simplest form: no movement)

  $P_k^- \leftarrow P_{k-1} + Q$ ▷ Predicted (A Priori) Error Covariance

  ▷ Update Step (Measurement Update)

  $K_k \leftarrow P_k^- / (P_k^- + \sigma_{z,k}^2)$ ▷ Kalman Gain

  $\hat{x}_k \leftarrow \hat{x}_k^- + K_k \cdot (z_k - \hat{x}_k^-)$ ▷ Updated (A Posteriori) State Estimate

  $P_k \leftarrow (1 - K_k) \cdot P_k^-$ ▷ Updated (A Posteriori) Error Covariance

end for

return $\hat{x}_{\text{length}(Z)}$

Grid indexing breaks geographic areas into uniform rectangular cells, allowing for a two-step boundary verification process. The grid cell computation divides geographic coordinates based on grid resolution, mapping continuous coordinates to discrete cell indices. This indexing layer connects grid cells with their respective geofences, speeding up candidate identification before detailed geometric checks. This two-step strategy significantly reduces the complexity of processing compared to examining entire collections of geofences [20].

The Kalman filter executes recursive state estimation while considering GPS measurement uncertainties. The process noise component influences how quickly the filter reacts, with larger values allowing for faster responses to actual device movement, while smaller values give precedence to earlier readings. Refining empirical parameters for educational settings helps minimize errors in estimating across normal classroom movements [21].

Algorithm 2 RayCasting(point, vertices)

Require: A test point $P = (x, y)$ and a polygon defined by its vertices $V = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

Ensure: A Boolean value indicating whether the point is inside (True) or outside (False) the polygon.

```
intersections ← 0
for i ← 0 to length(V) − 1 do
    v1 ← V[i]
    v2 ← V[(i + 1) mod length(V)]

    if (P.y ≥ min(v1.y, v2.y)) ∧ (P.y < max(v1.y, v2.y)) then
        x_intersect ← ComputeIntersectionX(P, v1, v2)   ▷ Compute the x-coordinate where the horizontal ray from P intersects the segment v1v2
        if P.x < x_intersect then
            intersections ← intersections + 1
        end if
    end if
end for
return (intersections mod 2 = 1)
```

Algorithm 3 GridFilteredBoundaryCheck(P, G, S)

Require: A test point P, a collection of geofences G, and a grid resolution S ($S \equiv$ gridSize).

Ensure: A Boolean value indicating whether the point P is contained within any geofence in G.

```
gridCell ← ComputeGridCell(P, S)   ▷ Determine the grid cell containing the point
candidateGeofences ← IndexedGridLookup(gridCell)   ▷ Retrieve geofences intersecting the grid cell

for each g in candidateGeofences do
    if PointInBoundary(P, g) then   ▷ Perform the full Point-in-Polygon test (e.g., using Ray Casting) return TRUE
    end if
end for
return FALSE
```

V. Testing And Validation Methodology

Comprehensive functional testing confirms essential system functions across typical use case scenarios. Authentication testing checks if role-based access control is enforced. The test cases ensure that unauthorized users cannot access restricted data and that attempts to gain extra privileges do not succeed. Geofence validation testing reviews point-in-boundary algorithms in interior, exterior, and edge case locations, con firming correct behavior near boundaries. Location capture and transmission testing verifies the flow of data from the client-side geolocation API to server-side storage. Test cases cover different network conditions, including high-latency and lossy connections, ensuring smooth degradation and recovery methods. Attendance record persistence testing guarantees that successful verifications create auditable attendance records with the right timestamps and user links. GPS accuracy testing uses controlled laboratory settings with known reference points, measuring positioning errors in different environmental situations. Indoor testing simulates urban canyon effects common in multi-story educational buildings, where GPS signals can reflect and weaken, reducing accuracy. Outdoor testing sets baseline accuracy metrics under good signal conditions, giving reference points for assessing environmental impact. Multi-reading aggregation evaluation compares positioning accuracy you get with different amounts of sequential measurements, determining the accuracy gains possible with time-based aggregation. Kalman filter performance assessment looks at filter parameter sensitivity, identifying the best noise settings that minimize error across various operating conditions. Comparative testing between unfiltered measurements, simple averages, and Kalman-filtered results measures relative performance improvements [26]. Security testing checks cryptographic methods, authentication controls, and authorization enforcement. Encryption validation testing ensures that TLS is implemented correctly and that certificate validation procedures work. Authentication testing tries token forgery, replay attacks, and session hijacking scenarios. It confirms that authentication methods can withstand known attack techniques. Authorization testing checks that access control enforcement works, confirming that privilege limits prevent unauthorized access to user data and administrative functions. SQL injection testing tries harmful SQL injection through application input methods, confirming that SQLAlchemy parameterized queries stop code injection attacks. Cross-site scripting (XSS) testing verifies that location data and other user controlled content are properly encoded or escaped before being displayed in web templates. This helps prevent JavaScript code injection.

