A Location – Based Alarm System for Android Devices

Kunal M Suryawanshi Author
Department of Computer Engineering,
Nanasaheb Mahadik Polytecnic
Institute,Peth , Maharashtra, India
kunalsuryawanshi2608@gmail.com

Suyash J Jadhav Author Department of
Computer Engineering, Nanasaheb Mahadik
Polytecnic Institute,Peth , Maharashtra, India
suyash3324@gmail.com

Sahil D Patil Author Department of
Computer Engineering, Nanasaheb
Mahadik Polytecnic Institute,Peth ,
Maharashtra, India
Sahilpatil9033@gmail.com

Dnyaneshwar S Patil Author Department of
Computer Engineering, Nanasaheb Mahadik
Polytecnic Institute,Peth , Maharashtra, India
dnyneshwar1045@gmail.com

Priyanka D .Jadhav Author
Department of Computer Engineering
Nanasaheb Mahadik Polytecnic
Institute,Peth , Maharashtra, India
priyajadhav2593@gmail.com

Fahim A Shaikh Author Department of
Computer Engineering Nanasaheb Mahadik
Polytecnic Institute,Peth , Maharashtra, India
Kfan41003@gmail.com

ABSTRACT

This work presents the location and Android- based, location-aware reminder application that focuses on alarms when a user approaches or reaches a target destination without using any commercial or paid web services. This applicat ion combines Android built-in location and geofencing features, also the free nomination serv ices, and it uses OpenStreetMap data. This lets users search for places by name, find their des tination, and set an alarm directly on the map.

The custom Nominatim service makes sure your app follows Nominatim rules. The app is all owed to send only one request each second to avoid overloading the free Nominatim services. If the user makes many requests quickly, the app puts them in a line and processes them one by one instead of flooding the services. When the request fails (for example, due to a network error), the app waits a short time before retrying. The app doubles the waiting time with each attempt so it does not keep hitting the services too many times.

The app uses a three-level backup system to make the location search always work smoothly. First, it tries the Nominatim geocoding service, and if it fails or slows down, it automatically switches to an Indian location API; finally, it uses Android's built-in geocoder on the device it self. This service allows the app to continue working even when the main service is unavailab le or overloaded. Each location alarm is linked with a geofence.

Introduction

Location-
based reminder and geo alarm applications have gained significant importance for users such as commuters, travelers, and logistics personnel, as these systems provide alerts triggered by geographic context rather than fixed schedules. Such functionality enables timely and situatio n-aware notifications when users approach or depart predefined locations, improving conveni ence and task efficiency. However, many exisitig commercial solutions rely on proprietary m apping platforms and paid geocoding services, which often impose licensing fees, request quo tas, and vendor dependencies. These constraints make them less suitable for academic project s, experimental research systems, and cost-sensitive deployments.

To address these limitations, this work presents the design and implementation of a location- based alarm system for Android devices that operates entirely without paid web APIs. The pr oposed solution utilizes open-
source geocoding resources based on OpenStreetMap, particularly Nominatim, while strictly complying with its acceptable-use policies. An intermediate service layer is introduced to ma nage request-rate control, queuing, retry mechanisms, and structured error handling, thereby c onverting raw network responses into meaningful application-level information. This approac h enhances reliability and user feedback while respecting public service constraints.

Furthermore, the system incorporates a multi-level fallback strategy by integrating region-
specific geocoding services and the native Android Geocoder to improve availability under v arying network conditions. On the client side, Android's fused location provider and geofenci ng mechanisms are employed to monitor user movement in an energy-
efficient manner. Alarms are triggered based on configurable distance thresholds and estimate d arrival metrics, allowing users to track progress toward their destinations and receive timely alerts. The proposed architecture demonstrates that a robust and user-friendly location alarm application can be developed using open technologies while remaining compliant with servic e usage policies.

Literature review

- Early location-based reminder systems (pre-2015) relied on basic GPS with Google Maps for single-trigger alerts at predefined spots, but suffered from limited reminders, poor scalability, and proprietary API dependencies that incurred costs or quotas
unsuitable for academic projects

- Patil and Pawar (2015) introduced an Android reminder using location awareness,
combining time/location triggers for multiple tasks, yet still dependent on commercial services limiting open deployment.

