International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 6, Nov- Dec 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Smart Delivery System with Customer Time Selection

Dr. PR. Patil, Ashwini P. Wagh, Bharavi P. Patil, Pradyumna V. Mali,
(School of Computer Science Engineering, Sandip University, Trimbak Road, Nashik, Maharashtra, 422213, India
Email: ashwiniw584@gmail.com)

----------**********----------

Abstract:

The rapid growth in the field of e-commerce has increased the demand for delivery services that are swift and customer-friendly. Traditional systems usually operate within fixed schedules, which leads to missed deliveries, higher operational costs, and customer dissatisfaction. This paper presents a Smart Delivery System with Customer Time Selection that can take delivery time preferences from the customer while the customer places an order.

Advanced predictive analytics and route optimization algorithms are integrated with the system to ensure timely deliveries and reduce the distance traveled along with operational costs. The customers have options to choose convenient delivery windows, and the system dynamically assigns orders to various delivery agents depending on their availability, proximity, and route efficiency. The architecture will follow a modular approach: customer, delivery agent, and admin modules follow a centralized database and real-time monitoring. Design models, including use-case, class, sequence, and component-level diagrams, reveal the workflow and interactions between system components.

Simulation results show that the proposed system significantly enhances on-time delivery rates, reduces failed deliveries, and greatly improves customer satisfaction while optimizing resources. Besides, it contributes to green logistics by minimizing redundant delivery trips and fuel consumption. This is a scalable, adaptable framework for modern last-mile delivery operations, effectively merging customer convenience, operational efficiency, and environmental responsibility.

Keywords — **Smart Delivery System, Customer Time Selection, Last-Mile Delivery, Delivery Time Window, Route Optimization, Predictive Analytics, Machine Learning in Logistics, E-commerce Logistics, Dynamic Scheduling, Sustainable Delivery.**

----------**********----------

I. INTRODUCTION

The e-commerce boom has placed an unprecedented burden on last-mile delivery systems. Customers increasingly expect not just fast delivery, but flexible delivery times-the ability to choose a time window when they are available. Missed deliveries due to customer unavailability result in higher operational costs, wasted fuel, and degraded customer satisfaction. Traditional delivery systems often operate on fixed schedules or rigid routing strategies that fail to account for individual customer preferences and dynamic real-world conditions.

A smart delivery system with customer time selection bridges this gap by incorporating customer timeslot choice in the delivery scheduling process. Through this system:

It allows customers to choose delivery time windows at the time of order placing, hence enhancing customer experience and reducing failed deliveries.

Logistics providers can optimize routes not just on a distance or cost basis but on customer availability, therefore reducing idle waiting times, re-deliveries, and other inefficiencies.

Predictive analytics and real-time data, such as on traffic or vehicle location, may help dynamically adjust schedules and make operations more efficient and sustainable.

This system can thus lead to a win-win situation-higher customer satisfaction, a lower cost per delivery, with reduced environmental impact due to fewer redundant trips.

II. LITERATUE REVIEW

Various delivery systems, logistics optimization, and customer-centric services have been explored in several studies:

Dynamic Delivery Scheduling: The studies prove that when clients are given a chance to choose a delivery time slot, it helps to increase the rate of on-time deliveries and reduce cases of failed delivery attempts. Smith et al., 2022.

Route Optimization: Dijkstra's and A* algorithms have been widely performed to reduce delivery time and fuel consumption (Chen & Li, 2021).

Customer-centric logistics: Researchers have established that customized delivery services lead to increased customer loyalty and satisfaction.

Mobile Integration in Delivery Systems: Smartphone applications provide real-time tracking and allow customers to modify delivery times, hence improve transparency and engagement. Patel et al., 2019

IoT and Smart Logistics: Sensors and IoT devices enable the monitoring of the conditions of delivery and asset tracking for on-time and safe delivery. Zhang et al., 2021

Gap Identified: Most of the available systems are focused either on route optimization or on tracking, none effectively integrating customer time preferences into delivery scheduling. Our proposed system would fill that gap.

III. METHODOLOGY

Requirement Analysis

Identification of stakeholders: Customers, delivery agents, dispatchers/admin, and possibly system operators.

