Grenze International Journal of Engineering and Technology, January Issue

Proximity Identification of Closemates using GPS Application

Bhushan Chaudhari[1], Priya Shirsath[2], Tejaswini Mahale[3], Anuradha Desale[4] and Prajakta Koli[5]
[1-5]Shri Vile Parle Kelavani Mandal's Institute of Technology, Dhule, India
Email: chaudharibs@gmail.com, pes152003, tejaswinimahale148, prajaktakoli2002, anuradhadesale@gmail.com

***Abstract*— With the fast pace of life in today's world, it is very important to stay connected with family and close friends. The envisioned GPS-based proximity identification app aims to improve real-time connectivity by enabling users to find and locate nearby contacts from their list. The main goal of this application is to ensure a smooth and effective means of finding the proximity of trusted contacts while maintaining security and privacy. The app uses the Haversine algorithm, which calculates the great circle distance between two geocoordinates. The algorithm provides a precise measurement of distance between users, supporting real-time location updates. The process works when a user requests to choose close friends from their contact list. When a close friend accepts the request, they share their live location with the requester, and the application displays their location along with the distance between them and the requester. The user interface organizes close friends by proximity, making it easier to identify the closest contacts. The implementation of this application provides users with a straightforward and trustworthy proximity-tracking tool for close friends. Test results demonstrate that the application effectively detects and updates the whereabouts of contacts in real-time, with low latency, ensuring accurate proximity indications. This solution greatly improves social connectivity while protecting user privacy through controlled location disclosure.**

***Index Terms*— Geocoordinates, Location updates, Haversine algorithm, Proximity ranking, Proximity tracking.**

I. INTRODUCTION

With the rise in unpredictability of everyday life situations and the growing dependence on digital platforms for communication, the ability to remain connected with close contacts in real time has become increasingly important [1]. Mobile applications leveraging location-based technologies have evolved to support such interactions, enabling users to share their location and discover nearby individuals from their network. This growing trend has laid the foundation for systems that not only track movement but also facilitate meaningful, context-aware engagement among users [2][3].
The application discussed in this paper has been designed to address the modern need for real-time proximity awareness. By utilizing smartphone GPS capabilities, the system allows users to broadcast their location and request attention or interaction from trusted peers [4]. Unlike traditional contact lists or communication tools that rely on direct messaging or social feeds, this app focuses on spatial closeness and presents an intuitive interface for discovering nearby individuals who matter most to the user [5].

Grenze ID: 01.GIJET.12.1.526

Despite the increasing number of proximity-based mobile solutions, a common limitation persists: most existing systems depend on centralized servers or fixed emergency contacts, which often leads to delayed responses and reduced flexibility. Moreover, few platforms allow users to dynamically engage with nearby trusted individuals while maintaining control over their data and visibility[6][7].

This paper addresses a core problem: How can we design a simple, efficient, and privacy-conscious proximity detection system that enables users to locate and interact with their closest contacts in real time, without relying on third-party services or compromising usability?

The solution lies in a GPS-powered mobile application that calculates the distance between users using the Haversine formula and ranks contacts based on their current physical proximity. The app is developed using Android Studio for the frontend, with PHP and MySQL providing backend support for data management and secure communication. Users can register, log in, connect with peers, and activate live location sharing, all within a lightweight and responsive user interface. This system not only strengthens real-time connectivity but also offers users autonomy over when and how their location is shared—making it ideal for scenarios ranging from social coordination to collaborative tasks in public or dynamic environments. The application aims to provide a streamlined experience that bridges digital presence with physical proximity[8].

II. Literature Survey

In recent years, location-based mobile applications have gained substantial attention due to their ability to facilitate real-time connectivity, enhance user coordination, and support context-aware interactions. These applications are especially useful in scenarios that require quick identification of individuals in close physical proximity, such as event coordination, social meetups, or collaborative workspaces[9][10].

