RESEARCH ARTICLE OPEN ACCESS

GeoReminder: An Intelligent Location-Based Reminder System Using GPS and Geofencing

[1]V Dinesh, [2]Ms M Geethanjali M.Sc.,
[1]Final year Department of computer science student
[2]Head of the Computer Science Department, *Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore, Tamilnadu, India.*

Abstract *- Location-based services have become an essential component of modern mobile applications, enabling context-aware functionality based on user location. However, most existing reminder systems are time-based and fail to provide relevant notifications for tasks that depend on geographic context. This limitation reduces the effectiveness of reminders in real-world scenarios.*

*To address this issue, this paper proposes **GeoReminder**, an intelligent location-based reminder application that utilizes GPS and geofencing techniques to deliver context-aware notifications. Initially, user-defined reminder data, including location coordinates, radius, and task details, are collected and stored using cloud-based services. The system continuously monitors the user's real-time location and preprocesses location data to ensure accuracy and efficiency.*

A distance calculation mechanism based on geographic coordinates is implemented to determine proximity between the user and predefined locations. When the user enters a specified geofence boundary, the system triggers notifications using local notification services. The application integrates key modules such as authentication, real-time location tracking, proximity detection, and cloud database management.

Additionally, the system is optimized to reduce battery consumption by using periodic location updates instead of continuous tracking. The results are presented through an intuitive user interface with interactive map visualization for better usability. Experimental results demonstrate that the proposed system significantly improves the relevance and effectiveness of reminders compared to traditional time-based approaches, thereby enhancing user productivity and task management.

Keywords *– Location-Based Reminder, Geofencing, GPS, Flutter, Firebase, Proximity Detection, Mobile Application*

1.Introduction

The rapid growth of mobile applications has significantly transformed how individuals manage their daily activities, tasks, and schedules. Reminder systems play a crucial role in assisting users to organize their responsibilities efficiently. Most existing reminder applications are primarily time-based, triggering alerts at predefined times regardless of the user's location. While such systems are effective for time-sensitive tasks, they are often inadequate for real-world scenarios where task execution depends on geographic context.For instance, reminders such as purchasing groceries, visiting a bank, or refueling a vehicle are more relevant when the user is near a specific location rather than at a fixed time. The lack of location awareness in traditional systems reduces the effectiveness of reminders and may lead to missed or ignored notifications. As a result, there is a growing need for intelligent reminder systems that can adapt to the user's environment and provide context-aware alerts. Location-Based Services (LBS) and geofencing technologies offer a promising solution to this problem. By leveraging Global Positioning System (GPS) data, mobile applications can track the user's real-time location and trigger actions based on predefined geographic boundaries. However, implementing such systems introduces challenges such as battery consumption, accuracy of location tracking, and efficient proximity detection. Therefore, it is essential to design a system that balances performance, accuracy, and resource utilization.This paper proposes GeoReminder, an intelligent location-based reminder application that utilizes GPS and geofencing techniques to deliver context-aware notifications. The system allows users to create reminders associated with specific locations and define a proximity radius. It continuously monitors the user's location and triggers notifications when the user enters the defined geofence region. The application is developed using Flutter for cross-platform mobile development and Firebase for backend services, ensuring scalability and real-time data synchronization.The proposed system integrates multiple components, including location tracking, distance calculation, and notification services, to provide an efficient and user-friendly solution. Additionally, optimization techniques such as periodic location updates are employed to reduce battery consumption while maintaining accuracy.

2.Related Works

Various approaches have been proposed in the domain of reminder systems and location-based services to improve task management and user productivity. Traditional reminder applications primarily rely on time-based notifications, which trigger alerts at predefined times without considering the user's geographic context. While these systems are effective for scheduling tasks, they often fail to provide meaningful reminders for activities that depend on location, thereby reducing their practical usefulness in real-world scenarios.

