iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72150

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Notify: A Novel Location-Based Shopping Reminder App

Dr. G Madhusudan[1], Saakshi[2]
Department of Computer Science and Engineering JSS Science and Technology University, Mysuru, India

Abstract: ***The escalating reliance on mobile technologies has revolutionized the management of daily tasks, yet traditional tools such as note-taking and task-list applications frequently fall short in delivering contextually relevant support, partic- ularly for shopping activities. Cognitive overload, exacerbated by the absence of geospatial intelligence in these tools, often leads to overlooked purchases despite proximity to pertinent retail locations. This paper introduces an innovative solution through the development of an advanced, location-aware mobile application that harnesses geo-fencing[1] technology to provide sophisticated, context-driven shopping reminders. By analyzing user mobility patterns, the application delivers precise location-specific, store-targeted notifications, thereby optimizing efficiency and alleviating mental burden. This research addresses the defi- ciencies of current commercial solutions by introducing a user- oriented framework for geospatial task management, ultimately transforming the shopping experience within an increasingly mobile and fast-paced global context.***

Index Terms: Contextual awareness, Location based re- minders, OpenStreetMap, geo-fencing[1], Nominatim API, Haversine formula[2], notification systems

I. INTRODUCTION

The widespread integration of mobile technologies into daily life has transformed the management of routine tasks, with shopping presenting a critical area for technological advancement. Despite the abundance of digital aids, such as task organizers and note-keeping applications, individuals frequently struggle to recall necessary purchases. This issue arises from the lack of contextual awareness in conventional tools, which do not adapt to users' real-time proximity to relevant retail settings. For instance, a person might plan to purchase headphones, a non-urgent but necessary item, from a specific electronics shop, yet forget when passing by due to preoccupation with other tasks, as no alert con- nects their intent to their location. Such lapses underscore a broader issue of cognitive saturation in today's fast-paced, data-intensive world, where conventional digital tools lack the dynamism required for location-responsive task facilitation. Advancements in geospatial technologies, particularly geo- fencing[1], offer a compelling solution to these deficiencies. By establishing virtual boundaries around physical locations, geo-fencing[1] enables applications to initiate actions when users cross these thresholds. Although certain commercial tools utilize rudimentary geo-fencing[1] for broad notifica- tions, they often lack the precision to deliver store-focused prompts tied to user-specified retail locations. This deficiency calls for an advanced, user-oriented solution that integrates geospatial precision with purchasing intentions. This study introduces a novel mobile application, Notify, engineered to revolutionize the shopping experience through context-aware, geo-fenced[1] notifications. Users can designate a shop by searching its name or pinpointing its location and associate specific items to be purchased from that store. The application retrieves the shop's geocode[3] coordinates to enable precise proximity monitoring. By tracking the user's location in real time, it determines the distance to the designated store and, upon entering a predefined geo-fenced[1] perimeter, triggers a context-specific notification detailing items from the user's predefined shopping list. This seamless integration of spatial awareness and individual purchasing goals mitigates cognitive overload, enhances transactional efficiency, and sets a new standard for task management in a mobile-centric era. Notify serves as a robust solution for alleviating the challenge of forgetting necessary but non-urgent purchases, empowering users to act on their shopping intentions with precision and ease.

II. RELATED WORK

Location-based reminder systems have advanced with mo- bile technology, particularly through the use of geo-fencing[1]. However, most existing applications and research do not cater specifically to shopping-related needs.

A. Shortcomings in Existing Applications

Popular apps like Google Keep offer location-based re- minders but often trigger them inconsistently, reducing reli- ability for time-sensitive shopping tasks. Neplarm allows geo- fenced[1] alerts but lacks item-level features and has poor notification handling. Wake Me There reliably detects location changes but also misses item-list integration, forcing users to rely on memory or separate tools. These apps lack the depth needed for store-specific, item-aware reminders.

