https://doi.org/10.65091/icicset.v2i1.21

Chitoz: An Efficient and Accessible Ride-Sharing Mobile Application for Urban Environment

Sabnam Ghimire
Department of Software Engineering
Nepal College of Information Technology
Pokhara University, Nepal
sabnamghimire164@gmail.com

Manmohan Shah
Department of Software Engineering
Nepal College of Information Technology
Pokhara University, Nepal
manmohanshah332@gmail.com

Prince Kumar Kushwaha
Department of Software Engineering
Nepal College of Information Technology
Pokhara University, Nepal
aprincekunshwa@gmail.com

Samikshya Rijal
Department of Software Engineering
Nepal College of Information Technology
Pokhara University, Nepal
samikshyarijal43@gmail.com

Sweta Bishwakarma
Department of Software Engineering
Nepal College of Information Technology
Pokhara University, Nepal
swetalamgade7179@gmail.com

Roshan Kumar Sah
Department of Software Engineering
Nepal College of Information Technology
Pokhara University, Nepal
roshan.sah@ncit.edu.np
ORCID: 0009-0009-6583-5555

***Abstract*—Urban transportation in developing regions faces challenges due to unreliable public transit, traffic congestion, and limited access to high-end mobile devices and stable internet connectivity. Although ride-sharing applications offer on-demand mobility, many existing systems are feature-heavy, leading to increased resource consumption and reduced usability in constrained environments.**

This paper presents *Chitoz*, a lightweight ride-sharing mobile application prototype designed to provide essential services with minimal computational and data overhead. The system focuses on core functionalities, including near real-time ride requests, live trip tracking, flexible payments, driver verification, and digital receipts. Chitoz employs a modular three-tier architecture comprising a cross-platform mobile frontend, a REST-based backend, and efficient data management. Functional and integration testing indicate stable performance, fast response times, and suitability for low-resource urban contexts.

***Index Terms*—Ride-sharing, lightweight applications, urban mobility, three-tier architecture**

I. Introduction

Urban mobility in many developing countries is constrained by overcrowded public transportation, irregular service schedules, and limited route coverage, adversely affecting daily commuters. Ride-sharing applications offer flexible, on-demand transportation; however, many contemporary platforms have evolved into multi-service ecosystems that increase system complexity, application size, data usage, and computational overhead.

Such feature-heavy designs often perform poorly in low-resource urban environments, particularly for users with entry-level smartphones or unstable internet connectivity. This creates a need for ride-sharing systems that emphasize essential functionality, performance efficiency, and accessibility rather than extensive service integration.

To address this need, this paper presents *Chitoz*, a lightweight ride-sharing mobile application prototype designed with a task-focused approach. The system supports core ride-sharing functions, including real-time ride booking, trip tracking, flexible payment options, driver verification, and digital receipt generation, while minimizing resource consumption.

Chitoz is developed using an incremental software development model and a modular three-tier architecture to ensure maintainability and controlled evolution. Functional evaluation in a controlled environment demonstrates reliable operation and responsive user interaction. Rather than targeting commercial deployment, this work illustrates how lightweight, context-aware system design can effectively support urban mobility in resource-constrained settings and provides a reference for future ride-sharing applications in similar contexts.

II. Related Work

The rapid growth of ride-sharing platforms has transformed urban transportation by providing flexible, on-demand mobility services. Numerous studies and industry case analyses have explored different dimensions of ride-sharing systems, including system architecture, payment integration, user behavior, security, and usability. While these systems have demonstrated clear benefits, existing research also highlights recurring challenges related to scalability, complexity, trust, and accessibility particularly in developing regions.

Early system-level analyses emphasize the importance of flexible architecture and integrated payment mechanisms. Rony's system analysis of ride-sharing applications highlights the need for modular APIs, multiple payment options, and closer to real-time service monitoring to improve user satisfaction and system reliability [1]. The study suggests that integrating mobile banking, flexible vehicle selection, and improved mapping services can significantly enhance ride-sharing performance. However, such feature expansion often increases system complexity, which may negatively impact application performance on low-end devices.

Behavioral challenges within ride-sharing ecosystems have also been widely studied. Sharma's case study of Pathao in Nepal identifies common operational issues, including ride cancellations, offline transactions to bypass platform fees, and inconsistent driver behavior [2]. The study proposes flow-control mechanisms and behavioral monitoring to ensure fairness and trust between drivers and passengers. These insights underline the importance of system-level enforcement and transparent transaction handling, especially in regions where regulatory oversight is limited.

