ClassConnect: A Real-Time Attendance Management System Empowering Education Through Broadcasting Technology

Urmil Pawar  
Computer Engineering  
Thakur College Of Engineering and Technology  
Mumbai, India  
urmilpawar2003@

gmail.com

Prabhat Pankaj  
Computer Engineering  
Thakur College Of Engineering and Technology  
Mumbai, India  
prabhatpankajofficial@gmail.com

Shlok Prajapati  
Computer Engineering  
Thakur College Of Engineering and Technology  
Mumbai, India  
shlokprajapati2003@

gmail.com

Ms. Vaishali Nirgude  
Computer Engineering  
Thakur College Of Engineering and Technology  
Mumbai, India vaishali.nirgude@thakureducation.org

***Abstract - Attendance monitoring is a crucial aspect of educational and organizational environments, ensuring accountability and efficient management. Traditional attendance methods are often time taking and vulnerable to proxies. This paper proposes an app called ClassConnect which is accessible, cost effective, fast and fault tolerant method of taking attendance by utilizing GPS and biometric which is present in all the phones making it accessible and cost effective. By using ClassConnect app, teacher can take attendance of the whole class in a constant time (t = c) and 100% accuracy in contrast to other methods e.g., installing a fingerprint scanner which as much time as the number of students presents while being fault tolerant. The proposed method is implemented using the biometric authentication feature along with the GPS technology, both present in every modern smart phone. Considering the heavy server usage the load balancing of the servers is done using least connections and round robin algorithm. WebSocket server is created and hosted for creation of attendance sessions.***

***Index Terms - Attendance tracking methodologies, GPS authentication, Biometric authentication, Automated attendance, Real-time tracking attendance***

# INTRODUCTION

Attendance management is a crucial aspect of organizational operations, ensuring compliance, participation tracking, and accountability. Traditional methods, such as manual roll calls or paper-based systems, are prone to inefficiencies, errors, and manipulation, often resulting in distractions, missed records, and proxy attendance [1]. These shortcomings undermine the reliability of attendance tracking, particularly in large educational or organizational settings where manual processes consume significant time and resources [2, 3].

Recent advancements in technology have spurred the adoption of automated systems, including QR codes [2, 3], RFID tags [4], and biometric authentication [1, 5]. While QR code-based systems offer simplicity, they often rely on static codes vulnerable to misuse or require frequent regeneration, complicating scalability [2, 3]. RFID and Bluetooth Low Energy (BLE) beacon systems, though efficient, demand specialized hardware installations, increasing infrastructure costs [4, 5]. Biometric methods, such as fingerprint and facial recognition, enhance security but face challenges like environmental sensitivity (e.g., lighting variations for facial recognition [5]) and hygiene concerns with shared hardware [1]. GPS-based solutions, while promising for location verification, struggle with accuracy in indoor environments or diverse settings like online classrooms [3, 7].

In this context, our paper surveys existing attendance methodologies and introduces a novel solution leveraging **GPS and smartphone-based biometric authentication** to address these limitations. Modern smartphones, equipped with built-in biometric sensors (fingerprint, facial recognition) and GPS capabilities, provide a ubiquitous, cost-effective platform for secure attendance tracking [7]. By integrating location verification with biometric identity checks, our system eliminates the need for external hardware, reduces proxy risks, and ensures real-time accuracy. Additionally, to mitigate server load during peak usage, we implement load balancing via *least connections* and *round-robin* algorithms, coupled with a WebSocket server for seamless attendance session management.

This approach not only streamlines attendance marking in constant time—irrespective of class size—but also enhances fault tolerance and accessibility, addressing gaps in existing systems [1, 2, 5]. Our solution aims to contribute a scalable, secure, and efficient framework for modern attendance monitoring, adaptable to both physical and virtual environments.

Furthermore, security considerations play a vital role in attendance management, especially with the increasing risks of data breaches and identity fraud. While barcode and RFID-based identification methods offer automation, they introduce security vulnerabilities such as cloning, relay attacks, and data interception [8]. RFID systems, despite their efficiency in bulk identification, are susceptible to eavesdropping and unauthorized tracking, raising privacy concerns in educational and workplace environments [8]. Similarly, barcode-based approaches, particularly QR codes, face risks of tampering and phishing attacks, which can compromise attendance integrity and allow unauthorized access [8]. To address these concerns, our proposed solution capitalizes on smartphone-based biometric authentication combined with GPS tracking to create a secure, scalable, and cost-effective attendance system. Unlike traditional methods that require specialized scanning hardware or extensive manual input, smartphones provide an accessible and standardized platform with built-in security features such as encryption and biometric verification [8].

* 1. **LITERATURE SURVEY**

The literature survey explores different attendance systems that are currently in use, highlighting their advantages and limitations. Various technologies such as QR codes, face recognition, voiceprint verification, Bluetooth Low Energy (BLE) and biometrics have been analyzed for their feasibility, security, time and cost efficiency. The insights from this analysis allowed us to identify gaps in the current system and develop a fast, reliable, fault-tolerant, cost-effective solution in contrast to the reviewed methodologies.

