Architectural Decision Record (ADR) – Transportation App:

Introduction:

This Architectural Decision Record (ADR) outlines the key architectural decisions made for the development of a mobile app for modern transportation company. The app aims to provide users with the ability to book and track rides, view driver details and ride history, and make payments securely.

Decisions:

Native Mobile App:

Decision: We have decided to develop a native mobile app for both iOS and Android platforms.

Rationale:

* Native apps offer superior performance and user experience compared to web or hybrid apps.
* By leveraging platform-specific features and optimizations, we can provide seamless integration with the underlying operating systems.
* Native development allows us to access device functionalities such as GPS, push notifications, and camera with ease, enhancing the app’s functionality.

UI Framework: React Native

Decision: We will utilize React Native as the UI framework for developing the app.

Rationale:

* React Native allows for cross-platform development, enabling us to write code once and deploy it on both iOS and Android platforms.
* It offers a rich set of pre-built UI components and libraries, speeding up the development process.
* React Native provides a native-like performance while offering the flexibility of JavaScript for development.

Backend Language: Node.js

Decision: The backend of the app will be developed using Node.js.

Rationale:

* Node.js offers non-blocking, event-driven architecture, making it well-suited for real-time applications like ours.
* It provides a vast ecosystem of libraries and frameworks, facilitating rapid development and deployment.
* Node.js allows for easy scalability, which is crucial for handling a large number of concurrent connections in a transportation app.

Permissions:

Decision: The app will request the following permission:

* Location access: To track the user’s location for ride booking and tracking purpose.
* Camera access: To enable users to upload a profile picture and for driver verification purposes.
* Stroage access: To cache data locally for offline mode and store user preferences.

Rational:

* Location access is essential for providing ride-related services such as booking, tracking, and navigation.
* Camera access is required for user verification and authentication, enhancing security.
* Storage access enables offline functionality, ensuring a seamless user experience even in low connectivity areas.

Data Storage: Relational Database

Decision: We will use a relational database (e.g., PostgreSQL) to store user profiles, ride history, and payment information.

Rational:

* Relational databases offer ACID compliance, ensuring data integrity and consistency, which is crucial for financial transactions.
* Relational databases provide robust querying capabilities, enabling complex data retrieval and analysis.
* The structured nature of relational databases makes it easier to maintain data consistency and enforce relationships between entities.

Additional Framework: Google Maps API

Decision: We will integrate the Google Maps API for geolocation services and mapping functionalities.

Rationale:

* Google maps API provides accurate geolocation services, essential for tracking ride locations and estimating arrival times.
* It offers rich mapping functionalities, including route optimization, traffic data, and turn-by-turn navigation.
* Integration with Google Maps enhances the user experience by providing familiar and reliable mapping services.

Follow-up Actions:

* Develop and test the app’s frontend and backend components using the selected technologies. Implements user authentication and authorization mechanisms to ensure secure access to user data.
* Integrate Google Maps API for geolocation services and implement real-time ride tracking functionality.
* Conduct thorough testing to ensure the app’s performance, reliability, and security.
* Monitor user feedback and app analytics to identify areas for improvement and iterate on the app’s features.

Conclusion:

By making these architectural decisions, we aim to develop a robust and user-friendly transportation app that meets the needs of both passengers and drivers while security, scalability, and performance.