International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 11 | Nov 2026 www.irjet.net p-ISSN: 2395-0072

Food Delivery Application with Real-Time Tracking

Yogeshwari Kale[1], Kalyani Bargal[2], Prof. S. A. Chechare[3]

[1,2] Student, [3]Project Guide, Department of Computer Science and Engineering, ICEEM, Chhatrapati Sambhajinagar, Maharashtra, India

---

Abstract - *Online food ordering has become a convenient way for customers to discover restaurants, place orders, and receive meals at their preferred location. This paper proposes a Food Delivery Application with Real-Time Tracking that connects customers, restaurants, delivery partners, and an administrator through a common digital platform. The proposed system supports restaurant and menu browsing, cart management, order placement, order-status updates, delivery-partner assignment, and map-based tracking. A location service can periodically share the delivery partner's coordinates, allowing the customer to view the approximate delivery route and progress. The application is designed to improve transparency, reduce uncertainty during delivery, and simplify order coordination. The paper describes the system architecture, major modules, proposed methodology, functional requirements, and testing approach. The system is presented as a proposed academic design: performance figures and implementation results are not claimed without actual testing.*

Key Words: Food Delivery, Real-Time Tracking, GPS, Google Maps API, Online Ordering, Mobile Application, Order Management.

1. INTRODUCTION

Smartphones and internet services have changed how customers access food services. Traditional phone-based ordering may require repeated calls to confirm menu availability, order status, and delivery time. A digital food delivery application can combine these activities in one interface and maintain a shared order record for customers, restaurants, and delivery partners.

The proposed application enables customers to view restaurants, select food items, submit orders, and monitor the order journey. Restaurants receive order details and update preparation status. Delivery partners receive assigned deliveries and share location information while an order is in transit. An administrator oversees users, restaurants, and order records.

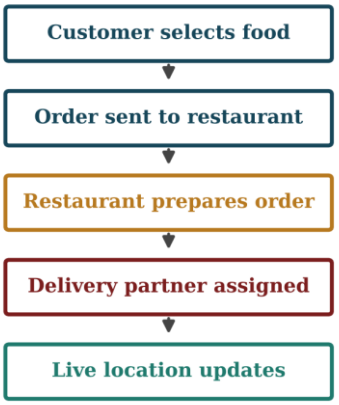

Fig-1: Proposed application workflow

2. PROBLEM STATEMENT

In many ordering workflows, customers cannot easily determine whether an order has been accepted, prepared, picked up, or is approaching the delivery address. Separate communication between customers, restaurants, and delivery partners can cause confusion and delays. The proposed system addresses these issues by keeping order status and delivery location information accessible from a single application.

3. OBJECTIVES

The main objective is to design a user-friendly food ordering platform with delivery progress visibility. The specific objectives are:

- to provide secure user access;
- to display restaurant menus and prices;
- to support cart and order management;
- to share order-state notifications;
- to enable location-based tracking during delivery; and
- to provide administrative control over key records.

4. LITERATURE SURVEY

Existing food ordering platforms demonstrate the value of digital menus, online payment options, notifications, and delivery tracking. Mapping services provide route visualization and location-based features, while mobile notification services can communicate order-state changes

[1], [2].

Common design considerations include location permission, network availability, data privacy, accurate status transitions, and usability on different screen sizes [3], [4]. This proposal combines these established concepts into a modular academic system built on accepted software-engineering practice [5], [6].

5. PROPOSED SYSTEM

The proposed system follows a client-server architecture. The customer application communicates with a backend service to authenticate users, retrieve restaurant and menu data, create orders, and obtain status updates. Restaurant and delivery-partner interfaces use role-based access. A database stores user profiles, restaurant details, menu items, order records, delivery assignments, and location updates. A map service displays the delivery partner's most recent available coordinates (Fig-2).

5.1 Main Modules

Customer module: registration/login, restaurant discovery, menu browsing, item selection, cart, address management, order placement, status view, and delivery tracking.

Restaurant module: restaurant profile, menu and availability management, order acceptance, preparation status, and handover confirmation.

Delivery partner module: assigned order list, pickup confirmation, location sharing during an active delivery, and delivery completion.

Administrator module: user and restaurant management, order monitoring, issue handling, and basic reports.