**Table 1: Literature Survey**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref paper** | **Methodology** | **Advantage** | **Drawbacks** |
| [1]  2021 | Fingerprint-based Attendance System | * Reduced fraud, ensuring **86.76% target achievement** in attendance tracking. * **93.01% effectiveness** in adapting to electronic attendance. | * Requires hardware (fingerprint scanners) * Potential hygiene concerns fingerprint scanning |
| [2]  2021 | QR Code-based Attendance System with Face Recognition | * Cost effective as RFID readers cost **10x more** than QR scanners. * During class, teachers have **unlimited rectifications** for attendance | * Requires students to have smartphones * Dependence on internet connectivity * Potential privacy concerns with face recognition |
| [3]  2021 | QR Code-based Attendance System with GPS and mobile device verification | * Prevents attendance cheating with multiple verification factors * Easy to use and integrates with mobile devices * -Helps forecast trends | * Limited to Android devices (not compatible with other OS) * GPS sensitivity issues in some devices * Requires students to be physically present on campus |
| [4]  2022 | Bluetooth Low Energy (BLE) and Beacon-based Attendance System | * Contactless attendance system reduces the risk of disease transmission * The **high mean ratings (~9.2/10)** and **low standard deviation (<1.1)** suggest that lecturers **consistently found** the system effective * **Cronbach’s Alpha for Students:** **0.992** (Very High Reliability) | * Limited to Android-based devices * Dependence on Bluetooth connectivity * Potential privacy concerns with location tracking |
| [5]  2023 | Facial Detection and Recognition using Haar Cascade and LBPH algorithms | * LBPH face recognition algorithm achieves up to **100% accuracy**, outperforming Eigenfaces (73.3%) and Fisherfaces (36.4%)​ * Can process up to **100 samples** efficiently * -Eliminates manual errors and enhances real-time attendance tracking​ | * Requires good lighting and image quality for accurate recognition * Privacy concerns biometric data collection * Difficulty in recognizing faces with abnormalities or disguises |
| [6]  2021 | Review of advanced attendance and monitoring systems | * Reduces manual errors by automating the verification process * Allows employers to track staff attendance in real-time * Automated systems prevent **buddy punching** and fraudulent entries, ensuring **100% authentic** attendance records | * High implementation cost * Privacy concerns * Dependence on internet connectivity |
| [7]  2022 | Face Recognition and Fingerprint-based Attendance System with GPS | * Attendance confirmation via QR scan takes **less than 1 minute per student** * Prevents proxy attendance and manipulation * Uses advanced biometric algorithms like the histogram of oriented gradient | * Require smartphones with biometric sensors * Dependence on GPS for location verification * Potential privacy concerns biometric data collection |
| [8]  2021 | Comparison of Barcode vs. RFID for auto-ID security | * Passive RFID tags are up to **1,000 times cheaper** than active ones * Dynamic barcode + RFID hybrid systems greatly reduce cloning and replay attack risks * - Enhanced security measures | * Privacy concerns * Potential security vulnerabilities * Higher implementation costs |
| [9]  2016 | Voiceprint and Location-based Attendance System | * High efficiency and accuracy in attendance tracking * Prevents proxy attendance with voiceprint and location verification * Easy to use and reduces manual effort | * Requires students to have smartphones with reliable internet * Potential privacy concerns with voiceprint data * Dependence on external APIs for voice and location services |
| [10]  2023 | QR code scanning with biometric (fingerprint) and GPS verification | * Prevents cheating * Cost-efficient * Speeds up attendance marking | * Requires biometric hardware * GPS accuracy issues * Potential privacy concerns |

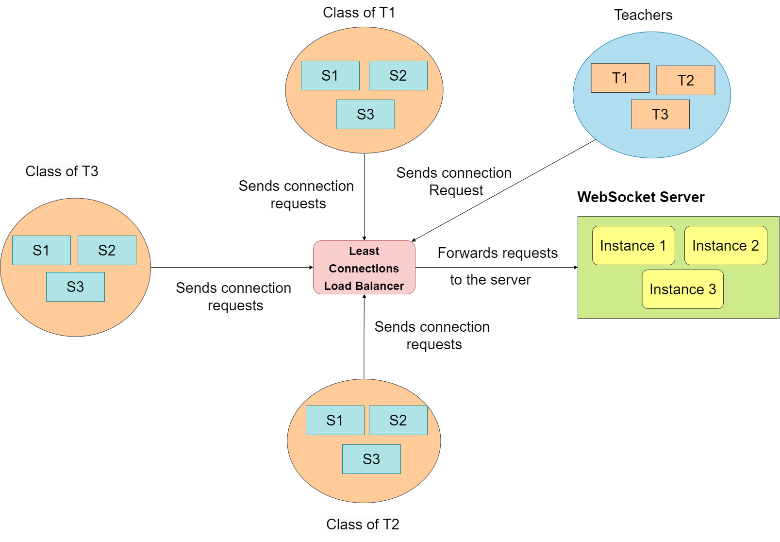
# 3.0 PROPOSED METHODOLOGY

# *3.1 System Architecture:*

Our Attendance Tracking and Management System is designed to streamline attendance marking by incorporating biometric authentication and geolocation checks.