©IJRASET: All Rights are Reserved

B. Academic Contributions and Gaps

Research has explored geo-fencing[1] across diverse con- texts, offering insights into its potential and limitations for reminder systems. A study on a workforce tracking app leverages GPS to monitor employee movements within virtual boundaries, triggering alerts for task allocation and emer- gencies [4]. However, its focus on workforce management lacks user-driven reminders or item-level tracking, limiting its relevance to shopping applications. Another investigation ap- plies geo-fencing[1] in mobile apps for delivering promotional ads in shopping centers, using location data to send targeted notifications [5]. While effective for advertising, this approach does not support personalized shopping reminders, focusing on business-driven promotions. Another study proposed a mobile communication-based reminder system using cellular networks and SMS to send location-aware alerts [6]. Although this early framework highlights the potential of mobile networks for reminders, it lacks item-specific task tracking and user cus- tomization, essential for modern shopping-oriented systems. These studies underscore geo-fencing's[1] versatility but reveal a gap in addressing shopping-specific, user-centric needs.

C. Notify's Contribution

These studies and applications highlight a key gap: the absence of item-specific, store-aware reminder systems. Notify addresses this by allowing users to associate detailed shopping lists with specific shop locations and receive timely, actionable notifications when nearby. This fills the usability void left by existing solutions, combining geo-fencing[1] precision with shopping convenience.

III. SYSTEM DESIGN

The Notify mobile application adopts the Model-View- Controller (MVC) architecture to deliver context-aware, geo- fenced[1] shopping reminders, addressing limitations in ex- isting tools by integrating store-specific notifications. This section outlines the system's architecture, comprising the Data Layer, Service Layer, and User Interface, and describes their interactions for proximity-based task management.

A. Architectural Overview

Notify's MVC architecture ensures modularity and scal- ability. The Model manages structured data for shops and items, supporting persistent storage. The View provides user interfaces for data input and task management. The Controller coordinates location tracking, geo-fencing[1] logic, and noti- fication delivery, ensuring seamless integration across layers.

B. System Components

Data Layer (Model): The Data Layer handles persistent storage and data structuring for shops and items: Structured Storage: A lightweight local database stores shop attributes, including name, geographic coordinates, and associated items, supporting basic CRUD operations. Shop Entity Representa- tion: A unified data structure represents each shop, encap- sulating metadata for consistent system-wide interaction and efficient location-based queries. Service Layer (Controller): The Service Layer governs dynamic operations: Geolocation Service: Interfaces with geocoding[3] services to convert shop names into coordinates and monitors the user's real- time location to calculate proximity to shops. Proximity- Based Trigger Logic: Defines geo-fenced[1] perimeters around shops, triggering notifications when users enter these regions. Notification Service: Generates shop-specific reminders upon proximity detection, incorporating a delay mechanism to pre- vent frequent alerts and reduce user disruption. User Interface Layer (View): The User Interface ensures intuitive interaction: Startup Interface: Provides a smooth launch experience, ini- tializing necessary services. Task Creation Interface: Allows users to input shop details, locations, and items, with validation to ensure completeness. Overview and Management Interface: Displays saved shops and items, enabling users to view, edit, or delete entries. Interactive Map Interface: Supports shop location selection via search or manual map markers for precise positioning.

C. Workflow Overview

The workflow starts with users entering shop details and selecting locations via the Task Creation and Interactive Map Interfaces. The Data Layer stores this information, while the Geolocation Service tracks user proximity. Upon entering a geo-fenced[1] region, the Proximity-Based Trigger Logic checks for recent notifications, and if none exist, the Notifica- tion Service issues a reminder listing associated items. Users can manage tasks via the Overview Interface. This flow is illustrated in Fig. 1.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