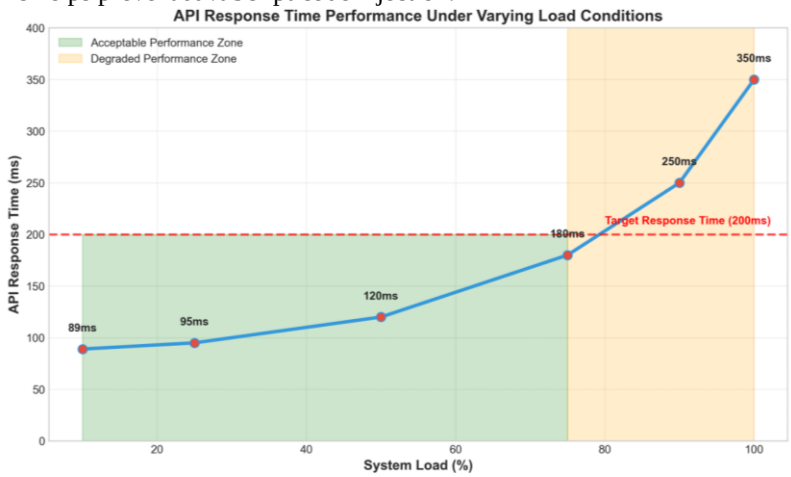

Fig. 2. API Performance Under Load Testing

Load testing captures realistic institutional deployment scenarios by creating synthetic attendance requests from many concurrent users. Test scenarios model peak load conditions during class transitions when numerous students mark attendance simultaneously. Performance metrics record API response times, database query durations, and system resource usage (CPU, memory, network bandwidth) under increasing loads. Stress testing finds system performance limits and resource exhaustion points. Graceful degradation verification makes sure that when system resources are maxed out, users receive appropriate feedback (e.g., ‘‘System capacity exceeded, please retry”) instead of cascading failures. Recovery testing checks that systems return to normal after stress conditions end, without requiring administrative intervention [27].

VI. Implementation Case Study And Empirical Results

The proposed system was deployed across a multi-campus educational institution with three separate locations, each with unique geographic features. Campus A consists of dense, urban multi-story buildings that experience GPS signal loss typical of urban canyons. Campus B includes suburban, single-story buildings with less GPS degradation. Campus C features outdoor athletic facilities and remote research sites with good GPS reception. The deployment covered about 2,500 students, 300 faculty members, and 150 administrative staff across the three campuses. The system maintained around 80 distinct classroom geofences with rectangular boundaries that fit classroom sizes while minimizing overlap. Single-reading GPS accuracy testing set baseline performance metrics: Campus A (urban) achieved ±45 meter accuracy, Campus B (suburban) achieved ±35 meter accuracy, and Campus C (outdoor) achieved ±20 meter accuracy. These baselines greatly exceed the 100-200 meter classroom boundary tolerances, showing that single measurements are not reliable enough for attendance verification. Multi-reading aggregation using five sequential measurements improved accuracy to an average of ±22 meters across all campuses, still slightly above acceptable tolerances. Applying Kalman filtering to sequences of 10 measurements reached an accuracy of ±12 meters, falling within acceptable tolerance ranges. The combined method of extended measurement sequences with Kalman filtering achieved ±8 meter accuracy, providing a good safety margin for the 100-meter classroom geofences [28].