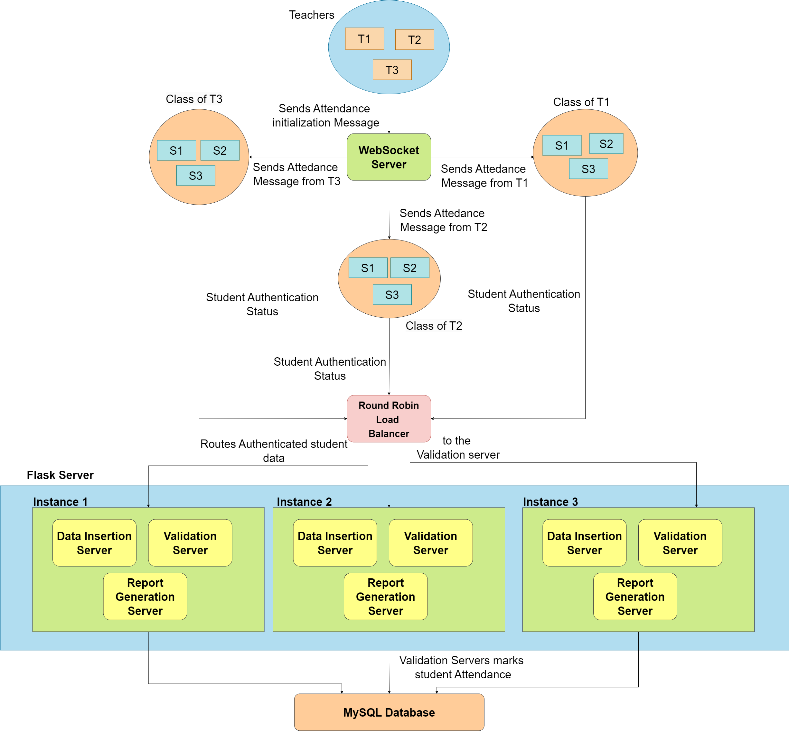
**The system consists of three main phases:**

**1. Client Connection and Session Initialization**: Here, Teachers and students connect to the WebSocket server as clients. Students cannot connect before the respective teacher joins. When the teacher gets connected a session is initialized with a map with the teacher as a key and the set of respective students as the value corresponding to the key, ensuring proper mapping between the teacher and their students.



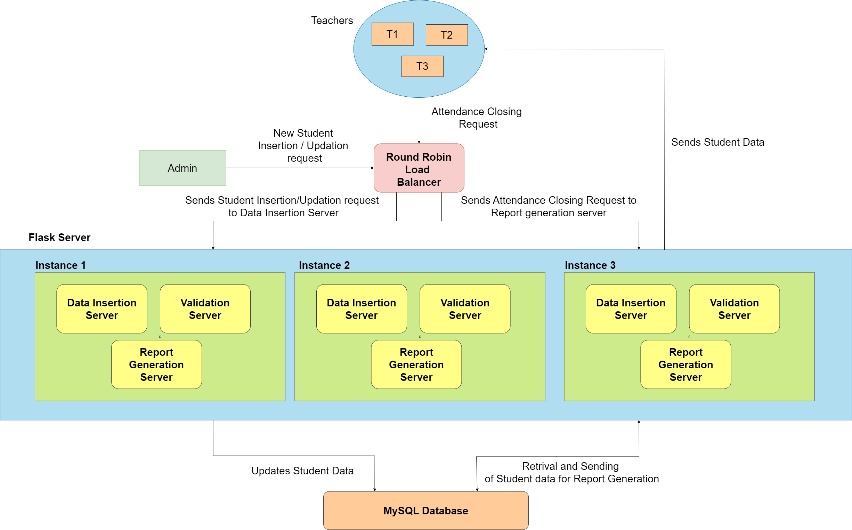
**Figure 1: Client Connection and Session Initialization**

**2. Attendance Verification and Marking**: When the teacher starts the attendance session, students undergo biometric authentication and geolocation validation. If both conditions are successfully met, the students' attendance is marked as present.



**Figure 2: Attendance Verification and Marking**

**3. Session Closure and Report Generation**: Once the attendance session is completed, the teacher can stop the session, which triggers the generation of the attendance report, summarizing the students’ attendance status.



**Figure 3: Session Closure and Report Generation**

***3.2 System Workflow:***

**3.2.1 The workflow of the Attendance Tracking and Management System is as follows:**

1. **Teacher Connection**: The teacher fills in the required information on the interface form, such as user type, name, initials, department, division, and year, and connects to the WebSocket server.
2. **Session Creation**: Upon the teacher’s connection, a new session map is created with the teacher’s initials as the key.
3. **Student Connection**: Students fill out their details, including user type, name, roll number, branch, division, year, and the teacher’s initials (who is currently taking the lecture) on their interface form. They then connect to the WebSocket server.
4. **Mapping Students to Teachers**: Students are mapped to the session based on the teacher’s initials. This ensures that each student is linked to the correct session for attendance tracking.
5. **Attendance Start**: When the teacher clicks the 'start' button, a message indicating that attendance has started is sent to the students' devices. The message also includes the teacher's location, and the session map on the WebSocket server is cleared.
6. **Student Authentication**: Upon receiving the attendance message, the students’ devices will open an interface prompting biometric authentication. Simultaneously, the system checks if the students are within a specific radius of the teacher’s location (geolocation check).
7. **Data Validation**: Students who successfully authenticate and are within the required radius will have their data sent to the 'validation server' to be stored in the database.
8. **Attendance Stop**: When the teacher clicks the 'stop' button, a request is sent to the report generation server. The server will fetch and display a detailed list of the students present in the session.

**3.2.2 Leveraging Teacher Initials and Mobile Biometric Authentication for Attendance**

**1.** **Use of Teacher Initials**: A teacher initial is simply a unique identifier derived by creating a subsequence from the teacher’s name. In cases where lectures or practicals are conducted in a batch-wise mode (with a single class split into multiple batches), the teacher remains a unique identifier. Regardless of the number of batches, the teacher is the constant entity to identify which students belong to which session, ensuring clarity.

**2.** **Authentication only Using Mobile OS Verification**: The system leverages the biometric authentication capabilities of the student’s mobile device [11]. The mobile OS verifies the student's identity, similar to the phone’s unlocking mechanism [11]. This eliminates the need for external devices, enhancing the system's ease of use.