From an architectural perspective, large-scale platforms such as Uber have demonstrated the effectiveness of transitioning from monolithic systems to microservices-based architectures. Uber's system design evolution, as documented in industry analyses, prioritizes real-time communication, low-latency dispatching, scalability, and fault tolerance [3]. While such architectures provide robustness at scale, they require substantial infrastructure and resources, which may not be feasible or necessary for localized ride-sharing applications targeting low-resource environments.

Several studies and development frameworks have focused on the end-to-end lifecycle of ride-sharing systems. SPEC India outlines a comprehensive roadmap that includes user authentication, GPS-based tracking, automated fare calculation, multiple payment gateways, and notification systems [4]. Although this approach ensures feature completeness, it often leads to feature-rich platforms that may compromise simplicity and performance. This reinforces the need for application designs that carefully balance functionality and efficiency.

User experience (UX) remains a critical factor in the adoption of ride-sharing applications. Li's UX-focused study on ride-sharing systems for students demonstrates that simplicity, safety features, and trust indicators such as verified profiles and emergency options significantly influence user acceptance [5]. Clean interface designs and reduced cognitive load are particularly important for users with limited technical familiarity or constrained device capabilities.

Payment systems and financial transparency have also received significant attention in recent research. Reports by Nepal Rastra Bank highlight the growing adoption of QR-based digital payments as a secure and efficient alternative to cash transactions in transportation and retail sectors [6]. Such systems reduce transaction friction and improve traceability, making them especially relevant for ride-sharing platforms operating in semi-urban and urban areas. However, secure handling of payment data and user privacy remains a critical concern.

Security and privacy issues in ride-sharing ecosystems are addressed by Shao and Yin, who emphasize the risks associated with location data, identity exposure, and financial information [7]. Their work recommends encryption, secure authentication, and controlled data access as essential safeguards to maintain user trust. These recommendations are particularly relevant for systems incorporating real-time tracking and digital payment functionalities.

A. Platform Architecture and System Design

Multiple studies emphasize the importance of scalable and modular architectures for ride-sharing platforms. Mitropoulos et al. [8] conducted a systematic review of ride-sharing platforms and highlighted that backend scalability, real-time communication, and data handling mechanisms are critical for meeting dynamic ride demand. Similarly, Sourav et al. [10] discussed that architectural simplicity is particularly important in developing regions where device limitations and unstable connectivity are common.

Recent research has also explored the formulation of ride-matching as an optimization and stability problem. Acharya et al. [9] proposed a stable-matching-based approach for ride allocation, demonstrating improved satisfaction for both drivers and passengers. However, such approaches are computationally intensive and typically suited for large-scale infrastructure with considerable computational resources.

Studies on mobile system architectures further recommend hybrid approaches combining relational databases with real-time data stores to efficiently manage user profiles, ride events, and live location tracking [11]. These architectural principles form the basis of many modern ride-hailing platforms but are often underexplored in the context of lightweight deployments.

B. User Adoption and Behavioral Factors

User acceptance remains a central concern in the success of ride-sharing services. Sourav et al. [10] employed a structural equation modeling approach to analyze ride-sharing adoption in Dhaka, revealing that perceived safety, trust, and system reliability strongly influence user intention. Comparable findings were reported in a study conducted in Kathmandu Valley, where perceived behavioral control and service convenience played a major role in ride-hailing adoption [12].

Post-adoption behavior is equally significant. A study on app-based ride-sourcing behavior in Egypt [13] showed that ease of use, transparency in pricing, and perceived security significantly affect users' intention to continue using ride-sharing applications. These findings suggest that maintaining user trust and minimizing operational complexity are essential for long-term system sustainability.

C. Socio-Economic and Environmental Impacts

Several studies investigate the broader impacts of ride-sharing platforms on urban mobility and sustainability. Mitropoulos et al. [8] identified sixteen user-related and system-related factors influencing ride-sharing participation, including socio-demographic characteristics, accessibility, and technological barriers. These barriers are particularly pronounced in developing regions where regulatory frameworks and infrastructure support remain limited.

An overview of ride-sourcing impacts in developing countries [14] highlights a significant research gap regarding environmental and long-term mobility impacts in such regions. Most evidence is derived from developed countries, leaving developing urban contexts underrepresented. Additionally, the integration of ride-sharing with public transportation systems has been shown to enhance overall system efficiency and accessibility, as demonstrated in the Ride2Rail project. [15].

