WavyRider: An Interoperable Mobility Platform for Real-Time Fare Comparison

SHRUTHI BL[1], ADITYA KUMAR[2], SHROJEET BAIDYA[3], ABHINAV RAWAT[4], JEFRIN JOBITH[5]
[1]*Assistant professor, Department of Computer Science Engineering, Rajiv Gandhi institute of technology Bangalore, India*
[2,3,4,5] *Dept. of Computer Science, Rajiv Gandhi institute of technology, Bangalore*

***Abstract*—This paper presents WavyRider, an interoperable web-based mobility platform designed to provide real-time fare comparison across multiple transportation providers, including Ola, Uber, and Rapido. The system integrates cab, auto, and bike services into a unified platform, enabling users to identify and book the most affordable rides between chosen pickup and drop locations. The proposed solution addresses the inefficiency of switching between multiple ride-hailing applications by offering transparent pricing, seamless booking, and enhanced commuter convenience. Through literature review, modular design, and API integration, this paper explores the economic, technical, and operational implications of implementing such a platform.**

***Keywords*—Mobility as a Service (MaaS), ride-hailing, fare comparison, interoperability, WavyRider.**

I. INTRODUCTION

Mobility as a Service (MaaS) is an emerging paradigm that integrates various transportation modes—such as bikes, cabs, and autos—into a unified platform for planning, booking, and payment [1]. This model aims to reduce car dependency, enhance transport efficiency, and promote sustainable mobility. Existing MaaS models include the System Integrator Model, Platform Model, and Intermediary Model, each differing in pricing control and market outcomes.

WavyRider is a web-based platform integrating Ola, Uber, and Rapido ride services. Users can compare fares for cabs, autos, and bikes, then directly book their preferred ride. By eliminating the need to switch between apps, the system enhances convenience and transparency. Additionally, *WavyRider* is designed for broad accessibility, including budget-conscious and mobility-challenged users.

II. LITERATURE SURVEY

Smith and Johnson [1] analyzed pricing algorithms of Ola, Uber, and Lyft, finding significant fare variations that support the need for real-time comparison tools. Kumar and Sharma [2] showed that integrating multiple transport modes improves trip efficiency and user satisfaction, aligning with WavyRider multi-modal approach. Li and Wang [3] developed accurate fare prediction models, which could enhance WavyRider price accuracy.

Commercial rideshare aggregators (2024) offer limited multi-service fare comparisons but are mostly mobile-only, unlike WavyRider's web-based solution. The Urban Mobility Report [10] highlighted that transparent, coordinated ride-sharing can reduce congestion and emissions. Jain and Patel (2024) found that price and convenience are the top priorities for ride-hailing users, reinforcing WavyRider's focus on affordability and ease of booking.

III. EXISTING SYSTEM

Current urban mobility is dominated by independent ride-hailing applications such as Ola, Uber, and Rapido. Each operates within its own ecosystem, requiring users to download the respective mobile application, create an account, and search for rides individually. While these platforms offer features like estimated fares, live driver tracking, and digital payments, they operate in isolation—meaning there is no native mechanism to compare prices or availability across different providers.

1. Lack of Integration – No existing web-based service aggregates data from multiple ride-hailing providers in real time for all ride types (cab, auto, bike).
2. Limited Platform Availability – Most aggregator services, where they exist, are mobile-only, excluding users who prefer desktop or kiosk-based access.
3. Inconsistent Fare Transparency – Fare calculations vary widely between providers, and the absence of real-time comparative tools can lead to overpayment.

4. User Inconvenience – Frequent app switching not only delays decision-making but can also lead to missed ride opportunities during high-demand periods.

IV. PROPOSED METHOD

The proposed WavyRider platform is a web-based mobility aggregator that integrates APIs from Ola, Uber, and Rapido to provide real-time fare comparison for cabs, autos, and bikes. Users can enter pickup and drop-off locations, view ranked ride options based on cost, time, or eco-preference, and proceed with booking directly or via the provider's platform.

The system architecture includes:

1. User Interface Layer – Built with React.js and Tailwind CSS for a responsive, intuitive design.
2. API Integration Layer – Fetches live fare and ETA data from providers.
3. Fare Comparison Engine – Aggregates and sorts results in real time.
4. Booking & Payment Module – Supports secure in-platform booking or redirection, with payment gateways like Razorpay/Stripe.
5. Location Services – Powered by Google Maps API for geolocation and routing.
6. Notification & Analytics Modules – Provide ride updates and performance insights.

This method ensures commuters access accurate, instant, and transparent ride information without switching between multiple apps, improving both convenience and cost efficiency.

V. WORKING PRINCIPLE

WavyRider operates by aggregating and comparing real-time fare data from multiple ride-hailing providers through API integration. The process begins when the user enters pickup and drop-off locations. The system then sends simultaneous API requests to Ola, Uber, and Rapido to fetch live fare estimates, availability, and estimated arrival times.

The Fare Comparison Engine processes this data and ranks ride options based on user preferences such as lowest cost, shortest ETA, or eco-friendly options. The results are displayed in a single interface, allowing direct booking via the platform or redirection to the provider's app.

Payments, if processed in-platform, are handled through secure gateways. Users receive booking confirmations and ride status updates until trip completion. This streamlined workflow ensures fast, transparent, and cost-effective travel decisions without switching between multiple apps.

VI. IMPLEMENTATION AND TECHNOLOGY

The implementation of WavyRider is carried out using a modular architecture, ensuring scalability, maintainability, and efficient integration with multiple ride-hailing providers. The platform is designed as a web-based application to ensure device independence and wider accessibility compared to mobile-only solutions.

A. Implementation Overview

WavyRider is implemented as a modular web application to ensure scalability and real-time performance.

- Frontend – Developed with React.js and Tailwind CSS for a responsive interface, integrated with Google Maps API for location search and routing.
- Backend – Built using Node.js and Express.js to handle API requests, process fare comparisons, and manage secure communication with Ola, Uber, and Rapido APIs.
- Database – MongoDB or Firebase Firestore stores user data, preferences, and booking history.
- Payments – Razorpay or Stripe integrated for secure transactions.
- Authentication – Managed via Firebase Auth and OAuth 2.0.
- Notifications – Firebase Cloud Messaging, Twilio, or SendGrid used for updates.
- Analytics – Google Analytics and Mixpanel track usage trends.

This technology stack ensures fast, secure, and accurate fare aggregation while allowing future expansion to additional transport providers

REFERENCES

[1] A. Smith and R. Johnson, "Comparative Analysis of Ride-Hailing Platforms," *Int. J. Transp. Logistics*, vol. 15, no. 3, pp. 210–225,2021.

[2] P. Kumar and V. Sharma, "Urban Mobility and Multi-Modal Transport Systems," *J. Smart City Planning*, vol. 12, no.4,pp.98–110,2020.

[3] X. Li and J. Wang, "Predictive Models for Real-Time Fare Estimation in Ride-Sharing Apps," *IEEE Trans. Intell. Transp. Syst.*, 2019.
[4] Google Maps API Documentation, [Online].Available: https://developers.google.com/maps/documentation
[5] Uber Developer API, [Online]. Available: https://developer.uber.com
[6] Ola Developer Platform, [Online]. Available: https://dev.olacabs.com
[7] Rapido Developer API, retrieved from open-source communities and third-party aggregators. Available: https://dev.rapido.com