

24 Hour Driver Tracking System

S.kaviyarasu¹, Mr.P.Vallarasu²,

Department of Master of computer Application, AVS Engineering college , Salem, Tamil Nadu, India

Department of Artificial intelligent and Data Science,AVS Engineering college , Salem, Tamil Nadu, India

Abstract—The rapid growth of urban transportation demands intelligent, automated systems to manage cab services efficiently. This paper presents the design, development, and evaluation of a 24 Hr driver Tracking System, a web-based platform engineered to automate taxi service operations, provide GPS-based real-time ride tracking, enable dynamic driver allocation, and facilitate seamless passenger management. The proposed system addresses critical challenges inherent in manual booking workflows, including absence of ride visibility, inefficient driver dispatch, and delayed notifications. Built using a Java-PHP hybrid backend, MySQL relational database, and Bootstrap- JQuery frontend, the platform integrates AJAX and JSON technologies to deliver asynchronous, real-time updates. Eight primary functional modules have been implemented: User Registration & Login, Ride Booking, Driver Allocation, Real-Time Tracking, Payment Processing, Rating & Feedback, Ride History, and an Admin Dashboard. Experimental evaluation demonstrates significant improvements in booking response time, driver allocation efficiency, and overall user satisfaction when compared to conventional manual dispatch systems. The system provides a scalable, extensible architecture that can accommodate increased user loads and additional service features, establishing a robust foundation for next-generation urban mobility platforms.

Index Terms—cab management system, real-time tracking, GPS, driver allocation, web-based application, Java, MySQL, AJAX, ride booking, urban mobility.

I. INTRODUCTION

The transportation sector has undergone a paradigm shift with the proliferation of digital platforms and mobile connectivity. Traditional taxi services relying on manual dispatch, telephone bookings, and paper-based record keeping are increasingly inadequate for meeting modern urban mobility demands. Passengers expect instant booking confirmation, live ride tracking, transparent fare computation, and multiple payment options, while service operators require data-driven insights, efficient driver dispatch, and centralized management dashboards [1].

The emergence of ride-hailing giants such as Uber, Lyft, and regional competitors has demonstrated the immense potential of technology-enabled cab

D. Real-Time Tracking Module

The tracking module captures driver GPS coordinates at 5-second intervals using the browser Geolocation API and transmits them to the server via WebSocket. Passengers and administrators view live ride progress on an embedded map interface. Historical trajectory data is persisted for audit and dispute resolution purposes.

E. Payment Module

Fare calculation integrates base rate, per-kilometer charge, wait-time surcharge, and applicable taxes. The module supports multiple payment modes including credit/debit card processing via a payment gateway API, UPI transfers, and cash confirmation. PDF invoices are generated automatically upon ride completion and emailed to passengers.

services. However, many small and medium-sized taxi enterprises lack the technical infrastructure to deploy comparable solutions. This creates an opportunity for developing accessible, cost-effective, web-based platforms tailored for independent and fleet-based taxi operators.

This paper proposes and evaluates a Cab Driver & Ride Tracking System that consolidates all core taxi service operations within a single, browser-accessible platform. The system leverages a Java-PHP backend for robust business logic, MySQL for persistent data management, and modern frontend technologies including Bootstrap and jQuery for a responsive user interface. GPS integration provides live tracking capabilities, while AJAX-driven communications ensure asynchronous, real-time data exchange without page reloading.

The objectives of the proposed system are: (1) to eliminate manual booking inefficiencies through automated ride request workflows; (2) to provide passengers and administrators with real-time ride visibility; (3) to optimize driver allocation through proximity and availability-based matching; (4) to support cashless payment processing with automated invoice generation; and (5) to deliver actionable analytics through a comprehensive admin dashboard.

The remainder of this paper is organized as follows: Section II reviews related work; Section III describes the system architecture; Section IV details the functional modules; Section V presents the technology stack; Section VI discusses implementation and results; Section VII outlines future enhancements; and Section VIII concludes the paper.

II. RELATED WORK

Considerable research has been devoted to ride-hailing systems, GPS-based tracking, and intelligent driver-passenger matching algorithms. Agatz et al. [2] presented optimization models for ride-sharing and dynamic vehicle routing, establishing foundational algorithms for driver allocation. Their work highlighted the NP-hard complexity of optimal matching and proposed

F. Rating & Feedback Module

Following each completed ride, both passengers and drivers are prompted to submit star ratings (1 to 5) and optional text reviews. Ratings are aggregated using an exponentially weighted moving average to reflect recent performance trends. Accounts falling below threshold ratings are flagged for administrator review.

G. Ride History Module

Comprehensive ride records are maintained for all users, including timestamps, route polylines, fare breakdowns, driver details, and payment receipts. Passengers and drivers can filter history by date range, status, and payment mode. Administrators can export aggregated reports in CSV format for financial reconciliation.