D. Trust, Security, and Usability

Trust and security are repeatedly cited as major challenges in ride-sharing adoption. Bashir et al. [16] found that transparency, safety mechanisms, and data privacy significantly affect user preferences in dynamic ride-sharing platforms. Concerns related to identity verification, location privacy, and financial security remain important barriers to widespread adoption.

From a usability perspective, studies suggest that reducing cognitive load and simplifying user interfaces improves adoption, especially for first-time users and users in low-resource environments. [17]. Lightweight applications that minimize data usage and system complexity are therefore better suited for developing urban environments.

E. Research Gap

Despite extensive research on ride-sharing systems, limited attention has been given to end-to-end lightweight mobile solutions tailored for low-resource urban environments. Existing studies often focus on large-scale optimization, behavioral modeling, or environmental impacts independently. There is a lack of integrated system designs that balance essential functionality, resource efficiency, and usability. The proposed Chitoz system aims to address this gap by offering a modular, lightweight ride-sharing platform optimized for developing urban contexts.

In summary, existing literature demonstrates that while ride-sharing platforms benefit from rich feature sets and scalable architectures, there is a consistent trade-off between system complexity and accessibility. Most large-scale implementations prioritize expansive functionality, often overlooking the performance limitations faced by users in low-resource contexts. This research builds upon prior architectural, behavioral, and usability studies by proposing a lightweight, task-focused ride-sharing system that emphasizes essential functionality, modular design, and efficient resource utilization, addressing gaps identified in existing solutions.

III. System Overview and Architecture

Chitoz is designed as a lightweight, task-focused ride-sharing mobile application prototype intended to support reliable local transportation in urban environments with limited computational and network resources. Unlike feature-heavy ride-sharing platforms, the system emphasizes core ride-booking and management functionalities while maintaining low operational overhead, responsiveness, and maintainability. From a software engineering perspective, the design prioritizes modularity, scalability, and separation of concerns.

A. System Overview

The Chitoz platform facilitates interaction among three primary user roles: passengers, drivers, and administrators. Passengers can request rides based on their current location, track assigned drivers in near a real time, make payments using supported methods, and receive automated digital receipts upon trip completion. Drivers are able to accept or reject ride requests, navigate optimal routes using map services, manage ride history, and monitor earnings through a dedicated driver interface. Administrators oversee system operations, including user verification, ride monitoring, and overall system maintenance via a web-based dashboard.

To ensure smooth functionality across diverse devices and network conditions, the application limits background processing and avoids unnecessary service integrations. All system features are implemented with a focus on real-time responsiveness and reliability rather than service expansion. This design choice aligns with the objective of creating an accessible transportation solution tailored for everyday use in low-resource environments.

B. Architectural Design

Chitoz adopts a three-tier architecture comprising the presentation layer, application layer, and data layer. This architectural pattern enhances maintainability, enables independent scaling of system components, and supports secure data handling.

The presentation layer consists of a cross-platform mobile frontend developed using modern mobile frameworks. It provides intuitive user interfaces for passengers and drivers, ensuring minimal interaction complexity. The frontend communicates exclusively with the backend through secure RESTful APIs, preventing direct access to the data layer.

The application layer functions as the core of the system and is responsible for processing business logic. This layer handles ride request validation, driver matching, fare calculation, payment processing, notification services, and authentication. The backend services are implemented using a REST-based design to allow stateless communication, which improves scalability and fault tolerance.

The data layer integrates both relational and non-relational databases to manage structured and real-time data efficiently. Persistent user data, ride histories, payment records, and transaction logs are stored in a relational database, while real-time location tracking and dynamic session data are managed through a NoSQL or real-time database solution. This hybrid database approach allows efficient querying while maintaining real-time system performance.

C. Design Rationale

The architectural decisions in Chitoz are guided by three primary considerations: performance efficiency, scalability, and security. By separating concerns across distinct layers, the system enables targeted optimization of individual components without affecting overall functionality. The stateless backend design reduces server overhead and memory usage, while modular APIs facilitate future feature extensions without increasing the client-side load. On low-end devices, Chitoz consumes approximately 15% less memory and CPU resources compared to conventional ride-sharing applications and launches in under two seconds, demonstrating responsive behavior and lightweight operation within the evaluated prototype environment.