Functional requirements: customer registration/login, slot selection, order tracking, dynamic route updates, notification system.

Non-functional requirements: scalability, performance, fault tolerance (what if a driver cancels?), data security.

Data Collection & Preprocessing

Historical delivery data is required: timestamps, address, and successful/failed delivery.

Traffic data-if using real-time or historic traffic for route planning.

Customer behavior data: Which time slots are chosen, success rates, cancellations

Predictive Model for Time-Slot Assignment

Predicted Probability: Using a machine learning model (such as classification or regression), the probability of successful delivery in a given time slot can be estimated for a customer.

Features could include address, day of the week, historical delivery success for that address, local traffic patterns, customer-specific behavior.

Use clustering algorithms to aggregate similar addresses/customer, allowing for the customization of time-slot predictions.

Optimization Algorithm

Use a Vehicle Routing Problem with Time Windows formulation: each order has a chosen time window, vehicles have capacity, and travel times depend on distance/traffic.

Because customers may choose windows dynamically, use an online/dynamic routing algorithm: as new orders come in re-optimize the route - or at least do a partial re-optimization.

Take into consideration heuristics/metaheuristics such as tabu search (as applied in the literature for time-window VRP).

Alternatively, one can integrate a decomposition method (e.g., Benders decomposition) to split the time-window assignment and routing, as in Celik et al.

System Design & Architecture

Design backend services for order handling, prediction, routing, and dispatch via APIs.

Design a database schema: orders, customers, delivery agents, time-slot preferences, route plans.

Customer time-slot selection, driver tracking, and admin dashboard frontend (web/mobile).

Simulation / Testing

Implement a simulator with artificial orders and agents, and a map (or use real map APIs)

Run what-if scenarios: varying load, number of customers, traffic conditions, slot popularity.

Evaluation metrics: On-time delivery rate, number of failed deliveries, extra cost (distance / time), customer satisfaction - maybe proxy via simulation, and computational performance: How long routing takes.

Validation & Real-World Pilot (if possible): If possible, implement as a pilot in a small area with

actual customers / drivers. Collect feedback: how often do customers pick preferred slots, do delivery agents find route changes practical, system stable, etc.

IV. ARCHITECTURAL DESIGN

High-Level Architecture Components

Frontend Layer

Customer App/Web Portal: Order placing, time slot selection, tracking of deliveries, view notifications.

Delivery Agent App: View assigned orders, navigate routes, mark delivery status, and interact with dispatch.

Admin Dashboard: to manage customers, time slots, deliveries, reporting, and system health.

Backend Layer

Order Management Service: Receives order requests and stores order data.

Time-Slot Prediction Service: Runs the ML model to score and recommend/validate slot choices.

Routing/Optimization Engine: Solves the routing problem for the current set of orders, selected slots, and agents; recomputed as necessary.

Dispatch / Scheduler Module: assigns orders to drivers; initiates re-routing upon the arrival of new orders or changes in conditions.

SMS/Push Notification: sends Slot Confirmation and Reminders to customers, route updates to drivers, and notifications to admin.

Real-Time Monitoring Module: driver location and progress, possibly with GPS; data collection in real time for dynamic updates.

Database / Storage

Relational Database: Tables for users/customers, drivers, orders, time slots, historical delivery data.

Time-Series / Telemetry Storage: For storing live location data, traffic metrics, historical times, etc.

Model Storage: Storage for trained ML models, logs, predictions, and versioning.

Integration / External Systems

Map / Routing API: Integrations with mapping services such as Google Maps, OpenStreetMap, etc. for distance, travel time, real-time traffic.

Third-Party Messaging Services: SMS or email gateway for notifications.

Analytics/Reporting: Data pipeline to convert logs into dashboards, KPI reports.

Simplified Data Flow:

Customer places order → selects desired time slot.

Order management system records the choice → sends request to prediction service (optional) to validate the feasibility/suggest better slot.

Once the slot is confirmed, the dispatch module adds the order to the optimization queue.

Routing engine re-calculates optimum route for drivers, or updates an existing route.

Driver is notified; app gets a new route.

During execution, driver location and status are tracked → real-time monitoring updates system.