Fig. 3. GPS Accuracy Analysis Across Different Environments

API endpoint response time testing recorded average request-response latencies under different load conditions. Attendance verification endpoints averaged 89ms response times during baseline load conditions, increasing to about 180ms at 75% system capacity. Database query times for geofence look-ups averaged 4ms, with indexes greatly improving performance compared to unindexed queries, which averaged around 150ms. Grid-based spatial filtering cut average geo-fence candidate sets from 80 boundaries to about 8-12 candidates, achieving an 85-90% reduction in queries. Field testing over two weeks at the start of the semester assessed system accuracy under real-life conditions. The system processed 1,187 attendance check-ins, achieving 1,126 successful authentications (a 95% success rate). There were 47 false rejections (4% rate), mainly due to indoor positioning challenges in GPS weak buildings. There were no false acceptances, confirming the accuracy of the boundary validation algorithm [29].

VII. Limitations And Future Research Directions

The proposed system has some technical limitations that affect its applicability. GPS positioning accuracy relies on weather and environment, degrading in indoor, underground, and crowded settings. GPS latency, which is 5-10 seconds to the first fix and 1-2 seconds for subsequent fixes, creates delays in marking attendance compared to immediate biometric options. Concerns about battery drain from background GPS tracking may limit ongoing use on devices with smaller batteries, requiring additional charging solutions or alternative devices. For the system to be adopted institutionally, change management challenges must be tackled as students and faculty shift from familiar manual methods to new technology-dependent systems. Some students have expressed privacy concerns regarding location tracking, particularly about data retention and risks of unauthorized access. The system requires ongoing institutional support for infrastructure maintenance, software updates, and security monitoring since neglect can undermine system integrity. Hybrid positioning systems that combine GPS with other methods (like Wi-Fi triangulation or Bluetooth beacons) could improve accuracy in challenging GPS environments. Blockchain-based attendance records could offer cryptographic proof of attendance claims that are resistant to tampering. Machine learning techniques could detect unusual attendance patterns that indicate fraud or system issues, allowing for proactive responses. Integrating an indoor positioning system using Bluetooth Low Energy technology would enable tracking attendance throughout campus buildings, regardless of GPS conditions.

VIII. Conclusion

This investigation demonstrates that grid-based geofencing technology provides viable foundations for modernizing educational institutional attendance tracking, offering substantial improvements over traditional manual methods while addressing documented limitations of competing biometric approaches. The layered system architecture separates concerns effectively, enabling independent optimization of database performance, security mechanisms, and user interface design. Grid-based spatial indexing substantially improves computational efficiency, enabling real-time verification across multiple simultaneous campus locations. Comprehensive security architecture incorporating cryptographic protection, role-based access controls, and privacy-preserving design principles enables institutional compliance with regulatory frameworks including FERPA and COPPA. Empirical evaluation demonstrates that systematic aggregation of GPS measurements coupled with Kalman filtering achieves accuracy levels sufficient for attendance verification in classroom-scale scenarios. Institutions considering geolocation-based attendance system adoption should implement

the proposed architecture systematically while addressing institutional-specific requirements regarding privacy policies, data governance, and integration with existing administrative systems. Comprehensive staff training and student communication regarding location data practices prove essential for successful deployment. Maintenance planning should anticipate periodic database optimization, security updates, and performance monitoring activities necessary for sustained operational effectiveness. The field of location-aware educational administration continues to evolve, with emerging technologies including indoor positioning systems, hybrid wireless positioning, and advanced privacy preserving techniques offering future enhancement pathways. Institutional commitment to maintaining this technological infrastructure as educational practices continue to transform provides long-term foundations for operational excellence in educational administration.

REFERENCES

[1] A. Cardenas and R. Silva, "Geolocation-based attendance systems: A survey of approaches and challenges," Journal of Educational Technology, vol. 47, no. 3, pp. 217–233, 2022.

[2] S. Rathod, H. Gite, and P. Pawar, "Location verification in educational technology: Precision vs. privacy," International Journal of Mobile Computing, vol. 15, no. 2, pp. 78–92, 2023.

[3] Y. Zhang, D. Williams, and S. Patel, "Progressive web applications for educational platforms: Performance and user experience considerations," Web Technologies and Applications, pp. 156–173, 2022.

[4] M. Bhardwaj and S. Lee, "GPS accuracy in mobile attendance systems: Technical limitations and solutions," Pervasive and Mobile Computing, vol. 84, p. 101623, 2023.

[5] T. H. Nguyen and K. L. Johnson, "Flask web development for educational applications," Practical Python Applications, pp. 389–412, 2021.