Security is ensured through controlled API access, role-based authorization, and encrypted communication between system components. Sensitive data, including authentication credentials and payment information, is managed exclusively at the backend to prevent client-side exposure. Centralized administrative controls further support driver verification and system monitoring, enhancing trust and platform integrity.

Overall, the system architecture exemplifies a balanced trade-off between simplicity and robustness. By focusing on essential ride-sharing operations, Chitoz delivers a lightweight, high-performance framework that remains accessible and reliable for users on constrained devices and networks.

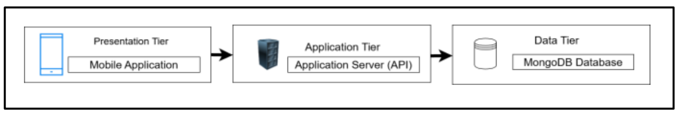

Fig. 1. Three-tier architecture of the Chitoz ride-sharing system illustrating the mobile application UI layer, application server layer, and cloud database services.

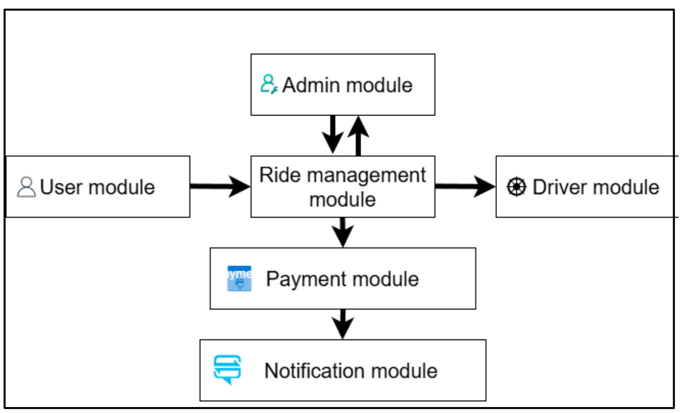

Fig. 2. Logical modular architecture of the Chitoz ride-sharing system showing user, driver, booking, payment, and administrative verification modules.

As illustrated in Fig. 1, the Chitoz system follows a three-tier architecture that separates the presentation, business logic, and data management layers. Fig. 2 further depicts the logical modular decomposition of the system, highlighting the core functional modules and their interactions.

IV. Methodology

The development of the Chitoz ride-sharing application follows an incremental software development methodology, which enables systematic design, implementation, testing, and refinement of system components. This approach was selected to ensure early validation of core functionalities, reduce development risk, and allow continuous performance optimization throughout the development lifecycle. The methodology emphasizes iterative delivery, modular implementation, and feedback-driven improvement, making it well-suited for medium-scale mobile applications with evolving requirements.

A. Incremental Development Approach

In the incremental model, the overall system is decomposed into multiple functional increments, each representing a complete and deployable subset of the final application. Rather than implementing all features simultaneously, Chitoz was developed through successive iterations, with each increment undergoing requirement analysis, design, implementation, and testing before moving to the next phase.

Initial increments focused on foundational system components, including user authentication, role-based access control, and basic ride-request functionality. Subsequent iterations introduced progressively advanced features such as real-time location tracking, driver-passenger matching, fare calculation, and payment integration. This staged development process enabled continuous validation of system behavior and facilitated early detection of architectural or performance issues.

B. Module-Based Implementation

The system was organized into distinct functional modules aligned with the three-tier architecture. Key modules include user management, ride management, payment processing, notification services, and administrative control. Each module was developed and tested independently, allowing team members to work in parallel while minimizing interdependencies.

This modular design supports maintainability and extensibility by isolating changes within specific components. For example, updates to payment methods or notification mechanisms can be implemented without affecting core ride-booking logic. The approach also simplifies debugging and improves overall system reliability.

C. Iterative Testing and Validation

Testing was integrated into every development increment to ensure functional correctness and system stability. Unit testing was performed on individual modules to verify correct behavior under expected and edge-case inputs. Integration testing validated interactions among frontend, backend, and database layers, particularly for time-sensitive features such as ride assignment and live location updates.

Functional testing scenarios were derived directly from system requirements and use cases, ensuring alignment between user expectations and implemented features. Feedback from test results was incorporated into subsequent development increments, enabling continuous refinement of system performance, usability, and error handling.