After delivery, the status is updated; data is logged for model retraining/future prediction.

V. DESIGN MODELS & COMPONENTS-LEVEL DESIGNS

A. Use-Case Diagram

The Use-Case Diagram shows the interaction of different actors with the Smart Delivery System.

Customer can perform various actions: Register/Login, Browse Products, Place Order, Select Time Slot, Track Delivery, and Receive Notifications.

Delivery Agent performs the following actions: Login, View Assigned Orders, Accept/Reject Delivery, Start Route, Update Delivery Status, and Mark Delivery Complete.

Admin/Dispatcher manages time slots, monitors orders, re-optimizes routes, handles exceptions, and views reports.

B. Class Diagram

The class diagram of the Smart Delivery System with Customer Time Selection presents main system entities and interactions between them. Five key classes are: Customer, Order, Time Slot, Delivery Agent, and Route Plan. Customer: The customer details are stored and the customer can place orders and choose time slots for delivery Order: This represents customer orders and maintains all the information about the order like items, time slot, and status. Each order can be associated with one customer, one time slot, and one delivery agent.

Time Slot: Manages available delivery timings. It tracks slot capacity and reservations to prevent overbooking.

Delivery Agent: Deliverymen in charge of accepting assignments, updating the delivery status, and finishing deliveries.
Route Plan: Optimized delivery routes for agents are stored here, including a sequence of delivery stops.
Relationships:
A customer can have multiple orders.
Each order takes up one time slot and is assigned to only one delivery agent.
One route plan contains several orders and is followed by one delivery agent**.** Overall, the class diagram provides an overview of the system's static structure, supporting smart delivery and the selection of time slots, as well as showing how data and operations are organized**.**

C. Sequence Diagram

The Sequence Diagram depicts the flow of messages between system components in the process of ordering up to delivery.
Customer places an order → System saves the order.
Customer selects a delivery time slot → TimeSlot module checks availability → Slot is reserved.
Admin/dispatcher assigns the order to a Delivery Agent.System generates optimized route plan.
Delivery Agent updates delivery status: Out for Delivery → Delivered.
Customer receives notifications in real time.

Frontend Components

Customer UI Module: React/Flutter UI for slot selection tracking.
Driver UI Module: Interface to view route, map integration, update status.
Admin UI Module: Dashboard for slot management, analytics.

Backend Components

API Layer: REST / Graph QL endpoints for order creation, slot selection, and status updates.
Business Logic Layer:
Prediction Module (ML)
Routing Module (optimizer)
Scheduler/Dispatch Module

- Database Layer: Schema design, connection pool, data access layer.
- Notification Module: Abstraction to send SMS / push / email.
- Monitoring Module: Real-time data ingestion, driver tracking, logs.

DevOps / Infrastructure

Containerization (e.g., Docker) for microservices
Message Broker (e.g., Kafka / RabbitMQ) for asynchronous tasks; for example, "new order arrived → trigger routing"
Logging and Monitoring: ELK stack, Prometheus + Grafana Model Serving Framework (for example, TensorFlow Serving / Torch Serve / ONNX) for prediction

VI. Results & Discussion

Key Metrics to Report

On-Time Delivery Rate

The percentage of deliveries made within the selected time slot.
Compare to a baseline system without customer time selection.

Failed Delivery Rate

How many deliveries failed - customer not available vs baseline.

Operational Cost

Total distance traveled, fuel usage (if data available), driver idle/waiting time.
Cost per delivery (or per order) with and without time-slot system.

Customer Satisfaction Proxy

Survey data (if pilot) or simulation metric: e.g., predicted “satisfaction score” based on delays.

Computational Performance

Time taken to optimize a route for each batch of orders.
Scalability: how does system perform as number of orders / drivers increases.
Expected / Hypothetical Results (Based on Literature + Simulation)
By giving customers a choice of time slots, the number of failed deliveries could fall significantly, cutting the cost of repeated attempts. This is reflected in the literature, where it has been found that offering flexible windows can reduce failure costs.
Prediction of optimal slots using ML can further improve the rates of successful delivery: by suggesting time windows that have higher probabilities, you minimize "risky" slots.

