IJVAST | Vol. 2, Issue 2 | June 2026 | Open Access — Peer Reviewed Research Paper

Porter Delivery and Logistics Platform

Vaishanvi Patil[1], Amol Navale[2]

Department of Computer Application

Agasti Institute of Management, Computer Application and Research (AIMCAR), Akole, Maharashtra, India

Abstract—Intra-city goods transportation, spanning small-parcel courier delivery, two-wheeler and mini-truck freight, and last-mile logistics for local businesses, has traditionally relied on informal, phone-based coordination between customers and independent transporters, resulting in inconsistent pricing, limited vehicle availability visibility, and no real-time shipment tracking. This paper presents the design and development of a Porter-style Delivery and Logistics Platform that connects customers requiring on-demand goods transportation with a network of registered delivery partners, supporting real-time vehicle-type selection, fare estimation, driver-partner matching, live shipment tracking, and digital payment processing. The system was developed using a three-tier architecture with a relational and geospatial database backend, a server-side application layer implementing partner-matching logic, dynamic fare calculation, and real-time location tracking via WebSocket-based communication, and separate mobile-responsive interfaces for customers and delivery partners. Key features include vehicle-type-based fare estimation, proximity-based driver-partner matching, live GPS shipment tracking, in-app digital payment, and an administrative dashboard for partner onboarding and platform analytics. The system was evaluated through functional testing across core booking and delivery workflows, simulated driver-matching performance testing, and a usability survey administered to sample customer and delivery-partner users. Results show that the proximity-based matching algorithm assigned an available delivery partner within an average of 2.4 seconds under simulated load, achieved a 94.6% successful first-match rate without requiring re-assignment, and received favorable usability ratings from both customer and delivery-partner user groups. These findings indicate that a purpose-built, real-time matching and tracking platform can meaningfully improve the efficiency, transparency, and reliability of intra-city goods delivery and logistics services.

Index Terms*—Delivery Platform, Logistics, Real-Time Tracking, Driver-Partner Matching, Fare Estimation, Geospatial Database, WebSocket Communication, On-Demand Transportation.*

I. INTRODUCTION

Local goods transportation needs, ranging from small-parcel courier delivery to mini-truck and two-wheeler freight for individuals and small businesses, have historically been served through informal, phone-based coordination with independent transporters. This approach suffers from inconsistent pricing, limited visibility into vehicle availability, no standardized real-time tracking, and reliance on personal networks or word-of-mouth referrals to locate available transport.

On-demand logistics platforms, following the operational model popularized by ride-hailing and courier-aggregator applications, address these limitations by digitally connecting customers with a network of registered delivery partners, providing transparent, algorithmically computed fare estimates, automated proximity-based partner matching, and real-time shipment tracking throughout the delivery journey.

This paper presents the design, development, and

C. Real-Time Location Tracking Architectures

Real-time shipment and vehicle tracking is commonly implemented using persistent WebSocket connections or similar publish-subscribe messaging architectures, enabling low-latency location updates to be pushed to customer and administrative interfaces without repeated client polling.

III. SYSTEM DESIGN

A. System Architecture

The platform was designed using a three-tier architecture comprising a mobile-responsive presentation layer (separate customer and delivery-partner interfaces), an application layer (server-side booking logic, matching engine, fare calculation, and real-time communication handling), and a data layer combining a relational database for user and transaction records with geospatial indexing for partner location data.

B. User Roles

The system supports three primary roles: Customer, evaluation of a Porter-style Delivery and Logistics Platform supporting vehicle-type-based booking, real-time driver-partner matching, live shipment tracking, and digital payment, intended to demonstrate the core technical architecture underlying on-demand intra-city logistics platforms.

II. LITERATURE REVIEW

A. On-Demand Logistics Platform Models

Literature on on-demand logistics and ride-hailing platforms commonly identifies three core technical challenges: efficient real-time matching of demand (customer requests) to supply (available partners), dynamic and transparent fare computation, and reliable real-time location tracking, each requiring careful architectural design to support responsiveness at scale.