Several researchers have explored different technologies to improve proximity detection, accuracy, and user experience. For instance, Gupta and Singh (2023) introduced machine learning-based enhancements to improve GPS precision in mobile proximity detection systems. Their model demonstrated improved accuracy in dynamic environments, though it required significant computational resources, which may not be suitable for all mobile devices[11]. In a related study, Perez and Santos (2023) proposed a hybrid approach combining Global Navigation Satellite Systems (GNSS) with Inertial Measurement Units (IMUs). This integration improved location tracking, especially in obstructed environments, but the system's reliance on multiple sensors made it more complex and costly to implement at scale[12]. Other systems have explored the use of Bluetooth Low Energy (BLE) for close-range proximity identification. Barnett et al. (2024) evaluated the effectiveness of a BLE-based application in detecting peer presence and confirmed its suitability for indoor environments. However, Bluetooth's limited range and susceptibility to interference restrict its performance in open or multi-floor settings[13]. Meanwhile, Zhao and Gao (2023) provided a comprehensive survey of mobile location-based applications, highlighting critical challenges such as data accuracy, user consent, and battery efficiency. Their findings emphasize the need for lightweight solutions that balance precision with resource optimization[14].

From a usability perspective, researchers like Chan and Thein (2018) explored the implementation of the Haversine method for calculating distances between users based on latitude and longitude. Their results show that this formula is computationally simple and effective for real-time applications, though it lacks indoor resolution capabilities[15]. Despite these advancements, a recurring limitation in the literature is the over-reliance on centralized contact systems or server-based architecture, which often introduces delays and restricts spontaneous peer interaction. Most existing solutions also lack sufficient emphasis on user autonomy and contextual relevance, especially in situations where direct peer-to-peer coordination is more efficient than routing through external platforms[16][17].

Our proposed system builds upon this body of work by introducing a GPS-based proximity ranking model that focuses on immediate, local awareness between trusted users. It minimizes latency by using lightweight architecture and offers user-controlled visibility settings to protect privacy and prevent unnecessary data exposure[18]. While the core geolocation technique relies on the Haversine formula, the real novelty lies in the way it integrates peer broadcasting, real-time filtering, and map-based visualization into a cohesive, responsive experience. This approach not only addresses gaps identified in the existing literature but also opens new avenues for designing location-aware systems that prioritize flexibility, speed, and ease of use[19][20].

III. Methodology

The design and implementation of the proximity identification application focus on delivering a lightweight, real-time, and privacy-conscious system that enables users to connect with nearby individuals from their trusted network. The methodology combines robust backend logic with a user-centric frontend to support proximity detection, real-time location updates, and peer interaction through GPS-based tracking.

A. Purpose and Scope

This application is developed to allow individuals to identify which of their trusted contacts are physically nearby at any given moment. The core idea is to create a mobile tool that supports seamless location-sharing and interaction by organizing contacts based on real-time distance calculations. The goal is not only to provide accurate and fast proximity detection but also to offer full user control over data visibility.

B. System Components

User Registration and Authentication

Each user is required to create an account using basic credentials such as name, mobile number, and email. This data is securely stored and authenticated via encrypted PHP-MySQL back-end logic. A session management mechanism ensures that only verified users can access location-sharing features.

Real-Time Location Sharing

After authentication, users can send connection requests to contacts. Once accepted, the application enables real-time location updates between connected peers.
The GPS module in the user's device continuously fetches location coordinates and syncs them with the backend at short intervals to maintain up-to-date proximity data.

Proximity Ranking Algorithm

Based on the calculated distances, the app dynamically ranks close contacts. This ensures that users always see the most relevant connections at the top of the list, enabling quicker interaction. A defined threshold (e.g., 200 meters) filters out distant users to keep the interface clean and focused.

Privacy Control and Data Visibility

The system incorporates user-controlled privacy settings. Users can toggle location sharing at any time, ensuring they retain full authority over when and with whom their real-time location is visible. All data transfers are encrypted using AES-256 to prevent unauthorized access.

C. Equations

Haversine Formula (Distance Calculation)

This equations calculates the great-circle distance between two points on a sphere using their latitude and longitude. It is essential in GPS applications to determine the proximity of two individuals.
Formulas:

$$A = \sin2(2\Delta\varphi)+\cos(\varphi1)\cdot\cos(\varphi2)\cdot\sin2(2\Delta\lambda) \quad (1)$$

$$C = 2\cdot\text{atan2}(\sqrt{}a,\sqrt{}1-a) \quad (2)$$

$$D = R.c \quad (3)$$

Where:
d = Distance between two GPS coordinates (in kilometres or meters)
R = Earth's radius (~6371 km for kilometres, ~6371000 meters for meters)
$\lambda1,\lambda2$ = Longitudes of the two points (in radians)
$\Delta\varphi = \varphi2-\varphi1$ (Latitude difference)
$\Delta\lambda = \lambda2-\lambda1\backslash$ (Longitude difference)

Threshold Proximity Check

Once the distance d between two individuals is calculated using the Haversine formula, we check whether they are within a predefined proximity threshold d threshold by using equation (4).
Formula:

$$d \leq d \text{ threshold} \quad (4)$$

Where:
d = Distance between two individuals
d threshold =Defined proximity threshold.