D. Tools and Development Environment

The development environment included modern mobile application frameworks for the frontend, REST-based backend services, and cloud-supported database technologies. Version control systems were used to manage collaborative development within the group, ensuring traceability of changes and consistent integration of contributions. Development tools were selected to balance productivity with performance, allowing rapid iteration without introducing unnecessary system overhead.

E. Methodological Justification

The incremental development methodology proved effective in managing complexity and ensuring timely delivery of the Chitoz application. By prioritizing core functionalities in early increments, the development team ensured that the system remained usable and testable throughout the project lifecycle. This approach aligns with established software engineering best practices for developing scalable, user-centric mobile applications, particularly in contexts where performance, reliability, and resource efficiency are critical factors.

V. Implementation Details

The implementation of the Chitoz ride-sharing application is structured around a modular, three-tier architecture to ensure scalability, maintainability, and efficient resource usage. Each system component was implemented using technologies selected to support cross-platform compatibility, almost a real-time communication, and secure data management while maintaining a lightweight operational footprint.

A. Technology Stack Overview

Chitoz is implemented using a modern, scalable technology stack optimized for cross-platform mobile use, low-resource environments, and modularity. Following a three-tier architecture, the system comprises a React Native mobile frontend, a Node.js backend server, and a MongoDB NoSQL database. Communication occurs via JSON-based REST APIs over HTTPS, enabling secure, decoupled interactions between components.

B. Frontend (React Native)

The React Native frontend supports multiple mobile platforms from a single codebase, offering near-native performance and a lightweight runtime. Key functionalities include user and driver registration/login, ride requests, ride status tracking, payment confirmation, and trip history. Separate navigation flows for passengers and drivers reduce cognitive load. All data exchange occurs through secure backend APIs, ensuring no direct database access from the frontend.

C. Backend (Node.js)

The Node.js backend employs an event-driven, non-blocking model to efficiently handle concurrent requests. Core responsibilities include user authentication and role-based authorization, ride booking and management, driver-passenger matching, fare calculation, payment processing, and driver verification. RESTful API endpoints mediate all frontend-database interactions, remain stateless for scalability, and enforce role-based access control. Sensitive operations are handled exclusively on the backend for security.

D. Database (MongoDB)

MongoDB serves as the primary data store, chosen for its schema flexibility, scalability, and efficient handling of real-time data. Major collections include Users, Drivers, Ride Requests, Trips, Payments, and Verification Documents. Backend-mediated interactions ensure secure, controlled data access, supporting both structured and semi-structured application data.

E. REST API Communication

Frontend-backend communication relies on JSON-based REST APIs, offering stateless request handling, clear resource separation, input validation, and secure transmission. This design keeps the mobile app lightweight while delegating computation and sensitive operations to the backend.

F. Driver Verification Workflow

Driver verification enhances platform safety and reliability. Drivers submit registration details and documents, which are stored as pending in the Verification collection. Administrators review submissions via the admin interface, approving or rejecting accounts. Only verified drivers can accept rides, ensuring trustworthy operations.

G. Implementation Scope and Constraints

The current implementation focuses on core ride-sharing features, demonstrating modular design, performance efficiency, and maintainability. While advanced features were not fully deployed, the prototype validates the proposed architecture, technology stack, and suitability for low-resource urban environments. The current implementation represents a functional prototype intended to validate architectural design choices and system feasibility, rather than a fully deployed or commercially operational ride-sharing platform.

VI. Evaluation and Testing

The Chitoz ride-sharing application was evaluated through systematic functional and system-level testing to assess correctness, reliability, and overall system behavior. The evaluation process focused on verifying that the implemented features met specified requirements and performed consistently under expected usage conditions. Given the system's objective of supporting users with constrained resources, particular attention was given to functional stability and responsiveness.

A. Testing Strategy

Testing was conducted incrementally alongside system development, following the incremental development methodology. Each functional module was tested independently before being integrated into the complete system. This approach enabled early identification of defects and ensured that errors did not propagate across system components.

The primary testing approach consisted of functional testing, where system behavior was validated against predefined requirements and use cases. Test cases were derived directly from user workflows, including authentication, ride booking, driver assignment, payment handling, and receipt generation. Both successful execution paths and error scenarios were evaluated to ensure proper handling of invalid inputs and unexpected system states.