- Singh et al. (2017) developed a task reminder app with GPS tracking, improving UI but retaining reliance on paid geocoding, highlighting needs for quota-free
alternatives like your Nominatim integration.
- Kumar and Sharma (2019) proposed geofencing-based alarms on Android for entry/exit notifications, emphasizing power efficiency over continuous GPS— mirroring your fused location provider—but lacking distance/ETA feedback and open- source search compliance.

- Nevon Projects' mobile location alarm (recent) uses GPS for user tracking and
nearest-place alarms, similar to your set-alarm module, but without rate-limiting or multi-tier fallbacks, risking service overload unlike your queued Nominatim handler.

- IJARIIT's location-based alarm (recent) stresses reminders for location-specific tasks, aligning with your geofence alarms, though it omits adaptive retries and regional API backups for network resilience.

- IEEE's alarm reminder (2022) targets vicinity alerts without paid services, akin to your OpenStreetMap focus, but evaluates only basic proximity without your three-level geocoding (Nominatim → Indian API → Android Geocoder).

Proposed System/Methodology

The proposed Location Alarm system is implemented as an Android application, using XML for user interface design and Java for application logic. The methodology follows a client- side, event-driven architecture and is divided into four logical stages: location acquisition, geocoding wit h fallback handling, geofence creation with alarm association, and background monitoring. W hen a user enters a destination or selects a location, the request is processed by a custom geoc oding service to obtain geographic coordinates. After confirmation, the destination and alarm configuration parameters are stored locally, and the system prepares to monitor the user's mov ement relative to the selected location.

For location tracking, the application uses Android's fused location provider, which combines GPS, Wi-Fi, and cellular network data to achieve an optimal balance between accuracy and battery con sumption. Location updates are requested at adaptive intervals based on user movement, and are used both for displaying the current position and for computing distance to the target locat ion. Text-based location search is handled through an open-source geocoding approach based on OpenStreetMap data, with strict enforcement of usage p olicies through rate limiting, request queuing, retry mechanisms, and proper client identificati on. To improve reliability, a multi-tier fallback strategy is employed, switching to alternative regional services or the device's na tive Android Geocoder when the primary service is unavailable.

Once a destination is finalized, a circular geofence is created using the Android Geofencing API with a user-defined radius. The system triggers alarms when a geofence entry event is detected, or when t he computed distance falls below the configured threshold. In parallel, the application estimat es remaining distance and time-to-arrival using successive location updates and speed information, providing continuous progre ss feedback to the user. Background components ensure that notifications and alarms are deli vered even when the application is not active. Throughout the process, structured error handli ng is applied to geocoding and location failures, enabling clear user feedback while maintaini ng system robustness and compliance with open-service policies