B. Driver-Partner Matching Algorithms

Proximity-based matching algorithms, typically leveraging geospatial indexing structures such as geohashing or quad-trees, are widely used to efficiently identify the nearest available partner to a given pickup location, balancing matching speed against fairness in partner assignment across a service area.

who books deliveries, tracks shipments, and makes payments; Delivery Partner, who receives booking assignments, updates delivery status, and shares live location; and Administrator, who manages partner onboarding, monitors platform activity, and reviews analytics.

C. Driver-Partner Matching Engine

The matching engine identifies available delivery partners within a configurable radius of the customer's pickup location using geospatial proximity queries, ranks candidate partners by a combination of distance and partner rating, and dispatches the booking request to the top-ranked available partner, with automatic reassignment to the next candidate if the request is not accepted within a configured timeout.

D. Fare Estimation and Payment

Fare estimation is computed based on vehicle type, estimated distance, and estimated trip duration using a configurable base-fare-plus-per-kilometer pricing model, displayed to the customer prior to booking confirmation. Digital payment is processed through an integrated third-party payment-gateway API supporting card, UPI, and wallet-based payment methods.

An overview of the platform's core functional modules is presented in Table I.

TABLE I — Core Functional Modules of the Delivery and Logistics Platform

Module	Primary User	Key Functionality
Booking & Fare Estimation	Customer	Select vehicle type, view fare estimate, and confirm booking
Driver-Partner Matching	System (automated)	Identify and assign nearest available delivery partner
Live Tracking	Customer / Delivery Partner	Real-time shipment location display via WebSocket updates
Delivery Status Updates	Delivery Partner	Update booking status (accepted, picked up, delivered)
Payment Processing	Customer	Digital payment via card, UPI, or wallet
Partner Onboarding & Analytics	Administrator	Manage partner registration and view platform analytics

IV. IMPLEMENTATION

The system was implemented as a web and mobile-responsive application, with the application layer built using a server-side framework handling booking requests, matching logic, and fare computation, and a WebSocket-based real-time communication layer used to push live location updates and booking-status changes to connected customer and delivery-partner clients without requiring repeated polling.

Partner location data was indexed using a geospatial query structure supporting efficient nearest-partner lookups, with periodic location

B. Matching Performance Testing

Driver-partner matching performance was evaluated using a simulated environment with a configurable number of virtual available partners distributed across a service-area grid, measuring average time-to-match and first-match success rate (bookings matched without requiring reassignment) across repeated simulated booking requests.

C. Usability Evaluation

A usability survey based on the System Usability Scale (SUS) was administered separately to sample customer users and delivery-partner users following a supervised trial period, covering booking-flow clarity, updates transmitted from the delivery-partner mobile interface at short intervals during an active delivery to maintain tracking accuracy while managing data transmission overhead.

V. EVALUATION METHODOLOGY

A. Functional Testing

Functional testing was conducted across core workflows including booking creation, vehicle-type selection and fare estimation, partner matching and assignment, live tracking display, delivery-status updates, and payment processing, using both valid and boundary-condition test scenarios.

VI. RESULTS AND DISCUSSION

A. Functional Testing Results

All core functional test cases across booking, matching, tracking, and payment workflows passed verification against expected outcomes, with no critical defects identified during the testing cycle. Functional test coverage results are summarized in Table II.

B. Matching Performance Results

The proximity-based matching engine assigned an available delivery partner within an average of 2.4 seconds under simulated load with 50 virtual available partners distributed across the service area, achieving a 94.6% first-match success rate, with the remaining bookings requiring a single reassignment cycle due to simulated partner non-response.

tracking-interface responsiveness, and overall satisfaction with each respective interface.