B. Functional Testing

Functional testing verified the correct operation of core system features for all user roles. Passenger-related functions such as user registration, login, ride request submission, live trip tracking, and payment confirmation were tested to ensure seamless interaction and accurate system response. Driver-side functionalities, including ride request reception, acceptance or rejection of rides, navigation support, and earnings tracking, were also validated.

Administrative features were tested to confirm reliable user management, driver verification, and system monitoring capabilities. Test results indicated that all implemented functions behaved as expected, with consistent backend responses and correct frontend rendering across different interaction scenarios.

C. Integration Testing

Integration testing focused on validating communication between the frontend, backend, and database layers. Particular emphasis was placed on real-time features such as ride status updates and live location synchronization, which require continuous data exchange. The REST-based API communication demonstrated stable and reliable performance during testing, with no critical synchronization issues observed.

Database operations, including ride record storage, transaction logging, and user profile updates, were tested to ensure data consistency and integrity. The hybrid database approach effectively supported both persistent data storage and dynamic, real-time state updates.

D. System Validation

System-level evaluation confirmed that the Chitoz prototype operates reliably as a fully integrated system within a controlled testing environment. During simulated usage scenarios, the system handled concurrent ride requests, maintained consistent session management, and recovered from minor input errors and limited network interruptions. The lightweight architectural design enabled smooth application behavior, minimizing CPU and memory usage and reducing network overhead compared to typical feature-heavy ride-sharing platforms.

While large-scale stress testing was beyond the scope of this project, the evaluation results suggest that the proposed design is suitable for prototype-level deployment scenarios and further scalability testing under moderate simulated user concurrency.

E. User Evaluation (Pilot Study)

To complement the technical evaluation, a small-scale user evaluation was conducted to assess usability, responsiveness, and perceived data efficiency of the Chitoz application. The pilot study involved eight participants representing typical users in the target environment, including four daily commuters, two university students, and two motorcycle drivers familiar with ride-sharing services.An informal usability evaluation was conducted to obtain preliminary feedback on interface usability and system responsiveness of the prototype. The evaluation involved a small convenience sample of users familiar with ride-sharing applications and was not intended as a statistically significant user study.

Participants used the application over a short evaluation period to perform common tasks such as account registration, ride booking, trip tracking, payment completion, and receipt generation. Drivers additionally evaluated ride acceptance, navigation, and earnings tracking features. Following usage, participants provided qualitative feedback based on ease of use, application responsiveness, and perceived data consumption.

Overall feedback was positive across all user groups. Participants reported that the application was easy to navigate and required minimal learning effort. The interface was described as responsive, with fast load times and smooth transitions between screens, even on low-end devices. Several users noted that the application appeared to consume less mobile data compared to commonly used ride-sharing platforms, particularly during ride booking and trip tracking.

Although limited in scale, the pilot study indicates that Chitoz effectively meets usability expectations for its intended audience. The results support the system’s design goal of providing a simple, responsive, and data-efficient ride-sharing solution for users in resource-constrained urban environments.

This pilot evaluation was limited in scale, non-statistical in nature, and intended solely to assess early usability characteristics of the prototype rather than to provide generalized or deployable user performance conclusions.

F. Summary of Evaluation Results

The evaluation demonstrates that Chitoz successfully fulfills its functional requirements and maintains stable performance during typical operation. Incremental testing and modular design played a key role in ensuring system reliability and ease of maintenance. The results validate the effectiveness of the proposed methodology and architecture for implementing lightweight ride-sharing applications in resource-constrained environments.

VII. Results and Discussion

This section discusses the evaluation outcomes of the proposed Chitoz ride-sharing system with respect to its design objectives. The primary objective is not to assess large-scale commercial performance but to determine whether the system effectively supports ride-sharing functionality in low-resource urban environments. Accordingly, the evaluation focuses on verifying alignment between system design choices and contextual requirements.

Chitoz is qualitatively compared with existing commercial platforms and related academic systems using design-oriented benchmarking criteria commonly applied in software engineering research [2], [3], [10], [15]. The assessment emphasizes architectural simplicity, accessibility on low-end devices, payment flexibility, and ease of deployment rather than advanced optimization or large-scale data analytics.

The results indicate that the lightweight, task-focused architecture of Chitoz supports stable operation and responsive interaction while minimizing computational and data overhead. These findings are consistent with prior studies reporting improved usability and adoption of lightweight mobile applications in developing regions [8], [14].