Implementation

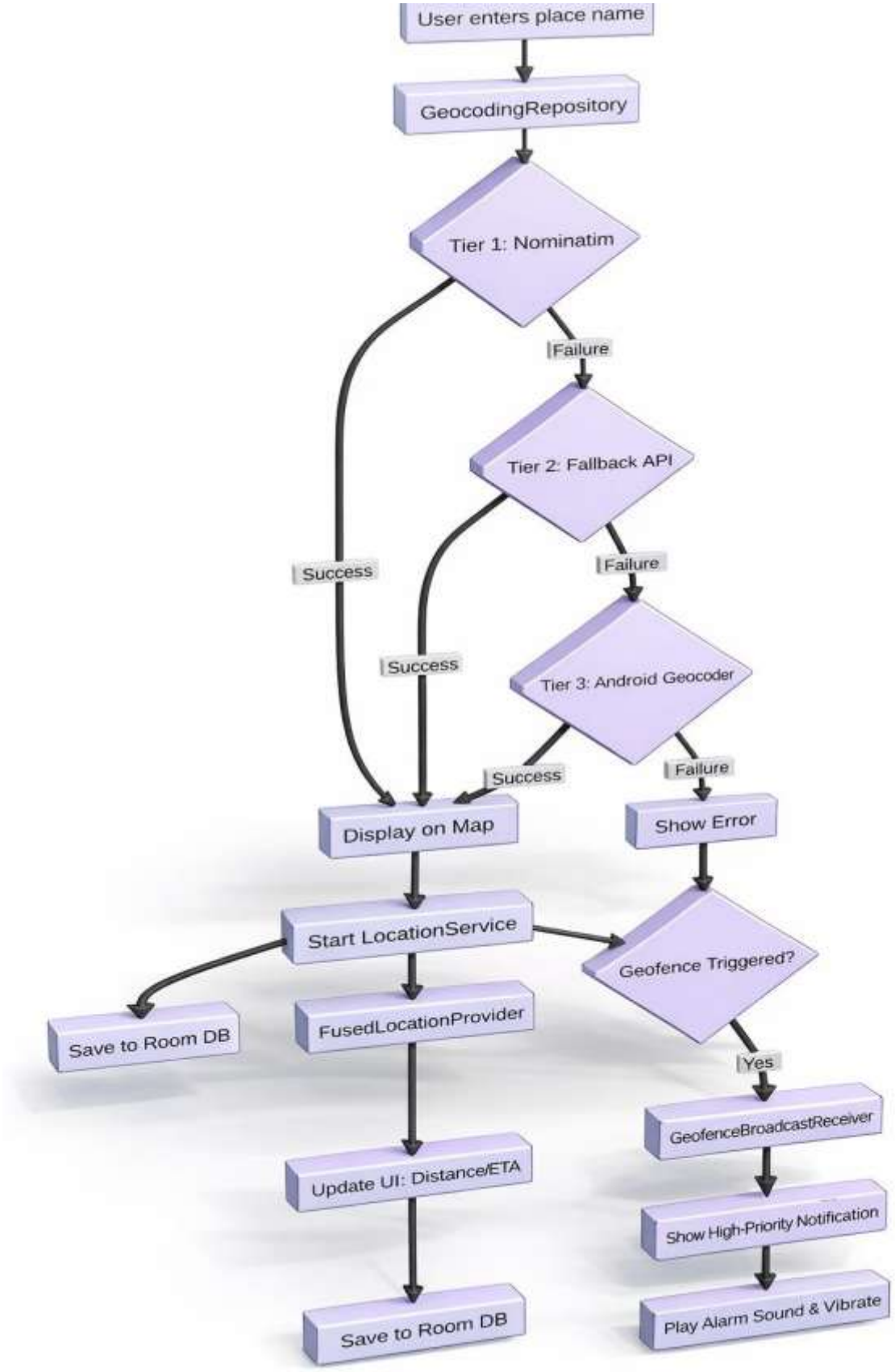
The Location Alarm application is developed for the Android platform, using XML to design the user interface and Java to implement the application logic. The user interface consists of multiple screens that allow users to search for locations, view selected destinations on a map, configure alarm settings, and manage active alarms. XML layouts define components such as the search bar, map view, and alarm configuration forms, while Java activities and fragments handle user interactions, validate inputs, and pass selected parameters to background services. This layered design separates presentation logic from core functionality, improving maintain ability and clarity.

For location tracking and alarm triggering, the application integrates Android's fused location provider and geofencing APIs. The fused provider supplies location updates by combining G PS, Wi-Fi, and cellular data to achieve accurate positioning with efficient battery usage. Based on use r movement, the system requests location updates at adaptive intervals suitable for travel scen arios. Once a destination is confirmed, a circular geofence is created using the Geofencing Cli ent API, defined by the destination's coordinates and a user-selected radius.

Working

The Location Alarm application operates through a structured workflow that combines user i nput, open geocoding, geofencing, and background location monitoring to deliver timely alert
s. When the user enters a destination name, the application resolves it into geographic coordin ates using a rate-limited open geocoding service, with multiple fallback options to ensure reli ability. After the destination is confirmed, the system allows the user to configure alarm para meters and creates a virtual geofence around the selected location. The application then conti nuously tracks the user's position using an energyefficient location provider and computes bot h the remaining distance and estimated time-to-
arrival. As the user approaches the destination, the system evaluates geofence entry events an d distance thresholds in the background. Once the predefined condition is satisfied, the applic ation triggers a high-
priority notification and alarm sound, ensuring alert delivery even when the application is not active. This end-
to-end workflow enables a fully automated, reliable, and user-friendly location-based alarm w ithout dependence on paid services.