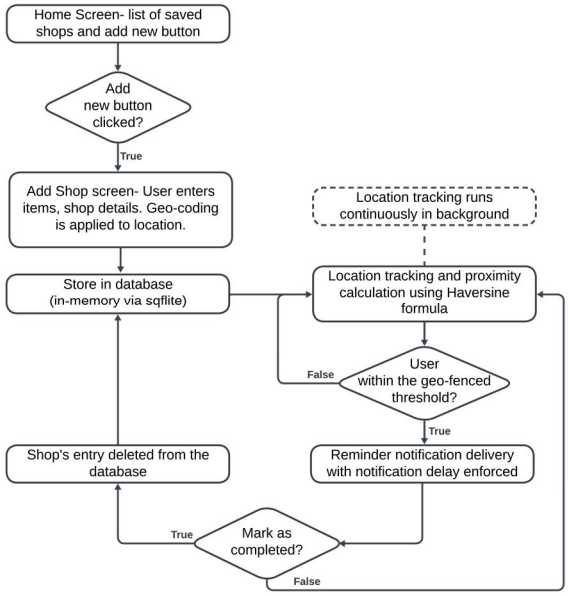


Fig. 1: Workflow of the Notify App

D. Discussion

This design framework ensures a clear division of re- sponsibilities and emphasizes context-driven functionality. The use of geo-fenced[1] triggers and shop-specific reminders addresses key usability concerns found in traditional task- reminder systems. By leveraging location-awareness, modular data handling, and user-centric design, the Notify system aims to reduce the mental load on users by offering timely and relevant prompts during real-world shopping activities.

IV. METHODOLOGY

This section details the implementation of the Notify mobile application, focusing on the realization of its MVC architecture to deliver geo-fenced[1] shopping reminders. The approach emphasizes accurate location tracking, efficient data manage- ment, and user-centric notification delivery.

A. Data Layer Implementation

Database Module: A local SQLite database stores shop entries in a table with fields for name, geocoordinates (latitude, longitude), and items. CRUD operations enable data manage- ment, with indexing on coordinates for efficient proximity queries. Shop Model: Defines shop attributes, supporting seri- alization/deserialization for SQLite compatibility and ensuring data integrity across application states.

B. Service Layer Implementation

Geolocation Service: Uses device GPS to track user loca- tion, updated periodically to balance accuracy and battery use. The OpenStreetMap (OSM) API retrieves shop coordinates via search or manual selection. Proximity is calculated using the Haversine formula [2]:

$$a = \sin^2(\Delta\phi/2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\Delta\lambda/2) \quad (1)$$

$$c = 2 \cdot 2(\sqrt{a}, \sqrt{1-a}) \quad (2)$$

$$d = R \cdot c \quad (3)$$

where: R is the radius of the sphere (Earth), typically 6371 km. $\Delta\phi$ is the difference in latitudes. $\Delta\lambda$ is the difference in longitudes. ϕ_1 and ϕ_2 are the latitudes of the two points. c is the angular distance in radians. d is the distance in the same units as

R (e.g., kilometers). A 500-meter geo-fencing[1] radius has been implemented to trigger notifications. Notifica- tion Service: Queries the database for items upon entering a geo-fenced area, delivering alerts via the system's notification manager with options to view or dismiss. Notification Man- ager: Implements a 60-second delay between alerts for the same shop using timestamps, preventing notification fatigue.

C. User Interface Implementation

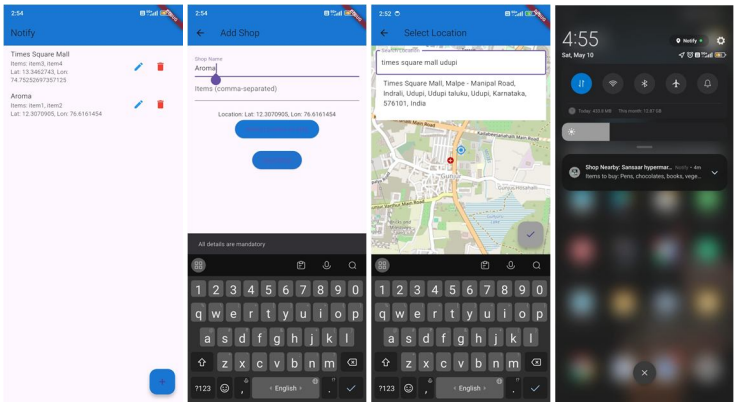
Task Creation Interface: Enables input of shop details with validation for non-null fields, integrating with the Geolocation Service for location selection. Interactive Map Interface: Uses the Nominatim API for search-based geocoding and supports manual pin placement for custom locations. Overview and Management Interface: Displays shops and items, allowing edits or deletions synced with the database. Startup Interface: Facilitates initialization, transitioning to the main interface.