5.2 System Architecture

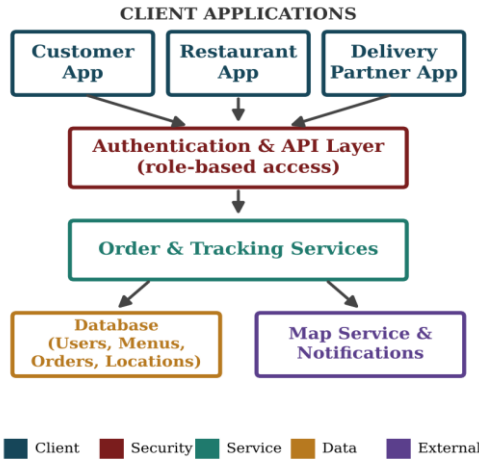

Fig-2: High-level architecture of the proposed system

6. REAL-TIME TRACKING APPROACH

When a delivery partner accepts an assigned order, the application requests location permission and obtains location updates while the delivery is active. Each update may include latitude, longitude, timestamp, and order identifier. The backend validates that the update belongs to an active delivery and makes the latest location available to the authorized customer. The customer interface places the latest coordinate on a map and refreshes the displayed marker as new updates arrive.

Tracking should be enabled only when required for an active delivery and should stop after completion or cancellation. If location permission is denied, GPS is unavailable, or connectivity is lost, the application should display the last update time and an understandable status rather than implying that the marker is live. Location access should be restricted to relevant users and protected from unauthorized viewing [3], [7].

6.1 Order State Sequence

Order Placed → Accepted → Preparing → Ready for Pickup → Picked Up → Out for Delivery → Delivered. The system should also support cancellation or rejection where applicable. State changes must be validated to avoid contradictory order states.

7. PROPOSED METHODOLOGY

- **Requirement analysis:** identify the needs of customers, restaurants, delivery partners, and administrators.
- **Interface design:** prepare simple screens for restaurant lists, cart, order history, and map tracking.
- **Data design:** define entities and relationships for users, restaurants, food items, orders, and location updates.
- **Service development:** implement authentication, order creation, assignment, and tracking APIs.
- **Map integration:** display location updates using a mapping service such as Google Maps Platform, subject to API configuration and usage terms [1], [2].
- **Testing:** verify normal workflows, error handling, permissions, and network interruptions.

8. FUNCTIONAL REQUIREMENTS

The system should allow users to sign in, browse restaurants, search menus, add items to a cart, submit an order, and review order history. Restaurant staff should be able to accept orders and update preparation stages. Delivery partners should see only assigned deliveries and share their location during active trips. Administrators should manage relevant application records. The customer should see the latest actual status and tracking timestamp.

9. NON-FUNCTIONAL REQUIREMENTS

- **Usability:** screens and messages should be easy to understand.
- **Performance:** menus and order details should load within a reasonable time under supported network conditions.
- **Reliability:** failed requests should be handled with clear messages and safe retries.
- **Security:** authentication, role-based authorization, input validation, and protected communication should be used.
- **Privacy:** personal addresses and location data should be collected only when needed and exposed only to authorized participants.
- **Scalability:** services and database structures should support growth in restaurants and orders.

10. DATABASE DESIGN

A relational or document-oriented database may be used depending on the implementation needs. Table-I describes the proposed data model.

TABLE-I: Proposed database entities

Entity	Key attributes
User	user_id, name, email/phone, password hash, role
Restaurant	restaurant_id, name, address, service status
MenuItem	item_id, restaurant_id, item name, description, price, availability
Order	order_id, customer_id, restaurant_id, total, address, current status, created_at
OrderItem	order_id, item_id, quantity, unit price
Delivery	delivery_id, order_id, partner_id, pickup time, completion time
LocationUpdate	delivery_id, latitude, longitude, timestamp

11. USER INTERFACE DESIGN

The customer home screen can present restaurant cards, cuisine categories, search, and a cart shortcut. A restaurant page lists available items and prices. The order screen shows a clear status timeline. During delivery, the tracking screen displays a map, delivery-partner marker when available, last-updated time, and order summary (Fig-3). Restaurant and delivery dashboards should prioritize pending tasks and clear status controls.

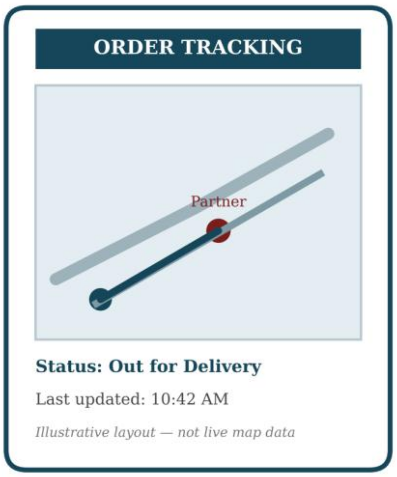

Fig-3: Illustrative tracking-screen layout; not live map data

12. TESTING STRATEGY

Testing should be carried out after implementation using representative test accounts and a controlled test environment. The cases in Table-II are proposed test cases, not recorded execution results.