H. Admin Dashboard Module

The centralized dashboard presents real-time KPIs including active rides, driver availability heat maps, revenue summaries, and complaint queues. Role-based access controls restrict sensitive administrative functions to authorized personnel. Automated alerting notifies administrators of anomalies such as abnormally long rides or repeated complaint filings.

V. TECHNOLOGY STACK

Table I summarizes the technology stack employed in the proposed system.

TABLE I
SYSTEM TECHNOLOGY STACK

Component	Technology
Frontend	HTML5, CSS3, Bootstrap 5
Interactivity	jQuery, AJAX, JSON
Backend Logic	Java (Servlets, JSP)
Server-side Script	PHP 8.x
Database	MySQL 8.0
Real-time Comms	WebSocket, SSE
Map Integration	OpenStreetMap / Leaflet.js

heuristic approaches applicable to real-time systems.

Qian et al. [3] developed a taxi GPS trajectory analysis framework that leveraged historical ride data to predict driver availability hotspots and passenger demand zones. Their findings informed demand-responsive allocation strategies that have been incorporated into modern ride-hailing platforms.

With respect to web-based transportation management, Mourad et al. [4] conducted a survey of shared mobility information systems, identifying key requirements including real-time data synchronization, scalable backend architectures, and intuitive passenger interfaces. Their taxonomy aligns closely with the modular design adopted in the proposed system.

Payment gateway integration and secure transaction management in ride-hailing contexts were studied by Kaur & Singh [5], who evaluated multiple payment processing APIs and proposed a security framework for cashless fare collection. Their recommendations for tokenization and SSL-based communication have influenced the payment module design in this work.

Prior work on driver rating and feedback systems by Chen et al. [6] demonstrated that two-sided rating mechanisms (passenger-rates-driver and driver-rates-passenger) significantly improve service quality metrics and reduce incident rates. This bidirectional feedback model is adopted in the Rating & Feedback module of the proposed system.

While existing literature addresses individual components of ride management, few studies present an integrated, full-stack implementation combining real-time tracking, automated allocation, payment processing, and administrative monitoring within a single web-based platform. The current work aims to bridge this gap.

III. SYSTEM ARCHITECTURE

A. Architectural Overview

The proposed system follows a three-tier client-server architecture comprising: (1) a Presentation

Build Tool	Apache Maven
Web Server	Apache Tomcat 10

The Java servlet layer handles complex business logic including allocation algorithms, fare computation, and session management, benefiting from Java's strong type system and mature concurrency support. PHP supplements the backend for lightweight data-handling scripts and API endpoints. MySQL provides ACID-compliant transaction support critical for payment processing integrity.

VI. IMPLEMENTATION & RESULTS

A. Development Environment

The system was developed on Windows 11 workstations using Apache NetBeans IDE 18. Version control was managed through Git with a GitHub remote repository. The application was deployed on Apache Tomcat 10 with MySQL Server 8.0 for local testing and later on a Linux-based virtual server for integration testing.

B. Performance Evaluation

Functional and performance testing was conducted with a simulated dataset of 500 user accounts, 120 driver profiles, and 2,000 historical ride records. Table II presents key performance metrics recorded under controlled test conditions.

TABLE II
PERFORMANCE METRICS

Metric	Manual	Proposed
Avg. Booking Time	4.2 min	28 sec
Driver Allocation Time	6.8 min	45 sec
Tracking Latency	N/A	< 5 sec
Invoice Generation	Manual	Automated
Concurrent Users (tested)	—	250+
DB Query Avg. Response	—	< 120 ms

Layer implemented with HTML5, CSS3, Bootstrap, and jQuery; (2) a Business Logic Layer built on Java servlets and PHP scripts; and (3) a Data Layer consisting of a MySQL relational database. This separation of concerns enhances maintainability, testability, and scalability.

B. Client-Server Communication

Asynchronous communication between the frontend and backend is achieved through AJAX calls that transmit and receive JSON-formatted payloads. This approach minimizes bandwidth consumption and eliminates full-page reloads, delivering a near-native application experience within the browser. WebSocket connections are employed for the real-time tracking module to support bidirectional, low-latency location updates.

C. Database Design

The MySQL schema comprises twelve normalized tables including Users, Drivers, Rides, Payments, Ratings, Locations, Vehicles, Notifications, Audit Logs, Zones, Fares, and Sessions. Foreign key constraints enforce referential integrity, while indexed columns on frequently queried fields ensure optimal query performance. Connection pooling is configured at the Java application layer to manage concurrent database connections efficiently.

IV. SYSTEM MODULES

A. User Registration & Login Module

The registration module supports role-based account creation for passengers and drivers. Input validation is enforced client-side via jQuery and server-side via Java servlets. Passwords are hashed using BCrypt prior to storage. Session tokens are generated upon successful authentication and validated on all subsequent requests.