D. API Integration and Development Considerations

The OSM and Nominatim APIs enable geocoding[3], with asynchronous requests for reliability. Notify was developed using Flutter for Android, optimizing location tracking with a 10-meter distance filter to ensure that location updates are triggered only when the user moves significantly, conserving battery life. Privacy is ensured through local data storage and explicit location permissions.

V. RESULTS

This section presents the outcomes of Notify's implemen- tation, demonstrating its ability to deliver context-aware, geo- fenced[1] shopping reminders. The application was tested on a mobile device, and key functionalities are illustrated through the following screenshots.



(a) Home Screen (b) Add Shop Screen (c) Map Screen (d) Notification from App

Fig. 2: (a) Home Screen, (b) Add Shop Screen, (c) Map Screen, (d) Notification from app

Figure 2(a) displays the Home Interface, which presents a list of saved shops with their metadata, including names and associated items, facilitating efficient task management. It also includes a button for adding new shopping details. Figure 2(b) shows the Add Shop Interface, where users enter mandatory details such as shop name, location, and items, ensuring data integrity. Figure 2(c) illustrates the Map Interface, enabling users to search for shops using the Nominatim API or to pin locations on an interactive map for precise positioning of shops not available in the Nominatim API. Finally, Figure 2(d) depicts a notification successfully delivered upon entering a geofenced[1] region, detailing the items to be purchased and validating Notify's core functionality. The notification includes an action associated with the shop, allowing users to mark the task as completed, which removes the shop's entry from the stored data.