***3.3 Mathematical Model:***

To formally define the system, we represent it using set notation:

**1. Set Definitions**

1. Let *T* represent the set of all teachers:

(i)

1. Let *L* represent the set of all sessions created by teachers:

(ii)

1. Let *S* represent the set of all students:

(iii)

1. Let *A* represent the set of students who are successfully authenticated for attendance:

(iv)

**2. Variables and Functions**

1. *t ∈ T*: A teacher in the system.
2. *s ∈ S*: A student in the system.
3. *L(t):* The lecture created by teacher t, representing a set of students present in the lecture.
4. *bio(s):*A function that returns 1 if the student successfully passes biometric authentication, otherwise 0.
5. *loc(s):* A function that determines the location of the teacher t or student s.
6. *geo(s,t):* A function that checks if the student s is within the required geolocation radius of teacher t.

(v)

where *d(loc(s), loc(t))* is the distance between the location of student s and teacher t, and r is the required radius.

1. *auth(s,t):* A function that returns 1 if student s is authenticated (biometrically and geographically) for teacher t’s session, otherwise 0.

(vi)

1. *att(s,t):* A function that represents whether a student s is marked as present in a teacher t’s session:

(vii)

**3. Conditions**

1. The distance between the student and the teacher must satisfy:

(viii)

1. The student must pass biometric authentication:

(ix)

1. If both conditions are satisfied, the student’s attendance is marked:

(x)

Otherwise:

(xi)

**4. Workflow Representation**

(xii)

***3.4 Distance Calculation :***

In our project, we utilized the Haversine formula to calculate the great-circle distance between two points (student and teacher) on the Earth's surface, represented by their geographical coordinates (latitude and longitude) [12]. The formula accounts for the spherical shape of the Earth [12].

According to [12,13], the Haversine formula calculates the distance (d) between two points on a sphere based on their latitudes and longitudes. It is defined as follows:

(xiii)

(xiv)

(xv)

Where:

→ Latitudes of the two points (in radians)

→ Longitudes of the two points (in radians)

→ Difference in latitudes (in radians)

→ Difference in longitudes (in radians)

R → Radius of the Earth (mean radius = 6,371 km)

d → Great-circle distance between the two points

**Key Aspects:**

**1. Earth’s Radius (R)**: The radius is set to 6,371,000 meters (mean radius), as the Earth's curvature significantly influences the accuracy of the distance measurement. [14]

**2. Conversion to Radians**: According to [15], the formula for converting degrees to radians is given as follows:

(xvi)

**3. Intermediate Values (a and c)**

a: Represents the square of half the chord length between the points, accounting for the spherical shape of the Earth [13].

c: The angular distance between the two points in radians, derived using the atan2 function, which correctly handles positive and negative values [13].

**4. Distance Calculation**: According to [12], The final distance d is computed as:

(xvii)

This provides the great-circle distance in meters [12].

***3.5 Database Design :***

The database consist of three tables:

1. **Students**: Contains data about individual student
2. **Lecture:** Contains details about the lecture
3. **Attendance:** Contains mappings of students to the respective lectures

The Structure of the tables is as follows:

1. **Students**

**Table 2 : Structure of Students Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| u\_id | VARCHAR (Primary Key) | Unique identifier for the student |
| roll\_no | INT | Roll number of the student |
| name | VARCHAR | Full name of the student |

1. **Lecture**

**Table 3: Structure of Lecture Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| l\_id | VARCHAR (Primary Key) | Unique lecture ID |
| lecture\_name | VARCHAR | Name of the lecture |
| branch | VARCHAR | Branch of study |
| division | VARCHAR | Class division |
| teacher\_initials | VARCHAR | Initials of the teacher (Uniquely identifier of teacher) |
| date | DATE | Date of the lecture |

1. **Attendance**

**Table 4 : Structure of Attendance Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| u\_id | VARCHAR (Foreign Key referencing Students.u\_id) | Student's unique ID |
| l\_id | VARCHAR (Foreign Key referencing Lecture.l\_id) | Lecture's unique ID |

* 1. ***Servers and Load Balancers:***

**1. WebSocket Server** : Responsible for connecting clients (mapping students to their respective teachers)

**2. Flask Servers** : Flask servers handle multiple functionalities, categorized by routes:

1. /attendance\_data: Route for inserting a lecture and mapping student with it in Database.
2. /student\_details: Route for inserting student details by college authorities.
3. /report\_data: Route for retrieving student attendance records for a lecture for report generation.

**3. Load Balancers**: Responsible to prevent overloading of WebSocket and flask servers

1. Round Robin: Sequentially distributes incoming requests among the flask servers [16].
2. Least connections: Retrieves the URL of the WebSocket server with the fewest active connections for making an appropriate request [17].

**4. Redis server**: Facilitates communication between WebSocket servers and the Least Connections Load Balancer, helping to monitor active connections.

* 1. ***Working of Load Balancers:***

In a real-time environment, multiple users may attempt to access the server simultaneously. To prevent server overload and enhance availability, load balancers are employed to efficiently distribute traffic among the servers.

1. The load balancer managing traffic on the WebSocket server uses the 'Least Connections' algorithm, which directs traffic to the server with the fewest active connections [17].

2. For the 'Validation' and 'Report Generation' servers, the load balancer uses the 'Round Robin' algorithm to evenly distribute traffic, sequentially sending requests to each server in turn [16].

In ClassConnect, round robin load balancer works independently by just transferring the incoming requests sequentially to the servers. For least connections load balancer the component that supports its functioning in real-time is Redis Pub/Sub.