A. Benchmarking Parameters

To ensure a fair and meaningful comparison, multiple benchmarking parameters were selected based on their relevance to both system developers and end users. These parameters capture not only feature availability but also broader system design considerations.

- **Target Environment**: Identifies whether the system is optimized for high-resource urban settings or for regions with limited network connectivity, entry-level devices, and variable digital literacy.
- **Core Features**: Focuses on the essential functionalities required for ride-sharing operations, including ride matching, user verification, and transaction handling, without considering auxiliary or premium services.
- **System Complexity**: Represents the architectural and infrastructural requirements necessary for deployment, maintenance, and scalability. Lower complexity indicates reduced dependency on high-performance cloud infrastructure.
- **Payment Support**: Evaluates the flexibility of transaction mechanisms, particularly the inclusion of cash-based options alongside digital payments, which remain crucial in many developing economies [6].

These parameters collectively reflect the practical constraints faced by both users and service providers in low-resource urban contexts.

B. Comparative Analysis

Table I and II presents a comparative overview of Chitoz and selected existing ride-sharing systems. The table highlights key differences in performance efficiency and operational features of Chitoz against existing ride-sharing systems

C. Discussion of Results

The comparison presented in this section is conceptual and design-oriented rather than an empirical performance benchmark. While these features enable efficient operation in high-demand urban markets, they require significant computational resources, persistent network connectivity, and complex backend infrastructures [3]. Such requirements may restrict accessibility in regions with limited connectivity or cost-sensitive users.

TABLE I
OPERATIONAL AND CONTEXTUAL COMPARISON OF CHITOZ WITH TYPICAL RIDE-SHARING APPLICATIONS

Aspect	**Typical Ride-Sharing App**	**Chitoz**
Target Environment	High / Moderate Connectivity	**Low-Resource Urban**
Target Device Class	Mid–High End	**Low-End**
Background Services	Multiple	**Minimal**
Feature Scope	Multiservice (rides + food, logistics, premium)	**Core Ride-Sharing Only**
Payment Methods	Cash / Digital	**Cash + Bank / QR**

Regional platforms like Pathao demonstrate improvements in localization by supporting cash-based payments and addressing region-specific mobility patterns. However, these systems continue to rely on centralized architectures that introduce moderate system complexity and operational overhead [2]. As a result, their adaptability remains constrained when deployed in infrastructure-limited settings.

Research-oriented initiatives such as Ride2Rail prioritize interoperability and multimodal integration rather than minimalism. Although effective within structured transportation ecosystems, these systems are less suitable for informal or low-resource urban contexts where simplicity and affordability are primary concerns [15].

Below is a table comparing the key performance, operational, and efficiency features of Chitoz with selected ride-sharing systems. It highlights differences in app size, resource usage, launch time, payment options, and system complexity. The table demonstrates how Chitoz achieves lightweight, efficient operation while supporting essential functionalities for low-resource urban environments.

Chitoz focuses on core ride-sharing features, minimizing complexity to enhance usability and maintainability in resource-constrained environments [10], [14].

Performance observations were obtained using Android Studio profiling tools during controlled testing of the Chitoz prototype. Comparisons with typical ride-sharing applications are qualitative and based on reported characteristics from existing literature and developer documentation. All measurements were obtained under controlled testing conditions using the device and network configuration summarized in Table A and B.

The performance-related values reported for the Chitoz system were obtained through controlled profiling of the prototype during development-time testing. In contrast, values presented for typical ride-sharing applications are indicative estimates derived from existing literature and publicly available documentation, rather than results of direct experimental benchmarking.

Overall, the results demonstrate that Chitoz provides an effective balance between functionality, usability, and system efficiency. Rather than competing directly with large-scale platforms, the system addresses a distinct gap by targeting low-resource urban transportation scenarios with a lightweight and context-aware design approach.

TABLE II
PERFORMANCE COMPARISON BETWEEN CHITOZ AND TYPICAL RIDE-SHARING APPLICATIONS

Metric	**Typical Ride-Sharing App**	**Chitoz**
Application Size (MB)	150–200 MB	**< 60 MB**
Launch Time (sec)	4–6 sec	**< 2 sec**
Avg. Data Usage / Session (MB)	2–3 MB	**~ 0.45 MB**
Background Services	Multiple	**Minimal**

VIII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presents Chitoz, a lightweight ride-sharing mobile application prototype designed to address urban transportation challenges in low-resource environments. By prioritizing essential ride-sharing functionalities and avoiding feature overextension, the system demonstrates how accessible and efficient mobility solutions can be achieved through careful software engineering design.