C. Usability Survey Results

The System Usability Scale survey yielded an average score of 83.7 out of 100 among sampled customer respondents and 79.5 out of 100 among sampled delivery-partner respondents, both indicating favorable usability, with customer feedback particularly positive regarding fare-estimate transparency and delivery-partner feedback favorable toward booking-notification clarity.

.VII. COMPARATIVE SUMMARY

Table III presents a summary of matching-engine performance across varying simulated partner-availability densities, illustrating how time-to-match and first-match success rate are influenced by the number of available delivery partners in the service area, providing a practical reference for partner-network capacity planning

VIII. CHALLENGES AND LIMITATIONS

Challenges encountered during development included designing a matching algorithm that balances assignment speed against fairness in partner workload distribution, ensuring location-tracking accuracy while managing mobile battery and data-usage overhead, and handling edge cases such as partner cancellation after assignment or intermittent connectivity during an active delivery. Matching-performance evaluation was conducted in a simulated environment rather than under live operational conditions with real delivery partners, and further field evaluation would strengthen confidence in the reported performance figures.

TABLE II — Functional Test Coverage Summary

Test Case Category	Test Cases Executed	Passed	Failed
Booking creation and fare estimation	11	11	0
Driver-partner matching and assignment	13	13	0
Live tracking display	8	8	0
Delivery-status updates	9	9	0
Payment processing	10	10	0

TABLE III — Matching Engine Performance Across Partner Availability Density

Available Partners (Simulated)	Avg. Time-to-Match (s)	First-Match Success Rate (%)
10	5.8	81.2
25	3.6	89.7

Available Partners (Simulated)	Avg. Time-to-Match (s)	First-Match Success Rate (%)
50	2.4	94.6
100	1.7	97.3

IX. FUTURE SCOPE

Future enhancements could include dynamic, demand-based fare adjustment during peak booking periods, route-optimization algorithms for multi-stop delivery bookings, integration of proof-of-delivery capture through photo or digital-signature confirmation, and predictive partner-positioning based on historical demand patterns to reduce time-to-match in high-demand zones. Expansion to support scheduled/advance bookings and integration with business customer accounts for recurring logistics needs are also identified as valuable directions for continued development.

X. CONCLUSION

This paper presented the design, development, and evaluation of a Porter-style Delivery and Logistics Platform supporting vehicle-type-based booking, real-time proximity-based driver-partner matching, live shipment tracking, and integrated digital payment. Evaluation results showed reliable functional behavior across core workflows, an average matching time of 2.4 seconds with a 94.6% first-match success rate under simulated load, and favorable usability ratings from both customer and delivery-partner user groups. These findings support the practical viability of a purpose-built, real-time matching and tracking architecture for on-demand intra-city logistics platforms, with several promising directions identified for further system enhancement.

XI. REFERENCES

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

[2] Sommerville, I. (2015). Software Engineering (10th ed.). Pearson Education.

[3] Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.

[4] Elmasri, R., and Navathe, S.B. (2015). Fundamentals of Database Systems (7th ed.). Pearson Education.

[5] Ozdemir, S., and Hussain, S.A. (2017). Beginning Geospatial Analysis with Python. Apress.

[6] Fette, I., and Melnikov, A. (2011). The WebSocket Protocol. RFC 6455, Internet Engineering Task Force.

[7] Zheng, Y. (2015). Trajectory data mining: An overview. ACM Transactions on Intelligent Systems and Technology, 6(3), 1-41.

[8] Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences, 114(3), 462-467.

[9] Brooke, J. (1996). SUS: A quick and dirty usability scale. Usability Evaluation in Industry, 189(194), 4-7.

[10] Kroenke, D.M., and Auer, D.J. (2015). Database Processing: Fundamentals, Design, and Implementation (14th ed.). Pearson Education.

[11] Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. ACM SIGMOD Record, 14(2), 47-57.

[12] PCI Security Standards Council. (2018). Payment Card Industry Data Security Standard (PCI DSS) Requirements and Security Assessment Procedures, Version 3.2.1.