VI. FUTURE WORK

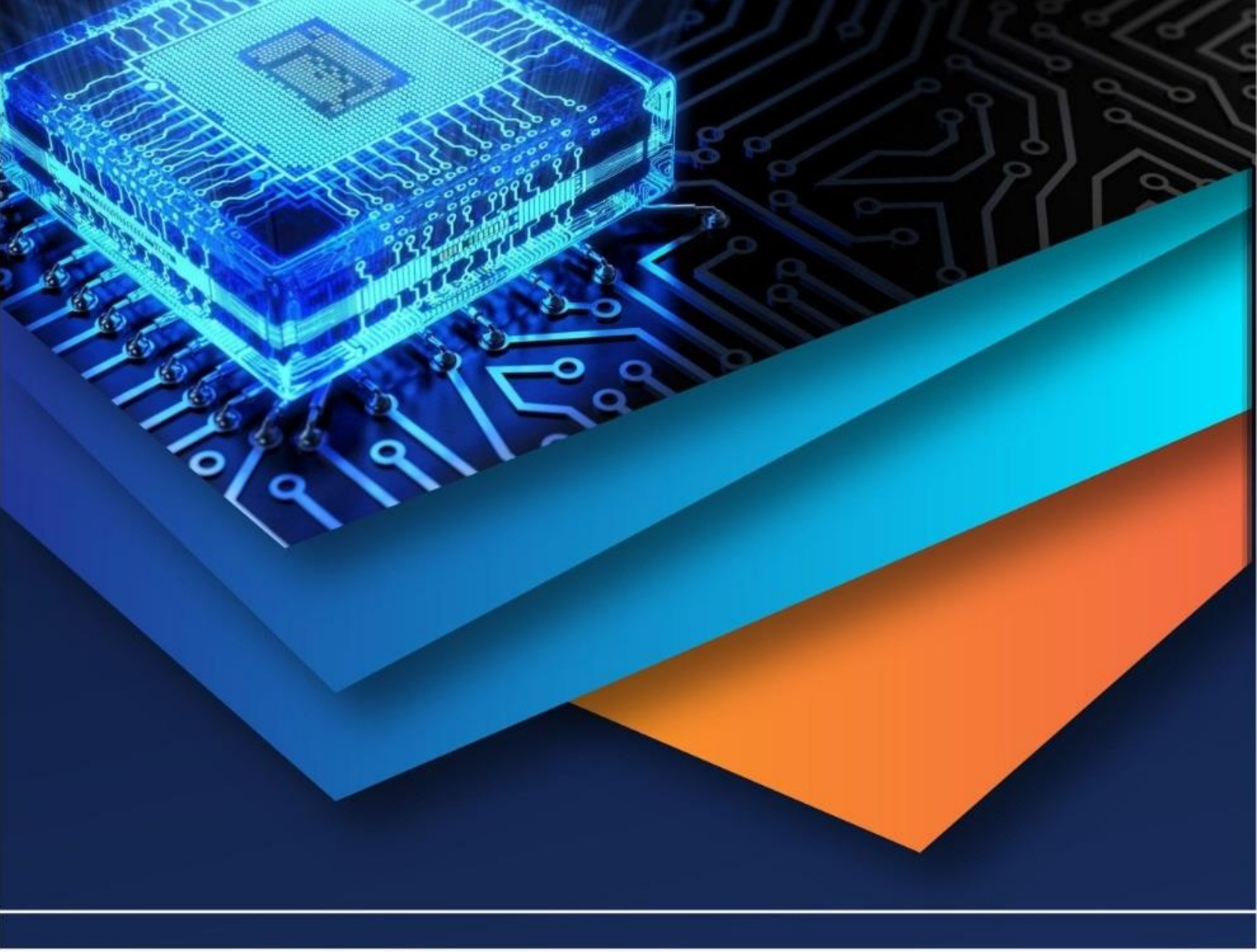
While Notify effectively delivers location-based, shop- specific reminders, future enhancements can significantly im- prove its impact. Background location tracking, even when the app is terminated, would ensure that reminders function consistently at all times. A "snooze" option would allow users to temporarily dismiss alerts, reducing notification fatigue near multiple shops or during inconvenient moments. A checklist system for items will enable users to mark purchased items while retaining unpurchased ones for future alerts, streamlin- ing the shopping process.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Additionally, lightweight machine learning models may be used to optimize reminder timing by learning user patterns such as frequent store visits and preferred shopping hours, enhancing adaptability while main- taining local data privacy and avoiding external dependencies.

REFERENCES

[1] Hightower, J., & Borriello, G. (2001). Geofencing: Engineering design and methodology. Retrieved December 05, 2024, from https://www.researchgate.net/publication/44260944 Geofencing Engineering Design and Methodology
[2] Trigo sphe´rique: Routes maritimes antiques en Me´diterrane´e. (n.d.). Ancient Ports. Retrieved December 23, 2024, from https://www.ancientportsantiques.com/wp-content/uploads/Documents/ ETUDESarchivees/MedNavigationRoutes/MedNav/TrigoSpherique.pdf
[3] Goldberg, D. W., Wilson, J. P., & Knoblock, C. A. (2007). From text to geographic coordinates: The current state of geocoding. URISA Journal, 19(1), 33–46. Retrieved December 2, 2024, from https://johnwilson.usc.edu/wp-content/uploads/2016/05/ 2007-Goldberg-Knoblock-Wilson-JURISA.pdf
[4] Ismailkhapathan, A., Nugroho, D. F., & Nugraha, I. G. B. G. (2024). Android geofencing app. Retrieved December 09, 2024, from https:// www.researchgate.net/publication/382086365 Android Geofencing App
[5] Susanto, Y., Sari, M. R. N., & Agustin, L. (2023). The use of geofencing in Android-based mobile applications for promotional ads in shopping centers. Retrieved December 09, 2024, from https://www.researchgate.net/publication/373695691 The Use of Geofencing in Android-Based Mobile Applications for Promotional Ads in Shopping Centers
[6] Pawar, V. R., Devale, P. R., & Mane, D. T. (2010). Mobile communication for reminder system. Retrieved December 09, 2024, from https://www.researchgate.net/publication/43655839 Mobile Communication for Reminder System



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)