B. Ride Booking Module

Passengers initiate ride requests by specifying pickup coordinates (via map interface or text input), destination addresses, and preferred ride type. The module computes estimated fares based on configurable zone-rate matrices and distance calculations using the Haversine formula.

Results indicate a 93% reduction in average booking time and an 89% reduction in driver allocation time compared to a manual dispatch baseline. Real-time tracking latency was maintained below 5 seconds under normal network conditions. The system sustained 250 concurrent simulated user sessions without observable degradation in response time, validating the scalability of the connection pool and database indexing configuration.

C. User Acceptance Testing

User Acceptance Testing (UAT) was conducted with 40 volunteer participants (25 simulated passengers and 15 simulated drivers) over a two-week period. Participants completed structured scenarios including ride booking, live tracking observation, payment simulation, and feedback submission. Post-test surveys using a 5-point Likert scale yielded mean satisfaction scores of 4.6 for interface usability, 4.4 for tracking accuracy, and 4.7 for booking convenience.

VII. FUTURE ENHANCEMENTS

Several directions are identified for future development of the system:

- **Machine Learning-based Demand Forecasting:** Integration of time-series prediction models to anticipate demand surges and pre-position drivers, reducing wait times during peak periods.
- **Mobile Native Applications:** Development of Android and iOS companion apps to extend accessibility beyond browser-based interactions, supporting offline booking queuing and push notifications.
- **Dynamic Surge Pricing:** Implementation of algorithmic fare adjustment based on real-time supply-demand ratios, consistent with industry standards for revenue optimization.
- **Route Optimization:** Incorporation of shortest-path and traffic-aware routing algorithms to minimize trip duration and fuel consumption for shared rides.
- **Multi-language Support:** Localization of the interface to support regional languages, broadening accessibility across India's linguistically diverse user base.

Scheduled rides support advance booking up to 72 hours ahead.

C. Driver Allocation Module

Upon ride request submission, the allocation engine queries all available drivers within a configurable radius, sorted by proximity and weighted by performance rating. Assignment notifications are pushed to the selected driver via server-sent events. If the driver does not accept within 30 seconds, the next-ranked candidate is notified automatically.

- Blockchain-based Payment Ledger: Exploration of distributed ledger technology for tamper-proof transaction recording and automated dispute resolution.

VIII. CONCLUSION

This paper presented the comprehensive design and implementation of a Cab Driver & Ride Tracking System, a fully integrated web-based platform addressing the operational limitations of traditional taxi service management. The system's eight modular components collectively deliver automated ride booking, proximity-based driver allocation, GPS-enabled real-time tracking, secure payment processing, bidirectional rating mechanisms, and centralized administrative monitoring.

Performance evaluation confirmed substantial efficiency gains over manual dispatch workflows, including a 93% reduction in booking time and an 89% reduction in driver allocation latency. User acceptance testing validated high levels of satisfaction across usability, tracking accuracy, and booking convenience dimensions. The Java-PHP hybrid backend architecture, supported by a normalized MySQL schema and AJAX-driven frontend, provides a scalable and maintainable foundation for ongoing platform evolution.

The proposed system contributes a practical, deployable reference architecture for small and medium taxi enterprises seeking to modernize operations without the capital investment required by proprietary ride-hailing platforms. Future work will focus on machine learning-driven demand forecasting, mobile application development, and dynamic pricing mechanisms.

REFERENCES

- [1] S. Shaheen and A. Cohen, "Shared Mobility: Current Practices and Guiding Principles," Federal Highway Administration, Washington, DC, USA, Tech. Rep. FHWA-HOP-16-022, 2016.
- [2] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295-303, 2012.
- [3] X. Qian, S. Ukkusuri, C. Yang, and F. Yan, "Short-term demand forecasting for on-demand mobility service," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4490-4499, 2019.
- [4] A. Mourad, J. Puchinger, and C. Chu, "A survey of models and algorithms for optimizing shared mobility," *Transportation Research Part B: Methodological*, vol. 123, pp. 323-346, 2019.
- [5] H. Kaur and R. Singh, "Secure payment processing framework for mobile ride-hailing applications," in *Proc. IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, 2020, pp. 1-6.
- [6] M. Chen, T. Sheldon, and Y. Xu, "Reputation systems for ride-hailing: Theory and evidence," *Management Science*, vol. 67, no. 11, pp. 6777-6799, 2021.
- [7] A. Furuhata, M. Dessouky, F. Ordonez, M. E. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transportation Research Part B: Methodological*, vol. 57, pp. 28-46, 2013.
- [8] N. Masoud and R. Jayakrishnan, "A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system," *Transportation Research Part B: Methodological*, vol. 106, pp. 218-236, 2017.