**3.7.1 Redis Pub/Sub in Load Balancing**

Redis Pub/Sub (Publish/Subscribe) is a messaging mechanism where messages are sent from publishers to subscribers through channels [18].

The key components are:

1. **Channels:** Named communication pathways where messages are published and received [18].

2. **Roles:**

1. **Publisher:** Sends messages to a channel [18].
2. **Subscriber:** Listens to a channel and receives messages published on it [18].

It’s working is as follows:

1. A client subscribes to one or more channels and listens for messages published to the channels [18].

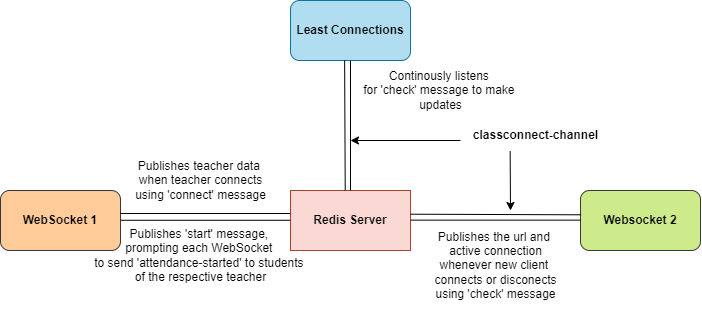
2. Another client publishes a message to the channel [18].

3. All clients subscribed to the channel immediately receive the message [18].

For ClassConnect, the Redis server is hosted on an Azure Virtual Machine using Docker**.** The WebSocket servers and the least connection load balancer are connected to the Redis server via the classconnect-channel.

Websocket system workflow using Redis:

1. Both WebSocket servers and the load balancer subscribe to the classconnect-channel.
2. WebSocket servers publish three types of messages on the channel:
3. Connect: Adds a teacher's entry to all WebSocket servers.
4. Start: Notifies all WebSockets that a teacher has started the attendance process.
5. Check: Updates the load balancer with the active connection count for each WebSocket server.
6. When a student or teacher connects or disconnects, a ‘check’ message containing the WebSocket URL and active connection count is published. The load balancer updates the count for the respective WebSocket based on this information.
7. Students cannot connect to a WebSocket unless the corresponding teacher is already connected. When a teacher connects, a ‘connect’ message with the teacher's data is published on the channel, allowing other WebSockets to register the teacher's presence. This ensures students are routed to an available WebSocket associated with their teacher.
8. When a teacher starts attendance, the WebSocket publishes a ‘start’ message on the channel. This message includes the teacher ID and location, notifying all WebSockets that attendance has begun for that specific teacher.



**Figure 4 : Communication on class-connect channel**

# 4.0 RESULT AND DISCUSSIONS

***4.1 Relations between time taken and number of students :***

Our proposed methodology successfully improves various aspects of the attendance in contrast to existing methods. One such aspect is time taken during the attendance process. After the mathematical articulation of the attendance process, we get the following three relations between time taken (t) and number of students (n):

*where, t = total time taken*

*k = average time taken for individual student (constant)*

*n = number of students*

This relation is derived from the traditional roll-call attendance management system. In the traditional roll call system, the time taken is linearly dependent on the number of students.

For example, If number of students are 20 and it takes on an average of 2 seconds for the student to respond to their respective roll number.

If number of students is increased to 40:

As we can observe from the above calculation that this method fails to address the problem of increasing number of students.

*where, t = total time taken*

*k = average time taken for individual student (constant)*

*n = number of students*

*c = sum of time taken for all failed attempts and*

*device boot and/or ready time*

This relation depicts the relation of time and number of students for all the device-based methods that are set in the following way: A device (be it fingerprint scanner, RFID scanner, face detection camera, etc) that is set outside the classroom and has check-check out method. The relation is linear due to the addition of a constant time overhead c.

For example, If the number of students is 20, the average time per student is 2 seconds, and the overhead time ccc is 10 seconds:

If the number of students increases to 40:

As observed, even with this method, the total time increases linearly with the number of students.

*where, t = total time taken*

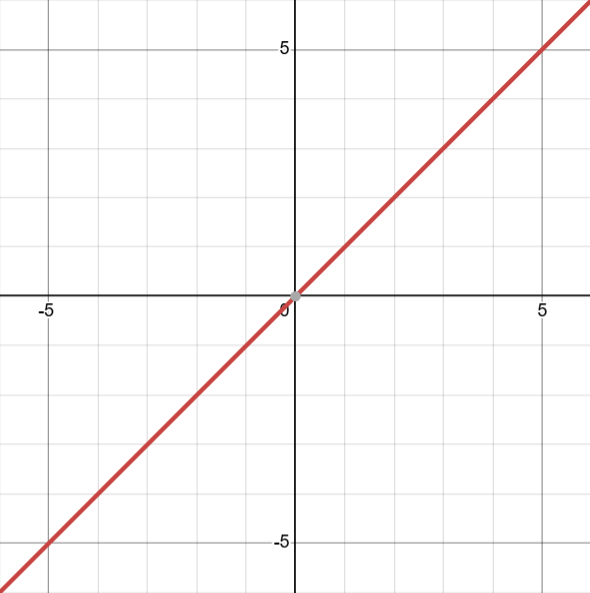
*c = average time taken by using our app (1 minute)*

This relation depicts time dependency for our proposed application. In the proposed application the time taken is