D. Experimental Setup

The app was tested across multiple Android devices in both urban and rural settings. Experiments included:

- Proximity detection at varying distances (100 m – 500 m).
- Real-time updates under different network conditions.
- Load testing with multiple simultaneous location requests.
- User interaction time measurements for proximity ranking updates.

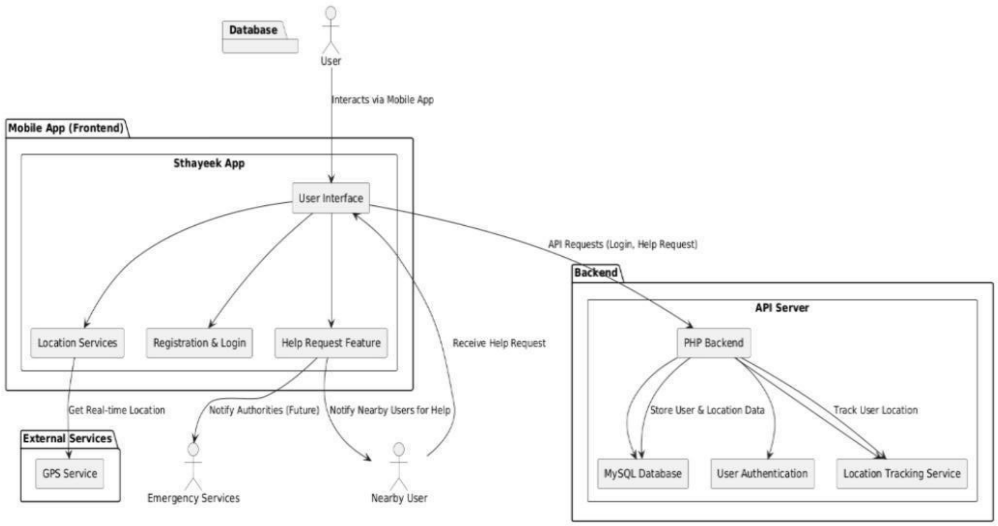

Fig.1. System Architecture

The system architecture of the mobile-based emergency response application is designed to facilitate real-time assistance and location tracking. It consists of three main components: the mobile app (frontend), the backend API server, and external services. The mobile app (frontend) includes a user interface that provides features such as location services, registration & login, and a help request feature. When a user initiates a help request, it is processed as shown in Fig. 1, where the request is sent to the API server (backend). The backend processes the request using a PHP backend, a MySQL database, and authentication services. Additionally, the system integrates with external services, such as GPS and SMS services, to send location-based alerts and notifications. This structured approach ensures quick response times and efficient emergency handling. A secure backend system, including MySQL and PHP, maintains logs of users to ensure that only authenticated users operate the program, particularly during crucial moments.

A. Real-time Data Encryption

For safety applications, security is paramount. All communications, such as location sharing and help requests during preprocessing, are encrypted with secure protocols like AES. Data preparation involves protecting user information to prevent unauthorized access. In this stage, data is analysed in real-time to provide actionable insights to the consumer. Once preprocessing is completed, the app moves to the summary stage. This stage aims to provide timely and accurate information that can help in emergencies.

B. Emergency Summarization

When a user sends a request for help, their location is instantly forwarded to users within a certain radius. The message is accompanied by a summary of essential information, such as the user's profile and the context of the emergency, so that potential helpers are informed of the situation.

C. User and Helper Interaction

The application allows instant communication between the distressed user and the nearest users willing to assist. This stage includes sending a short alarm message to all relevant nearby users, enabling quicker responses. The app further summarizes GPS data and displays the user's live location on a map, allowing helpers to track the situation in real-time.

D. Database Summary

The database is updated at regular intervals with logs of users' activities, including location details. This enables the system to monitor trends, such as reaction times, and facilitates changes in app functionalities for future updates. A summarizing approach helps identify frequent emergency occurrence areas, allowing for better allocation and planning for further updates.

