

## Problem Statement

Small and medium sized businesses , particularly in Ethiopia, face significant challenges in efficiently handling deliveries for their products. Many businesses rely on informal delivery methods, such as manually coordinating with drivers or delivery personnel through phone calls.

### 1. Lack of Pricing Stability

- Delivery prices are often subject to unpredictable fluctuations. This is due to the reliance on private motorcycle owners who negotiate prices directly with businesses. This inconsistency creates challenges for businesses trying to provide reliable and affordable delivery services.
- **Businesses are forced to pass these fluctuating costs to customers, leading to dissatisfaction and reduced trust in their service.**
- **End users (customers receiving the deliveries) may experience higher or unpredictable fees, especially during peak hours or in less accessible areas.**

### 2. Lack of Accountability for Product Safety

- Current non-formal delivery providers often lack formal mechanisms for addressing damages to goods during transit. Drivers, many of whom are independent contractors, are generally unable to reimburse businesses or customers for damages to high-value items.
- **Customers face disappointment and financial loss when products arrive damaged, negatively affecting their trust in the delivery provider and the business.**
- **Businesses risk losing customers and revenue due to an inability to guarantee the safe delivery of goods.**

### 3. Difficulty for Restaurants to Retain Customers on Food Delivery Apps

- The rise of food delivery platforms in Ethiopia has significantly benefited the restaurant industry. However, these platforms often run exclusive deals and promotions with specific restaurants, which continuously redirect customers toward the best available offer. This competition makes it challenging for restaurants to maintain consistent performance and customer loyalty on these platforms.

- Restaurants struggle to retain customers who are incentivized to switch based on deals rather than brand loyalty.
- A restaurant's visibility on these platforms often depends on participating in costly promotions or paying higher platform fees, affecting their profit margins.

#### 4. Barriers to Building Independent Delivery Infrastructure

- To address the issue of reliance on food delivery apps, restaurants may attempt to shift customers to their own platforms by creating a custom app and delivery infrastructure. However, building and managing this system is prohibitively challenging for most businesses due to:
  - **High Costs:** Managing call centers, hiring drivers, and purchasing, maintaining or repairing vehicles requires significant capital.
  - **Skill and Time Requirements:** Building and maintaining an infrastructure, training staff, and ensuring operational efficiency requires a specialized team and considerable time investment.
- Restaurants are forced to remain dependent on delivery platforms, losing control over customer experience and pricing.
- Small and medium-sized restaurants are disproportionately affected, as they often lack the resources to invest in independent solutions, leaving them at a competitive disadvantage.

#### 5. Lack of Brand Identity and Direct Customer Engagement for Businesses

- By relying on third-party delivery platforms, restaurants and other businesses lose the opportunity to build a direct relationship with their customers. The third-party platform becomes the face of the transaction, while the restaurant's or business's brand identity diminishes in the eyes of the consumer.
- **Limited Customer Loyalty:** Since customers associate the transaction with the delivery platform rather than the restaurant, there's minimal brand recognition or loyalty built for the business.
- **Loss of Customer Data:** Delivery platforms often retain valuable customer insights, such as purchasing behavior, preferences, and feedback. Businesses miss the opportunity to use this data to improve services or target customers directly.
- **Dependency on Platform Algorithms:** Businesses are at the mercy of algorithms that determine visibility on the platform, leading to inconsistent customer acquisition and retention.

### Solution Statement

**Liyt** offers a comprehensive delivery management solution tailored to the needs of businesses in Ethiopia. By providing an easily integrable API and operational delivery

infrastructure, Liyt enables businesses to focus on their core operations while delivering products efficiently and reliably to their customers.

### 1. Addressing Lack of Pricing Stability

Liyt standardizes delivery pricing through a structured system that leverages a network of trained and vetted drivers.

- **Fair and Transparent Pricing:** Liyt eliminates price fluctuations by establishing a consistent pricing model based on distance and time taken to cover the distance.
- **Cost Control for Businesses:** Businesses can rely on predictable delivery costs, improving their financial planning and avoiding the need to pass unpredictable fees to their customers.
- **Improved Customer Satisfaction:** End users benefit from stable delivery fees, increasing trust and loyalty toward businesses.

### 2. Ensuring Accountability for Product Safety

Liyt emphasizes the secure and reliable transportation of goods through formalized agreements and processes with its delivery personnel.

- **Dispute Resolution Mechanism:** In the event of damages, Liyt offers a transparent claims process to reimburse businesses or customers, reducing financial risk and maintaining trust.
- **Reliability:** Businesses and customers gain confidence knowing Liyt holds drivers accountable for the safety of their products.