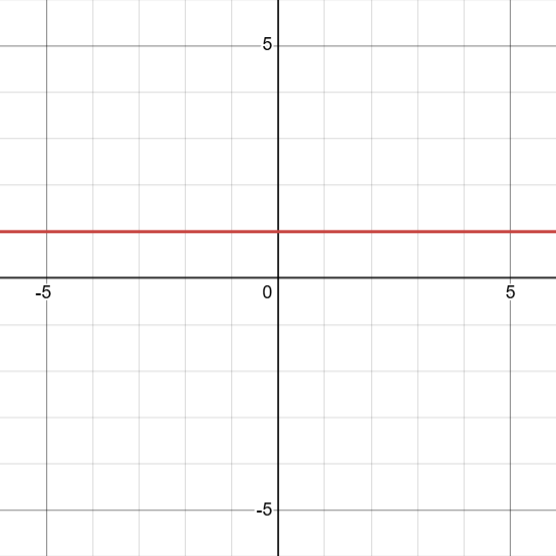
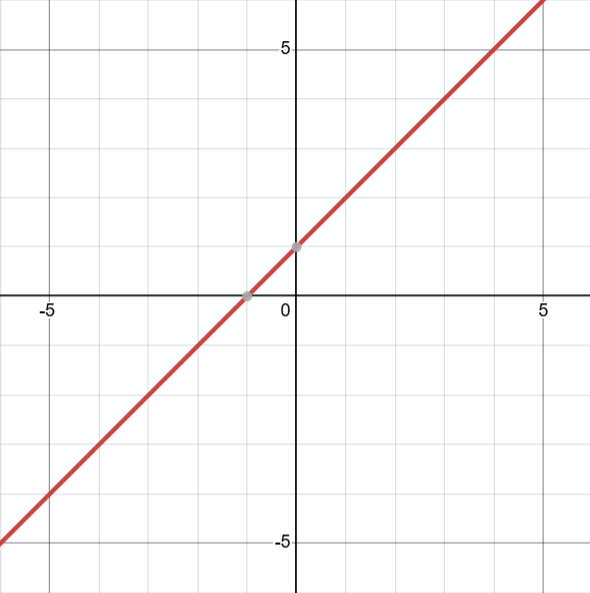
only dependent on the constant number which can be defined either as a range of constant time period (e.g., )or it can be taken as average time of attendance which can be derived after the series of tests (e.g., 60s).

In this case the number of students does not correlate with the time taken. So, for any number of students, it would take the

same approximated period of time.

Now if we compare the three methods with a graphical representation as follows:

**Figure 5 (a):**

 **Figure 5 (b):**

**Figure 5 (c):**

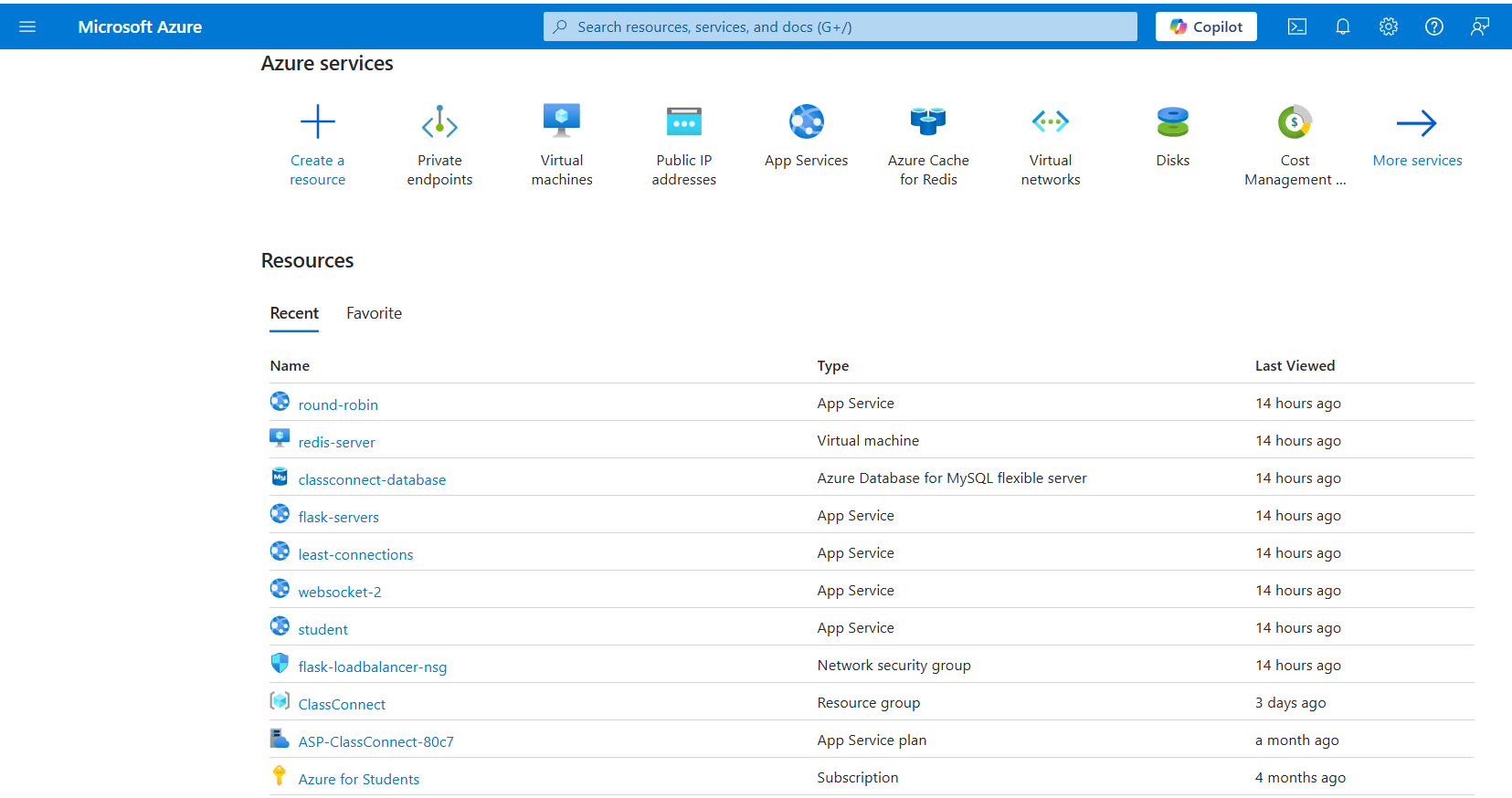
**4.2 Implementation Summary:**

The Attendance Tracking and Management System successfully integrates multiple technologies to provide an efficient, automated solution for attendance management. By combining WebSocket communication, biometric authentication, geolocation verification, and cloud hosting on Azure, the system offers a better solution for real-time attendance tracking.

* 1. The system ensures that both teachers and students can easily connect to the WebSocket server.
  2. The combination of biometric authentication and geolocation verification ensures that only students who are present at the correct location and authenticated can mark their attendance.
  3. The use of load balancing via Redis Pub/Sub and a Least Connections algorithm for WebSocket servers ensures optimal distribution of traffic, preventing server overloads even under high traffic.
  4. The MySQL database design is efficient and avoids redundancy, especially with the Attendance table, which ensures that student data is not duplicated.
  5. All servers, including WebSocket, Flask, and MySQL, are hosted on Microsoft Azure. The use of private endpoints between the Flask and MySQL servers adds an additional layer of security, ensuring that sensitive student and session data is protected from unauthorized access.

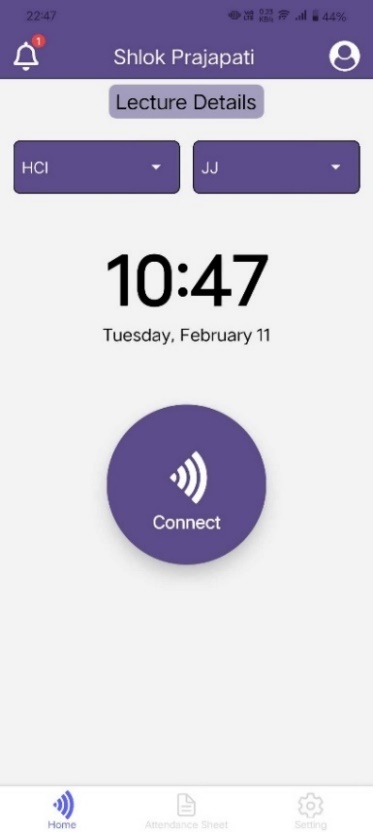
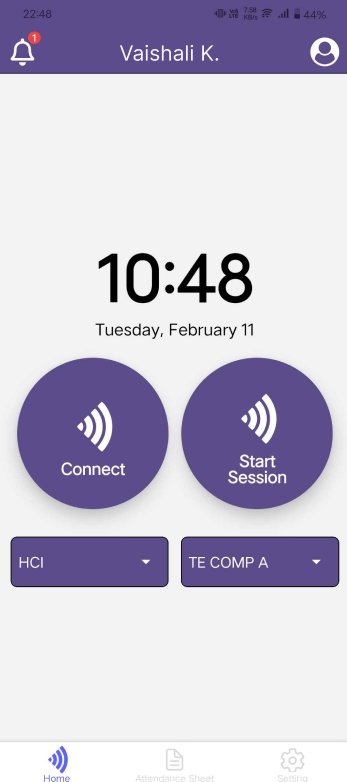
**4.2.1 Hosting:**

The deployment of the servers is done on Microsoft Azure using the Azure App service [19]. The MySQL server which hosts the database is created using the Azure MySQL server [20]. Additionally, the Redis server is deployed and managed using Docker on a Linux-based virtual machine (VM) [19].



**Figure 6: Azure portal homepage showing deployed servers**

Following are the images of the developed application:



**Figure 7: Teacher view (left) and Student view (right**

**4.3 Challenges and Limitations**

1. While biometric authentication adds a layer of security, it requires students to have devices capable of performing such authentication, limiting accessibility for users without advanced smartphones.

2. Transferring geolocation information raises privacy concerns. It is important that the system uses encryption and follows legal guidelines for handling sensitive personal data to ensure compliance with data protection laws.

**4.4 Future Upgrades**

1.Teachers will have the ability to manually edit the list of present students after clicking the 'stop' button, allowing for corrections.

2. A notification system will be available for students, providing real-time updates on their attendance percentage.

# CONCLUSION

The Attendance Tracking and Management System developed in this project effectively integrates biometric authentication and geolocation-based attendance marking, addressing key challenges in classroom management which involves eliminating proxies and reducing the overall time to track and manage attendance. The integration of various technologies, including Azure for hosting and MySQL for data storage, provides a robust infrastructure that ensures data integrity and accessibility.

Our literature review has provided a thorough examination of key research papers in the domain of Attendance Management Systems. Each paper contributed unique methodologies and approaches, shedding light on diverse techniques to address challenges in this field.

Our methodology proposes a system that streamlines attendance processes by utilizing WebSocket connections, biometric authentication, and geolocation verification, eliminating proxy attendance. It efficiently manages sessions using teacher initials and mobile OS biometrics, ensuring security and ease of use. With scalable architecture and potential upgrades, the system enhances performance and user experience.