Several studies have explored the integration of Global Positioning System (GPS) technology and geofencing techniques to develop location-aware reminder systems. These systems utilize geographic boundaries to trigger notifications when a user enters or exits a specified region. Such approaches enhance the relevance of reminders by incorporating spatial context. However, many existing implementations suffer from challenges such as high battery consumption due to continuous location tracking, limited accuracy in dense urban environments, and dependency on stable internet connectivity.

Recent research has also focused on improving the efficiency of location-based systems through optimized tracking mechanisms and intelligent notification strategies. Techniques such as periodic location updates and adaptive distance calculation methods have been introduced to reduce resource consumption while maintaining accuracy. Additionally, map-based interfaces have been widely adopted to improve usability and enable users to interactively define reminder locations.

Despite these advancements, several limitations still exist in current systems. Many applications lack customization features such as adjustable geofence radius and category-based organization of reminders. Furthermore, some systems do not support offline notification mechanisms, which affects reliability in low-connectivity environments.

The proposed GeoReminder system addresses these limitations by integrating efficient GPS-based tracking, customizable geofencing, and offline notification capabilities within a user-friendly mobile application.

3.System Design

The proposed system, **GeoReminder**, is designed to provide intelligent location-based reminders using GPS and geofencing techniques. The system architecture consists of multiple stages, including user input, location tracking, data processing, proximity detection, and notification triggering. The overall workflow ensures that reminders are delivered based on the user's real-time geographic location rather than fixed time intervals.

3.1.Dataset Description

The system does not rely on traditional datasets but instead uses real-time user-generated data. The data consists of reminder details such as location coordinates (latitude and longitude), reminder name, category, and proximity radius. User information and reminder data are securely stored in a cloud database using Firebase Firestore. Each reminder is associated with a specific user ID, ensuring data isolation and security. The data structure reflects real-world usage scenarios, where users dynamically create and manage reminders based on their needs.

3.2.Pre-processing

Data preprocessing is performed to ensure accuracy and consistency of the stored information. User inputs such as location coordinates and radius values are validated before being stored in the database. Location data is formatted into standard geographic coordinates, and invalid or duplicate entries are handled appropriately. This step ensures that the system processes only valid and meaningful data, improving the reliability of proximity detection and notification triggering.

3.3 Usage Analysis

In this stage, the system continuously monitors the user's real-time location using GPS services. The application periodically retrieves the user's coordinates and compares them with stored reminder locations. A distance calculation mechanism is applied to determine the proximity between the user and each reminder location. This allows the system to evaluate whether the user has entered the defined geofence region. The use of periodic tracking helps balance accuracy andbatteryefficiency.

3.4 Machine Learning Model

The system uses a distance-based method to detect user proximity instead of machine learning. The distance between the user and reminder location is calculated using the Haversine formula.

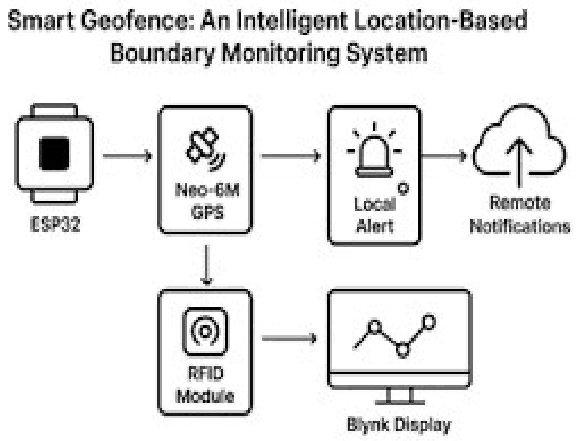
Based on this distance:

- Inside radius → Trigger notification
- Outside radius → No action

3.5 Decision-Making Process

A rule-based decision-making mechanism is implemented to trigger notifications. When the proximity condition is satisfied, the system generates a notification alert using local notification services. To prevent repeated alerts, a control mechanism is used to limit multiple triggers within a short time interval. The system ensures that notifications are delivered only when necessary, improving user experience

4.System Architecture



5.Algorithm Selection

The selection of an appropriate algorithm is essential for accurately detecting user proximity and triggering location-based reminders. Unlike traditional data-driven systems, the proposed GeoReminder application does not rely on labeled datasets or machine learning models. Instead, it focuses on real-time location tracking and efficient distance calculation to determine whether a user has entered a predefined geographic region.