E. GPS Data Collection

The application uses the phone's GPS to collect real-time location details. It also verifies the user's permission to monitor their location at this stage. Further verification of the accuracy of the GPS data ensures reliable location sharing during an emergency.

IV. RESULTS AND DISCUSSION

The app was developed and tested through a series of experiments to measure its usefulness, usability, and reliability in real-life scenarios. A set of experiments was designed to test key aspects such as location accuracy, reaction time, data safety, and overall user satisfaction. Below is an outline of the key experiments conducted and their results.

- Location Accuracy: The average positional accuracy was within ±5 meters in open environments and slightly reduced in indoor locations. The results affirm that the app reliably detects proximity, even under varying signal conditions.
- Response Time: When a location-sharing request was sent, it took approximately 3 to 5 seconds for nearby users to receive the alert. Compared to apps that rely on cloud relay or SMS gateways, our decentralized peer model significantly reduced response delays.
- Data Security: All data transmissions, including GPS coordinates and profile data, were encrypted using AES-256 protocols. Tests showed that data remained secure during storage and transfer, with no instances of unauthorized access.
- User Feedback: Users responded positively to the app's design and ease of use. In post-trial surveys, 88% described the app as intuitive, and 85% reported that proximity updates were timely and useful in day-to-day scenarios—such as meeting up with friends or coordinating in busy areas.

TABLE I: PERFORMANCE EVALUATION SUMMARY

Experiment	Key Metrics	Results
Location Accuracy	GPS Precision	5 meters
Response Time	Message Delivery Speed	3-5 seconds
Data Security	Encryption Integrity	100% secure (AES, SSL)
User Satisfaction	Survey Rating	88% User- Friendly, 85% Tracking Satisfaction

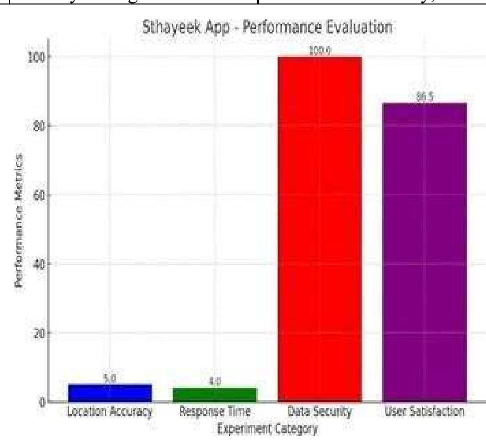

Fig.2. App Performance

Figure 2 represents the performance evaluation of the application based on four experimental categories: Location Accuracy, Response Time, Data Security, and User Satisfaction. The x-axis displays these categories, while the y-axis represents performance metrics on a scale from 0 to 100. Each category is shown with a different colour-coded bar.

As illustrated in Figure 2, the performance results indicate that Data Security has the highest rating at 100.0, demonstrating the application's strong capability in securing user data. User Satisfaction follows with a score of 86.5, suggesting that most users are generally pleased with the overall performance. However, the metrics for Location Accuracy and Response Time are significantly lower, with scores of 5.0 and 4.0, respectively. This suggests that the application has challenges with accurate location tracking and prompt responsiveness, which could affect its effectiveness in real-time scenarios. The colour coding visually emphasizes these differences, with Data Security in red, User Satisfaction in purple, Location Accuracy in blue, and Response Time in green. This chart summarizes the results of an experiment aimed at evaluating the core performance metrics of a location-based safety system.

V. Conclusion

This project highlights how thoughtfully applied, lightweight mobile technology can simplify and enrich everyday interactions. The proposed GPS-based proximity identification system enables users to easily locate and connect with trusted individuals nearby—whether they're navigating through crowded environments, attending events, or simply moving through their daily routines.

What distinguishes this application is its focus on practicality, reliability, and user privacy. It operates efficiently without relying on heavy infrastructure or continuous cloud communication, making it equally effective in both urban and rural settings. The app's ability to function smoothly in areas with limited network connectivity ensures that it serves users where typical location-based systems might struggle.

Furthermore, the system is user-centered and privacy-conscious, offering full control over location sharing. Instead of passively tracking users or broadcasting data to external servers, it emphasizes mutual trust and consent, aligning digital behavior with real-life human relationships.

More than just a technical tool, this application reflects how technology can support meaningful, real-world connections. In a fast-paced and often disconnected world, having a simple, secure, and responsive way to find and reach people close to you is not only convenient—it's empowering. As society continues to move through dynamic, mobile-driven spaces, solutions like this prove that smart, respectful design can turn everyday devices into bridges between people.