### 3. Solving Customer Retention Issues for Restaurants

Liyt empowers restaurants to maintain their brand presence and loyalty by acting as a delivery partner, not a competitor.

- **Neutral Platform:** Unlike traditional food delivery apps, Liyt doesn't promote or favor any restaurant. Businesses retain full control over their customer experience.
- **Equal Opportunity for All Businesses:** Without special deals or promotions overshadowing other restaurants, all businesses can thrive equally based on service and quality.

### 4. Streamlined Manual Order Creation

Liyt provides an easy-to-use platform for businesses operating on Telegram, Instagram, or other online platforms to manually input customer orders and initiate deliveries.

- **User-Friendly Dashboard:** Businesses can quickly log orders with delivery details directly into Liyt's platform, reducing the hassle of managing phone calls or spreadsheets.

- **Scheduled or Immediate Deliveries:** Businesses can choose to dispatch orders instantly or schedule deliveries for later, offering flexibility to meet customer needs.

## Code and Project Files Documentation

To ensure clarity and ease of understanding, we have included descriptive comments in the codebase and linked essential files below for quick reference:

- Access Controller: Manages user authentication and access control across the application.
- Orders Controller: Handles the creation and management of orders outside the API namespace.
- API V1 Orders Controller: Exposes RESTful endpoints for developers that let's them initialize and create orders.
- API Key Model: Responsible for managing API keys, including generation, validation, and association with users or organizations.
- User Model: Defines the user structure, including attributes such as `primary_address` and associations with API keys and orders.
- **Order Service**: Handles order-related logic such as geocoding, price calculation, and status updates.
  - Exposes methods to accept and complete orders, retrieve orders by driver or user, and calculate delivery routes.
- 1. **Token Service**: Manages JWT-based authentication, including token issuance and decoding.
  - Retrieves a secure secret key for encoding tokens.
- All core functionalities are thoroughly tested using RSpec. Tests are organized under the `/spec` directory, with each service or functionality having a corresponding `_spec.rb` file. Example:
  - `access_controller_spec.rb`: Tests for user access and authentication features.
  - `orders_controller_spec.rb`: Validates order creation, retrieval, and error handling scenarios.
  - `api_key_spec.rb`: Ensures proper API key generation, validation, and management.

# Gebeta API Usage in the Hackathon Project

## 1. Geocoding (Location Retrieval)

All methods can be seen in:

[https://github.com/Asrat77/liyt\\_api/blob/master/app/services/order\\_service.rb](https://github.com/Asrat77/liyt_api/blob/master/app/services/order_service.rb)

**Method:** `get_location(name)`

**When:**

- Used when a location name is provided to determine its geospatial coordinates.

**Where:**

- Endpoint: `https://mapapi.gebeta.app/api/v1/route/geocoding`
- Query Parameters:
- name - The location name.
- apiKey - Authorization key from environment variables.

**How:**

- Uses the HTTParty gem to send a GET request to the Gebeta Maps geocoding endpoint.
- Response provides details like latitude and longitude of the requested location.

## 2. Distance and Time Calculation

**Method:** `get_distance_and_time(origin, destination)`

**When:**

- Invoked during order pricing and routing tasks to calculate travel distance and estimated time between two locations.

**Where:**

- Endpoint: `https://mapapi.gebeta.app/api/v1/route/driving/direction`
  - Query Parameters:
  - la1, lo1 - Latitude and longitude of the origin.
  - la2, lo2 - Latitude and longitude of the destination.
  - apiKey - Authorization key from environment variables.

**How:**

- Sends a GET request and extracts the totalDistance (in meters) and timetaken (in seconds) from the response.

## 3. Directions Retrieval

**Method:**

- `get_directions(origin, destination)`
- **When:** Used to generate step-by-step driving directions for delivery routes.

**Where:**

- Endpoint: Same as distance calculation:  
`https://mapapi.gebeta.app/api/v1/route/driving/direction`
- Query Parameters: Same as above.

**How:**

- Sends a GET request and retrieves the direction field from the API response, which contains the detailed driving instructions.

#### **4. Delivery Price Calculation**

**Method:** `calculate_price(origin, destination)`

**When:**

- Every time an order is created to determine the total delivery price based on distance and time.

**Where:**

- Combines data from the `get_distance_and_time` and `get_directions` methods.

**How:**

- Applies a rate per kilometer (`rate_per_km`) and time rate (`time_rate`) to calculate the price dynamically.