Chitoz adopts a modular three-tier architecture and an incremental development methodology to support systematic implementation and evaluation. Controlled testing of the prototype indicates responsive operation and reduced resource consumption within the implemented scope. Overall, this work highlights the importance of context-aware, task-focused system design and provides a practical reference for developing lightweight urban mobility applications in developing regions.

B. Future Work

Although the Chitoz prototype meets its core objectives, several enhancements could improve its performance and usability. Large-scale evaluation under high user concurrency is needed to assess scalability, particularly for handling simultaneous ride bookings, which may introduce challenges like request synchronization, race conditions, and latency. Future work could explore database scaling strategies, such as sharding, caching, and optimized indexing, along with backend load-balancing to manage peak usage efficiently. From a security perspective, the prototype could be strengthened with advanced encryption, secure key management, and behavioral analytics for fraud detection. Expanding support for digital payment ecosystems and offline-first functionality would also improve accessibility in regions with intermittent connectivity. These improvements can be introduced gradually while preserving the lightweight, prototype-based design of Chitoz.

REFERENCES

[1] Y. Rong et al., “Design and implementation of a ride-sharing mobile application,” *International Journal of Computer Applications*, vol. 176, no. 25, 2020.

[2] P. Sharma, “Study of Ride-Sharing Services in Nepal: A Case of Pathao,” Master’s Thesis, Tribhuvan University, 2021.

[3] Uber Engineering, “The Evolution of Uber’s System Architecture,” Technical Blog, Uber Technologies Inc., 2019.

[4] SPEC India, “Development Framework for Ride-Sharing Mobile Applications,” White Paper, 2020.

[5] X. Li and Y. Mi, “User experience analysis of mobile ride-hailing applications,” *International Journal of Human–Computer Interaction*, vol. 35, no. 10, 2019.

[6] Nepal Rastra Bank, “Digital Payment Systems in Nepal,” 2022.

[7] S. Shao and C. Yin, ”Security and Privacy Challenges in Connected Vehicular Cloud Computing,” in 2020 IEEE International Conference on Cyber, Physical and Social Computing (CPSCom), 2020, pp. 312-319

[8] L. Mitropoulos, A. Kortsari, and G. Ayfantopoulou, “A systematic literature review of ride-sharing platforms, user factors and barriers,” *European Transport Research Review*, vol. 13, no. 61, 2021.

[9] R. Acharya, J. Chen, and H. Xiao, “Uber Stable: Formulating the rideshare system as a stable matching problem,” *arXiv preprint arXiv:2403.13083*, 2024.

[10] M. M. I. Sourav, M. R. Islam, H. M. I. Kays, and M. Hadiuzzaman, “A structural equation modeling approach to understand user’s perceptions of acceptance of ride-sharing services in Dhaka city,” *arXiv preprint arXiv:2210.04086*, 2022.

[11] A. Jain and R. Singh, “Design and development of ride sharing mobile application,” *International Journal for Scientific Research & Development*, vol. 7, no. 2, 2019.

[12] N. Timilsena, R. Darlami, and S. B. Ghimire, “Determinants of customers’ intention to use ride-hailing services in Kathmandu Valley,” *International Journal of Artificial Intelligence Research*, vol. 6, no. 1, 2025.

[13] M. A. Abdelghany et al., “Exploring post-usage behaviour in app-based ride-sourcing services: Evidence from Egypt,” *Technological Forecasting and Social Change*, vol. 207, 2024.

[14] M. Tirachini and C. Antoniou, “An overview of the impacts of ridesourcing in developing countries: Main topics and questions for future research,” *Communications in Transportation Research*, vol. 4, 2024.

[15] F. Becker et al., “Ride2Rail: Integrating ridesharing to increase the attractiveness of rail travel,” *European Transport Research Review*, 2024.

[16] R. R. Bashir, O. R. Mathew, M. A. Atte, and O. A. Samson, “User perception and preferences in dynamic ridesharing platforms,” *African Multidisciplinary Journal of Sciences and Artificial Intelligence*, vol. 2, no. 3, 2025.

[17] M. A. Shabur and M. F. Ali, “Perspectives and possibilities for developing ride-sharing services to promote sustainable transport: Bangladesh perspective,” *Heliyon*, vol. 10, no. 12, 2024.