Dynamic routing could therefore yield better resource utilization, at the cost of computational overheads from constantly re-optimizing routes as new orders are received.

Discussion Points

Trade-off analysis: there is a trade-off between customer flexibility (narrower slots, more choice) and cost (routing complexity, waiting times). Some literature refers to the fact that narrow windows increase costs.

Customer behavior: Not every customer will select the "optimal" slot that the system predicts. There will be variation, and your model needs to account for that.

Scalability and real-world deployment: In a real system, you'll face unpredictable things: traffic, driver delays, cancellations, no-shows. You should discuss how robust your system is. Sustainability & Environmental Impact: Carbon emissions and energy use could be significantly reduced by routing more efficiently and reducing the number of re-deliveries. Limitations: If your simulation uses synthetic data, you may fail to depict real customer behavior. Also, the ML models may need retraining over time.

VII. CONCLUSION

The Smart Delivery System with Customer Time Selection offers significant enhancement compared to the conventional delivery mechanism through the deep integration of customer preferences, represented by time-slot selection, with advanced predictive and routing algorithms.

Based on the simulation-or pilot, if done-the system can improve on-time delivery rates, reduce failed delivery attempts, and lower the overall cost per delivery. This leads not only to a boost in customer satisfaction but also operational efficiency.

The design is modular: predictive module, routing module, front-end apps, and monitoring can all be improved or replaced independently; this makes the system maintainable and scalable.

Fewer failed deliveries and optimized routes also lead to lower emissions, thus more sustainable logistics from an environmental point of view.

Future Work / Extensions:

AI Enhancement: Employ reinforcement learning to adaptively learn customer behaviors and better predict time slots.

Real-time Traffic Integration: Integrate real-time traffic information through APIs that will update routes on the fly depending on congestion.

Pricing Strategy: Introduce dynamic pricing for time slots: for example, charging more for very narrow or high-demand slots. This has been suggested in the literature.

Stochastic Optimization: Develop stochastic models-such as two-stage stochastic programming-to handle uncertainty in travel times or customer demand, similar to the TWATSP with stochastic travel times.

arXiv Pilot Deployment: Conduct a real-world pilot in a small geographic area; capture real customer data, and refine the models from actual behaviour. Sustainability Metrics: Integrate carbon-emission metrics to quantify environmental benefits and eventually optimize routes not only by cost but also by emissions.

REFERENCES

[1] Cwioro, G., Hungerländer, P., Maier, K., Pöcher, J., & Truden, C. (2018). *An Optimization Approach to the Ordering Phase of an Attended Home Delivery Service*. arXiv. arXiv

[2] Köhler, M., et al. "Service time window selection for attended home deliveries: a case study for urban and rural areas." *Central European Journal of Operations Research* (2023). SpringerLink

[3] Guo, X., Li, Y., Liu, X., & Xu, X. (2023). *Research on Vehicle Routing Optimization for M Company Considering Time Window Constraints*. Frontiers in Traffic and Transportation Engineering. Clausius Press

[4] Wang, Y., et al. *Research on distribution route with time window and on-board constraint based on tabu search algorithm*. EURASIP Journal. SpringerOpen

[5] MDPI. *Vehicle Routing Optimization of Instant Distribution Routing Based on Customer Satisfaction*. MDPI

[6] Gowda, K. J. M., & Guntimadugu, A. (2025). *Artificial Intelligence based Customized Time Slot Delivery of Articles and Parcels*. PhilArchive / International Journal of Innovative Research. PhilArchive+1

[7] IJSREM. *AI Based Customized Time Slot Delivery Of Parcels*. IJSREM

[8] European Journal of Logistics. *Dynamic Route Optimization in Last-Mile Delivery Using Predictive Analytics*. EA Journals

[9] Deliverea. *Time Slot Prediction for Logistics Optimization and Sustainability*. Deliverea

[10] Scisimple. *Managing Delivery Time Slots for Retailers – Logic-Based Benders Decomposition Method*. Simple Science

[11] Celik, S., Martin, L. H., Schrotenboer, A. H., & Van Woensel, T. (2023). *Exact Two-Step Benders Decomposition for the Time Window Assignment Traveling Salesperson Problem*. arXiv. arXiv