Since the system operates on continuous geographic data, a computationally efficient and accurate distance calculation method is required. Among various approaches, mathematical models for calculating distance between two coordinate points are most suitable. These methods ensure fast computation and minimal resource consumption, which is critical for mobile applications.

After evaluating different techniques, a distance-based proximity detection approach is selected for this system. This approach calculates the distance between the user's current location and stored reminder coordinates, enabling precise identification of geofence boundaries. The method is simple, efficient, and well-suited for real-time applications.

5.1 K-Means Clustering

The GeoReminder system uses the Haversine formula, a widely accepted mathematical model for calculating the shortest distance between two points on the Earth's surface using latitude and longitude coordinates. This formula is particularly effective for GPS-based applications as it accounts for the curvature of the Earth.

$$d = 2R \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

Where:

- d= distance between two points
- R= radius of the Earth
- ϕ_1, ϕ_2= latitudes
- $\Delta\phi$= difference in latitude
- $\Delta\lambda$= difference in longitude

In this system, each reminder is defined using geographic coordinates (latitude and longitude) along with a specified radius. The user's current location is periodically retrieved using GPS services, and the distance between the user and each stored location is calculated using the Haversine formula.

If the calculated distance is less than or equal to the predefined radius, the system determines that the user has entered the geofence region. This condition acts as a trigger for generating a notification.

The Haversine-based approach is selected because it is computationally efficient, easy to implement, and provides sufficient accuracy for real-world mobile applications. It does not require training data or complex models, making it ideal for lightweight systems.

To enhance performance, the system uses periodic location updates instead of continuous tracking, thereby reducing battery consumption. Additionally, a rule-based mechanism is integrated with the distance calculation to control repeated notifications and improve user experience.

Overall, the use of the Haversine formula enables accurate proximity detection and supports reliable, real-time decision-making in the GeoReminder system.

6. System testing and maintenance

System testing and maintenance are essential phases to ensure that the proposed GeoReminder application functions accurately, efficiently, and reliably in real-world scenarios. Testing is performed to verify that all modules operate as intended and that location-based notifications are triggered correctly based on user proximity.

The testing process begins with **unit testing**, where individual components such as authentication, location tracking, distance calculation, and notification modules are validated independently. Each module is tested to ensure correct functionality and error handling. For example, the distance calculation logic is verified using known geographic coordinates to ensure accuracy.

After validating individual modules, **integration testing** is conducted to ensure smooth interaction between system components. This includes verifying the communication between GPS services, database storage, and notification triggering mechanisms. The system ensures that location data flows correctly from tracking modules to the proximity detection component.

The complete system is then tested under real-world conditions using different locations and scenarios. The application successfully tracks user movement, calculates proximity, and triggers notifications when the user enters the defined geofence region. The results are verified to ensure consistency and correctness of alerts.

Performance testing is carried out to evaluate the efficiency of the system, particularly in terms of battery consumption and response time. By using periodic location updates instead of continuous tracking, the system demonstrates optimized performance with minimal resource usage.

Maintenance plays a crucial role in ensuring long-term reliability.

- **Corrective maintenance** is performed to fix bugs and errors identified after deployment.
- **Adaptive maintenance** allows the system to support new features and updates in mobile platforms.
- **Perfective maintenance** focuses on improving performance and user experience.
- **Preventive maintenance** includes updating dependencies and monitoring system behavior to avoid potential failures.

Overall, system testing and maintenance ensure that the GeoReminder application remains stable, efficient, and reliable for continuous usage.