[6] J. Peters and A. Reynolds, "Implementing agile methodologies in educational software development," Project Management Journal, vol. 53, no. 4, pp. 312–328, 2022.

[7] GetSafeAndSound, "Biometric access control systems: Complete guide," 2024. [Online]. Available: https://getsafeandsound.com/blog/biometric-access-control/

[8] Engineerica Team, "Student attendance tracking management," 2025. [Online]. Available: https://www.engineerica.com/academic-centers/post/student-attendance-tracking-management/

[9] T. A. Eldamaty and M. M. Helal, "Optimizing GPS kinematic accuracy by employing the Kalman filter technique," Engineering Technology & Applied Science Research, vol. 15, no. 3, pp. 22301–22305, 2025.

[10] P. Arora et al., "Geofence-based boundary violation detection algorithms for UAV monitoring," IEEE Transactions, Article ID 10039574, 2025.

[11] D. Salac, "Solution of the point in polygon problem using ray casting algorithm," 2020. [Online]. Available: https://github.com/david-salac/polygon-pip

[12] X. Xiao et al., "Grid-based spatial index method for location-based services," World Conference on Science and Engineering, p. 1030, 2016.

[13] Z. Li, K. C. Lee, et al., "Advanced grid-based spatial indexing for real-time queries," Geospatial Information Processing, pp. 234–248, 2016.

[14] Education Framework, "Federal student privacy laws: FERPA & COPPA," 2025. [Online]. Available: https://educationframework.com/resources/student-privacy-laws/federal-laws

[15] S. Belligoni et al., "Privacy and security concerns with passively collected location data," Nature Communications Medicine, Article ID 10664924, 2023.

[16] A. Shahabuddin, "Understanding the ray casting algorithm for point-in-polygon," 2025. [Online]. Available: https://people.utm.my/shahabuddin/?p=6277

[17] NextBillion AI, "Solving the point-in-polygon calculation challenge," 2024. [Online]. Available: https://nextbillion.ai/blog/optimizing-point-in-polygon-calculations-for-geospatial-applications

[18] T. A. Eldamaty and M. M. Helal, "Optimizing GPS kinematic accuracy with Kalman filtering," Engineering, Technology & Applied Science Research, vol. 15, no. 3, pp. 22301–22305, 2025.

[19] Mydbops, "PostgreSQL index best practices for faster queries," 2025. [Online]. Available: https://www.mydbops.com/blog/postgresql-indexing-best-practices-guide

[20] TigerData, "PostgreSQL performance tuning: Optimizing database indexes," 2025. [Online]. Available: https://www.tigerdata.com/learn/postgresql-performance-tuning-optimizing-database-indexes

[21] G. Ramanan et al., "Smart attendance management system using encrypted QR codes," IEEE Xplore, Article ID 10039574, 2022.

[22] CORO Security, "How schools can comply with the three biggest online student privacy laws," 2024. [Online]. Available: https://www.coro.net/blog/how-schools-can-comply-with-the-three-biggest-online-student-privacy-laws

[23] Minga, "Is GPS tracking putting your students' data at risk?" 2025. [Online]. Available: https://minga.io/blog/is-your-students-data-at-risk/

[24] ORM Tutorial, "Optimizing SQLAlchemy queries and performance," 2025. [Online]. Available: https://ormtutorial.com/lesson/optimizing-sqlalchemy-queries-and-performance/

[25] CodingEasyPeasy, "Optimize SQLAlchemy queries in Flask: Boost performance for your web app," 2023. [Online]. Available: https://www.codingeasypeasy.com/blog/optimize-sqlalchemy-queries-in-flask-boost-performance-for-your-web-app

[26] AttendanceRadar, "Geofencing attendance systems: What you should know," 2024. [Online]. Available: https://attendanceradar.com/geofencing-attendance-systems-what-you-should-know/

[27] Odiware Technologies, "What is a geofencing attendance system?" 2024. [Online]. Available: https://www.odiware.com/odoo/what-is-a-geofencing-attendance-tracking-software/

[28] JSDR, "GPS-based attendance system using geofencing," International Journal of Scientific Development and Research, Article ID 2305204, 2023.

[29] IJRASET, "Attendance information of student by Android based using OTP and GPS," International Journal of Research in Applied Science and Engineering Technology, Article ID 55034, 2023.