TABLE-II: Proposed test cases

Test ID	Scenario	Expected result
TC-01	Valid user login	User reaches home screen
TC-02	Add item to cart	Item and price appear correctly
TC-03	Place valid order	Order ID and initial status are shown
TC-04	Restaurant updates status	Customer sees the new status
TC-05	Location permission denied	Tracking limitation is explained
TC-06	Location update received	Authorized map view refreshes
TC-07	Network unavailable	Last update time and retry guidance appear
TC-08	Unauthorized user requests order	Access is denied

13. EXPECTED OUTCOMES AND LIMITATIONS

The proposed system is expected to give customers clearer visibility of order progress, reduce repeated status enquiries, and help restaurants and delivery partners coordinate tasks. Real-time tracking quality depends on device GPS, internet connectivity, location permissions, backend update frequency, and mapping-service availability. Estimated arrival times should be treated as estimates rather than guarantees. The application does not eliminate delays caused by traffic, weather, restaurant preparation, or unavailable delivery partners.

No measured accuracy, delivery-time reduction, or user-satisfaction result is reported because no verified implementation measurements were supplied. Such metrics should be added only after conducting documented tests with appropriate consent and a defined evaluation method.

14. SECURITY AND PRIVACY CONSIDERATIONS

The application should store password hashes rather than plain-text passwords, validate all incoming data, and apply role-based authorization to order and location endpoints [4]. Network traffic should use secure transport. Sensitive values such as phone numbers, delivery addresses, and coordinates should not be exposed in public links or logs. Location updates should be retained only for a justified period, with access limited to the customer and delivery workflow. Users should receive a clear explanation of when tracking is active.

15. ADVANTAGES

- One platform for ordering and order-status visibility.
- Clearer communication among customers, restaurants, and delivery partners.
- Map-based view of the delivery partner's latest available location.
- Structured records for menus, orders, and delivery states.
- Potential to reduce repetitive calls asking for delivery updates.
- Modular design that can be extended with online payments, ratings, and notifications.

16. CONCLUSION

This paper proposed a Food Delivery Application with Real-Time Tracking to bring restaurant discovery, order placement, order-status management, and delivery location visibility into one coordinated system. The design separates customer, restaurant, delivery-partner, and administrator functions and describes a location-update workflow for active deliveries. The proposed requirements and test cases provide a foundation for implementation and evaluation. The system's practical effectiveness must be established through development, usability testing, security checks, and measured trials under realistic network and delivery conditions.

17. FUTURE SCOPE

Future versions may include:

- online payment gateway integration;
- push notifications;
- delivery-time estimation based on route and traffic data;
- restaurant ratings and coupons;
- multilingual interfaces and accessibility improvements; and
- analytics for restaurants.

Additional safeguards could include anomaly detection for location updates and automated deletion of old location records. Any prediction feature should be evaluated against actual delivery data before being presented as reliable.

18. REFERENCES

[1] Google Maps Platform, “Maps JavaScript API Documentation,” Google Developers. https://developers.google.com/maps/documentation/javascript

[2] Google Maps Platform, “Geolocation and location-related APIs,” Google Developers. https://developers.google.com/maps

[3] Android Developers, “Request location permissions.” https://developer.android.com/develop/sensors-and-location/location/permissions

[4] OWASP Foundation, “OWASP Mobile Application Security.” https://owasp.org/www-project-mobile-app-security/

[5] I. Sommerville, Software Engineering, 10th ed., Pearson, 2015.

[6] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed., McGraw-Hill, 2019.

[7] Android Developers, “Location services.” https://developer.android.com/develop/sensors-and-location

BIOGRAPHIES

PROF. S.A. CHECHARE
ASSISTANT PROFESSOR (HOD) DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING INTERNATIONAL CENTER OF EXCELLENCE IN ENGGINERING & MANAGEMENT (ICEEM)

MS. YOGESHWARI KALE
Pursuing (CSE) ICEEM INTERNATIONAL CENTER OF EXCELLENCE IN ENGGINERING
& MANAGEMENT (ICEEM)

MS. KALYANI BARGAL
Pursuing (CSE) INTERNATIONAL CENTER OF EXCELLENCE IN ENGGINERING
& MANAGEMENT (ICEEM)