7. System Implementation

The implementation of the GeoReminder application is carried out using modern mobile development technologies and tools. The system is developed using **Flutter** and **Dart**, which enable cross-platform mobile application development with a responsive user interface. The development environment is set up using Visual Studio Code, which provides efficient tools for coding, debugging, and project management. The implementation of the proposed SaaS wastage detection system is carried out using Python and its associated libraries for data processing, machine learning, and visualization. The backend of the system is implemented using **Firebase**, which provides services such as user authentication and real-time database management through Cloud Firestore. Google Maps API is integrated to provide interactive map-based location selection and visualization

During implementation, the system integrates multiple components, including user input handling, location tracking, proximity detection, and notification services. The application retrieves the user's real-time location using GPS and processes it periodically. A distance calculation mechanism is applied to determine the proximity between the user and stored reminder locations..

Based on the calculated distance and predefined radius, the system identifies whether the user has entered a geofence region. When this condition is satisfied, a notification is triggered using local notification services. This ensures that reminders are delivered even without active internet connectivity

The implementation also includes a rule-based control mechanism to avoid repeated notifications within short intervals, thereby improving user experience. Reminder data, including location coordinates and user details, are securely stored and managed using Firebase Firestore.

The final stage of implementation focuses on the user interface, where the application provides an interactive and intuitive experience. Users can add reminders by selecting locations on a map, adjust radius using sliders, and manage reminders through a list view.

Overall, the implementation successfully integrates mobile development, cloud services, and location-based technologies to provide an efficient and practical solution for context-aware reminder systems.

8. Conclusion

The proposed GeoReminder application provides an effective solution for improving traditional reminder systems by incorporating location-based intelligence through GPS and geofencing techniques. The system continuously monitors user location, processes geographic data, and triggers context-aware notifications when users enter predefined regions. This approach ensures that reminders are delivered at the right place and time, thereby increasing their relevance and usefulness in real-world scenarios.

One of the key advantages of the system is its ability to automate reminder triggering based on user proximity, reducing dependency on manual time-based inputs. The application utilizes efficient distance calculation methods and optimized location tracking to ensure accurate performance while minimizing battery consumption. By integrating features such as customizable geofence radius, real-time tracking, and offline notification support, the system provides a comprehensive and user-friendly solution for task management.

The implementation of the GeoReminder system enhances user productivity by ensuring that important tasks are not missed due to lack of contextual awareness. The use of an interactive map-based interface further improves usability, allowing users to easily define and manage reminder locations. Additionally, the integration of cloud services ensures scalability, data security, and real-time synchronization across devices.

Overall, the proposed system demonstrates that location-based technologies can significantly improve the effectiveness of reminder applications. It provides a practical, scalable, and efficient solution that can be applied in real-world environments. Future enhancements may include the integration of artificial intelligence for predictive reminders, support for wearable devices, and advanced background tracking mechanisms to further enhance system capabilities.

9.References

D. A. Godse, S. Anjankar, P. Ganga and B. Swapnali,
"Android Based Location Reminder Using Geofencing,"
Journal of Emerging Technologies and Innovative Research (JETIR), vol. 6, no. 4, 2019.

S. Arfi and S. Rajesh,
"Location Based Alarm on Android,"
International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT), vol. 6, no. 3, 2020.

P. A. Lahare and S. A. Thakare,
"GeoPrompt: A Mobile Applications Reminder System Based on Location,"
Journal of Mobile Computing, Communications & Mobile Networks, vol. 11, no. 03, 2024.

N. Prasad and S. Swetha,
"GeoFencing: Location Based Services,"
International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT), vol. 5, no. 2, 2019.

J. Hightower and G. Borriello,
"Location Systems for Ubiquitous Computing,"
IEEE Computer Magazine, 2001.

G. Madhusudan and Saakshi,
"Notify: A Novel Location-Based Shopping Reminder App,"
International Journal for Research in Applied Science and Engineering Technology (IJRASET), 2025.

K. Michael and L. McNamee,
"Location-Based Services and Geo-Fencing: Privacy and Security Considerations,"
IEEE Technology and Society Magazine, 2016.

A. Küpper,
Location-Based Services: Fundamentals and Operation,
Wiley Publishing, 2005.

Google Developers,
"Geofencing API for Android,"
2023.

Google Developers,
"Location Services API,"
2023.

