**Architecture Decision Record Document (ADR)**

Decision

We have decided to develop a native mobile application for both iOS and Android platforms.

Decision 1: Choosing Between Native, Web, or Hybrid App Development

Rationale

Performance: Native apps outperform web or hybrid alternatives, ensuring seamless user experiences, particularly for real-time functionalities such as tracking.

Native Features: Leveraging native development enables efficient utilization of platform-specific features like geolocation services.

User Experience: Native apps offer more intuitive and responsive interfaces, critical for ensuring usability in a transportation app context.

Offline Functionality: Native apps support offline features, allowing users access in areas with limited network coverage, a vital consideration for a transportation app's reliability.

Decision 2: Selecting UI Frameworks

Rationale

Consistency: Embracing native UI frameworks guarantees adherence to platform-specific design standards, enhancing user familiarity and ease of use.

Performance: Native UI components, optimized for each platform, ensure superior performance compared to cross-platform alternatives.

Support: Regular updates and support from Apple and Google for UIKit and Jetpack Compose respectively ensure alignment with the latest platform advancements.

Customization: Native frameworks offer extensive customization options, enabling tailored interfaces to meet the transportation company's branding and UX needs.

Decision 3: Backend Language and Framework

Rationale

JavaScript Ecosystem: Node.js enables JavaScript usage across frontend and backend, simplifying development and promoting code reuse.

Scalability: Node.js' scalability capabilities suit real-time updates and concurrent connections, crucial for features like tracking.

Community Support: Node.js benefits from a large and active community, offering diverse libraries and resources for backend functionalities.

Asynchronous Nature: Node.js' asynchronous model efficiently handles I/O operations, enhancing responsiveness by managing tasks effectively.

Decision 4: Implementation of Permission System

Rationale

User Privacy: A granular permission system enhances user control over shared data, improving privacy.

Compliance: Platform-specific permission adherence ensures compliance with app store policies, reducing the risk of rejection.

Transparency: Clear communication about permission purposes builds user trust and minimizes resistance or confusion.

Security: Restricting access to sensitive device features mitigates potential security risks like unauthorized data access.

Decision 5: Choice of Data Storage

Rationale

Data Integrity: RDBMS like PostgreSQL ensure robust data integrity, enhancing consistency and reliability.

Scalability: PostgreSQL's scalability features accommodate growing data volumes, crucial for an expanding app.

ACID Compliance: PostgreSQL guarantees ACID compliance, essential for maintaining data consistency, especially for financial transactions.

Query Performance: PostgreSQL's optimization capabilities enhance query performance, ensuring a seamless user experience.

Decision 6: Adoption of Additional Frameworks or Technology Stacks

Rationale

Mapping Service: Google Maps offers accurate geolocation and customizable mapping, meeting ride-related display needs.

Real-time Updates: WebSocket technology enables bidirectional communication for seamless real-time updates, crucial for tracking functionalities.

Scalability: WebSocket's efficiency suits handling concurrent connections and continuous data streaming.

Decision 7: Integration of Payment Gateway

Rationale

Security: Stripe's robust security measures ensure safe payment transactions and protect user financial information.

Developer-Friendly: Stripe's APIs and SDKs simplify integration and customization of payment functionalities.

Global Support: Stripe's wide currency and payment method support cater to diverse user bases across regions.

Transaction Fees: Competitive fees minimize processing costs, enhancing app cost-effectiveness.

Decision 8: Selection of Push Notification Service

Rationale

Cross-Platform Support: Firebase Cloud Messaging ensures consistent delivery across iOS and Android platforms.

Reliability: FCM's reliability and scalability handle high volumes of notifications without disruptions.

Personalization: FCM enables personalized notifications, enhancing user engagement.

Message Targeting: FCM's targeting features ensure relevant communications, crucial for ride-related updates and announcements.