In the future, we are planning to include manual attendance edits for teachers after report generation and real-time attendance notifications for students, enhancing flexibility.

**REFERENCES**

1. Darwis, M., Niswaty, R., Arhas, S. H., Rudi, & Jamaluddin. (2021). Fingerprint electronic attendance application. *IOP Conference Series: Materials Science and Engineering, 1125*(1), 012076.
2. Mishra, S., Kumar, C., Ali, A., & Bala, J. (2021). Online Attendance Monitoring System using QR code (OAMS). 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM).
3. K. J. Liew and T. H. Tan (2021) QR Code-Based Student Attendance System. 2nd Asia Conference on Computers and Communications (ACCC), Singapore, 2021, pp. 10-14, doi: 10.1109/ACCC54619.2021.00009.
4. Jantan, S. R., Rasli, A., Mohd Zainudin, A. J., Mustafa@Dain, N. W., & Khairuzzaman, M. G. (2022). Smart Attendance for Faculty Monitoring System Using the Bluetooth Low Energy: Design and Implementation. Preprints. <https://doi.org/10.20944/preprints202211.0001.v1>
5. Jha, P. B., Basnet, A., Pokhrel, B., Pokhrel, B., Thakur, G. K., & Chhetri, S. (2023). An Automated Attendance System Using Facial Detection and Recognition Technology. Apex Journal of Business and Management, 1(1), 103–120. Retrieved from <https://www.nepjol.info/index.php/ajbm/article/view/61984>
6. K. Alagasan, M. H. Alkawaz, A. Iqbal Hajamydeen and M. N. Mohammed (2021). A Review Paper on Advanced Attendance and Monitoring Systems, 2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2021, pp. 195-200, doi: 10.1109/ICSGRC53186.2021.9515249.
7. Alaa Albahrani, Zainab Ali AL-Ali, Zainab Yousef Al-Ali, Aqeela Al-Mssri, Mashael AL-Shalan, Atta-ur-Rahman, and Gomathi Krishnasamy (2022). Smart attendance management system. *IJCSNS International Journal of Computer Science and Network Security, Vol. 22 No. 6 pp. 762-770,* DOI: 10.22937/IJCSNS.2022.22.6.96.
8. Tu, Y.-J., Zhou, W., & Piramuthu, S. (2021). Critical risk considerations in auto-ID security: Barcode vs. RFID. *Decision Support Systems, 142*, 113471, <https://doi.org/10.1016/j.dss.2020.113471>.
9. S. Yang, Y. Song, H. Ren and X. Huang (2016). An automated student attendance tracking system based on voiceprint and location, 11th International Conference on Computer Science & Education (ICCSE), Nagoya, Japan, 2016, pp. 214-219, doi: 10.1109/ICCSE.2016.7581583.
10. Vinay, M. D., Kumar, M. H., Hemanth, B., & Tomar, D. S. (2023). Smart attendance system using biometric and GPS. In *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, India, 2023, pp. 1-6, doi: 10.1109/SCEECS57921.2023.10062969.
11. A. M. W. S. Edoh, T. Djara, Abdou-Aziz Sobabe Ali Tahirou, and Antoine Vianou (2023). Biometric Authentication methods on mobile platforms: An Introduction to fingerprint strong feature extraction, International Journal of Mobile Computing and Multimedia Communications 14(1):1-16, DOI: 10.4018/IJMCMC.334130.
12. H. R. Hatta, M. H. Suroso, I. F. Astuti, D. M. Khairina (2021), Application of Haversine Formula in Education Game “Landmark Nusantara”, Proceedings of the 2nd Borobudur International Symposium on Science and Technology, PP: 234-237, ISSN- 2352-5401, https://doi.org/10.2991/aer.k.210810.039.
13. N. M. A. E. D. Wirastuti et al.(2023), Implementation of geographic information system based on Google Maps API to map waste collection point using the Haversine Formula method,” Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), journal-article, Sep. 2023. doi: 10.26555/jiteki.v9i3.26588.
14. Moritz, H. (1980). Geodetic Reference System 1980. Bulletin Géodésique, 54(3), 395–405.
15. Maknun, C. L., Rosjanuardi, R., & Jupri, A. (2021). Using GraphicCalculator in Trigonometric Learning: A Didactical Design for Teacher. MaPan: Jurnal Matematika dan Pembelajaran, 9(1), 85-101. https://doi.org/10.24252/mapan.2021v9n1a6.
16. N, M., & A, P. (2018). An Efficient Improved Weighted Round Robin Load Balancing Algorithm in Cloud Computing. International Journal of Engineering & Technology, 7(3.1), 110-117. <https://doi.org/10.14419/ijet.v7i3.1.16810>.
17. A. R. Rahmika, Z. Tahir, A. W. Paundu, and Z. Zainuddin (2023), Web Server Load Balancing Mechanism with Least Connection Algorithm and Multi-Agent System, CommIT Journal 17(2), 245–258,2023, DOI: 10.21512/commit.v17i2.8872.
18. D. Eddelbuettel et al (2022), Redis for market monitoring, pp. 1–5, 2022, [Online]. https://doi.org/10.48550/arXiv.2203.08323
19. V.P. Desai, K.S. Oza, P.P. Shinde, P.G. Naik (2021), **Microsoft Azure**: Cloud platform for application service deployment, International Journal of Scientific Research in Multidisciplinary Studies, Vol.7, Issue.10, pp.20-23, October (2021), E-ISSN: 2454-9312.
20. P. Pandit (2021), Cloud computing case study on **Microsoft Azure,** report, Aug. 2021. doi: 10.13140/RG.2.2.22421.86245.