Architecture Diagram



Result And Discussion

The Location Alarm system was evaluated in real travel scenarios, such as bus and train journeys, where users configured alarms for stops, stations, and landmarks at different
distances. In most test cases, the system triggered alarms accurately upon geofence entry or slightly before reaching the destination, providing sufficient reaction time. Distance and
estimated time-to-arrival information was generally reliable for walking and moderate vehicle speeds, with minor deviations observed in dense urban areas or locations with reduced GPS accuracy. The use of open geocoding with a three-tier fallback strategy significantly reduced visible search failures, ensuring that valid locations could be obtained even under network
instability or rate-limit constraints. Battery performance remained efficient due to the use of fused location services and geofencing, and alarm notifications were delivered reliably in the background when recommended Android execution policies were followed. Overall, the
results demonstrate that a carefully engineered, open-data-based approach can deliver a
practical and responsive location alarm solution comparable to commercial systems, while remaining suitable for cost-constrained and academic environments, with scope for future improvements in route-based arrival estimation and adaptive location updates.

Future Scope

1. Enhanced Time-to-Arrival Estimation:

Future versions can improve arrival accuracy by integrating multi-modal travel information, such as public transport schedules and route-based estimation, enabling more precise alerts for bus and train commuters.

2. Cloud Synchronization and Multi-Device Support:

Alarm data can be synchronized across multiple devices using cloud storage solutions,
allowing users to back up, restore, and manage alarms seamlessly when switching or losing devices.

3. Adaptive and Intelligent Geofencing:

Machine learning techniques may be applied to automatically adjust geofence radii and
location update intervals based on individual travel patterns, improving both accuracy and battery efficiency.

3.Voice and Wearable Integration:

Integration with voice assistants and wearable devices can enable hands-free alarm configuration and discreet alert delivery, enhancing usability in hands-busy or accessibility-focused scenarios.

Conclusion

This work presents a reliable Location Alarm application for Android that delivers location- based reminders without relying on paid or proprietary services. By combining Android's
fused location and geofencing APIs with a policy-compliant open geocoding approach and intelligent fallback handling, the system achieves robust performance under real-world
conditions. Experimental evaluation demonstrates accurate alarm triggering, low geocoding failure rates, and efficient battery usage, making the solution suitable for cost-sensitive and academic deployments. The results confirm that open-data services, when carefully
engineered, can support practical and scalable location-aware applications, with future enhancements offering further scope for improvement.

References

[1] Nominatim Project, OpenStreetMap Nominatim Geocoding Service. Accessed December 2025. [Online]. Available: https://nominatim.org.

[2] Google Android Developers, Create and Monitor Geofences, 2025. Accessed December 2025.[Online]. Available: https://developer.android.com/develop/sensors-and-location/location/geofencing

[3] OpenStreetMap Contributors, Nominatim Documentation - OpenStreetMap Wiki, 2025 . Accessed December 2025.[Online].Available:https://wiki.openstreetmap.org/wiki/Nominatim

[4] OpenStreetMap Foundation, Nominatim Acceptable Use and Usage Policy. [Online]. Available: https://operations.osmfoundation.org/policies/nominatim

[5] Google Android Developers, Fused Location Provider API, 2025. [Online]. Available: https://developers.google.com/location-context/fused-location-provider

[6] S. R. Patil and S. S. Pawar, "Design and implementation of a location-based reminder system using Android," International Journal of Engineering Research and Management, vol. 5, no. 10, pp. 21-25, 2015.

[7] R. K. Singh, A. Verma and P. Sharma, "An Android-based task reminder system using location awareness," International Research Journal of Engineering and Technology (IRJET), vol. 4, no. 4, pp. 879-882, 2017.

[8] A. Kumar and R. Sharma, "Geofencing-based location reminder application on Android," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 6, no. 4, pp. O25- O30, 2019.