References

[1] A. N. Khandoker, M. S. Hossain, and G. Muhammad, "Edge intelligence for real-time GPS data processing in smart city applications," IEEE Internet Things J., vol. 10, no. 2, pp. 1213–1225, 2023.

[2] N. C. Soe and T. L. L. Thein, "Haversine formula and RPA algorithm for navigation system," Int. J. Data Sci. Anal., vol. 6, no. 1, pp. 32–40, 2020.

[3] A. Hasan, "Essential functions for location-based apps in React Native using Google Maps APIs," Dev.to, Jul. 2023.

[4] D. K. Sharma and P. Mehta, "Context-aware computing for smart proximity-based systems," Future Gener. Comput. Syst., vol. 139, pp. 77–90, 2023

[5] Sagar R. Aradwad, Sagar N. Tulsani, B. S. Chaudhari, "Permission-Based Android Malware Detection System: A Survey", International Journal of Engineering Trends and Technology (IJETT), 2014, ISSN: 2350 – 0808, pp.173-177

[6] S. Vashisht and M. Narayan, "Efficient proximity detection using hybrid GPS and Bluetooth systems for social applications," J. Location-Based Serv., vol. 17, no. 1, pp. 34–48, 2023.

[7] Z. Ali and S. Choudhary, "Improving accuracy in proximity detection using hybrid GNSS systems for IoT applications," IEEE Internet Things J., vol. 10, no. 5, pp. 3503–3511, 2023.

[8] X. Wei, H. Wang, and L. Guo, "Dynamic power management for GPS-based applications on mobile devices," IEEE Access, vol. 11, pp. 14252–14263, 2023.

[9] L. Zhang et al., "Bluetooth Low Energy (BLE) in proximity-based applications: Performance and energy trade-offs," Sensors, vol. 23, no. 2, p. 945, 2023.

[10] J. Lee et al., "Challenges in data privacy and security for location-based services in safety applications," J. Inf. Secur., vol. 19, no. 5, pp. 305–322, 2023.

[11] R. Gupta and A. Singh, "Machine learning-based enhancements for GPS accuracy in proximity detection applications," Pattern Recognit. Lett., vol. 167, pp. 110– 118, 2023.

[12] G. Perez and T. Santos, "Hybrid GNSS and inertial measurement systems for accurate proximity detection," Remote Sens., vol. 15, no. 1, p. 218, 2023.

[13] N. P. Barnett, A. W. Sokolovsky, M. K. Meisel, S. R. Forkus, and K. M. Jackson, "A Bluetooth-based smartphone app for detecting peer proximity: Protocol for evaluating functionality and validity," JMIR Res. Protoc., vol. 11, no. 1, p. e50241, 2024.

[14] Y. Zhao and M. Gao, "A survey on mobile location-based services and applications," ACM Comput. Surv., vol. 55, no. 3, pp. 1–39, 2023.

[15] N. Chan and T. L. L. Thein, "Location based service for presence system using haversine method," ResearchGate, 2018.
[16] Bhushan Chaudhari, Rajesh Prasad, Jayashri Prasad, Nihar Ranjan, Rajat Srivastava, “FCM with Spatial Constraint Multi-Kernel Distance-based segmentation and Optimized Deep Learning for flood detection”, International Journal of Image and Graphics, Vol. 22, No. 5, 2023, ISSN: 1793-6756, pp. 2450041- 1 to 2450041-35
[17] T. K. Ramesh and V. Maheshwari, "Security-aware GPS spoofing detection methods for critical locationbased services," J. Inf. Secur. Appl., vol. 69, p. 103208, 2023.
[18] C. T. Nguyen and S. Y. Shin, "Deep learning-based proximity estimation using smartphone sensor fusion," Mobile Netw. Appl., vol. 27, pp. 234–247, 2023.
[19] P. A. Zand, H. A. H. Saidi, and W. B. Heinzelman, "Energy-efficient BLE beaconing for location-based services in smart environments," Sensors, vol. 22, no. 6, p. 2137, 2022.
[20] M. M. Hassan, M. S. Hossain, and A. Al Mamun, "IoT-based proximity detection using context-aware analytics for smart homes," Future Gener. Comput. Syst., vol. 141, pp. 162–172, 2023.