**Food Ordering App ADR**

1. Hybrid App

Our decision to use a hybrid app using React Native is driven by the necessity for cross-platform functionality without sacrificing the native user experience. React Native lets us maintain a consistent performance across both iOS and Android platforms, an important factor for user retention and satisfaction.

2. UI Framework

React Native serves as the base of our user interface, making use of additional libraries for enhanced functionality. React Navigation will facilitate the user flow throughout the app, while Redux will manage the application state. The incorporation of react-native-paper will help promote an aesthetic and responsive user interface.

3. Backend Language

We’re going to use Node.js and Express for our backend. This setup can handle real-time data, which is crucial for features like live order tracking and location updates. It’s about keeping the app quick and accurate.

4. Permissions

Permissions are straightforward with the react-native-permissions library. We're only asking users for access when we really need it. This means we'll request location data for restaurant suggestions, send notifications for order updates, and maybe use the camera for new features down the line, but always respecting user privacy.

5. Data Storage

We're setting up data storage on two fronts: AsyncStorage for quick access to local data and MongoDB for the heavy-duty cloud storage. This way, users can pull up data quickly, and we can manage anything from user profiles to large restaurant menus without a hitch.

6. Additional Frameworks or Technological Stacks

Firebase is in for user authentication and real-time database stuff. Axios is our pick for talking to APIs, and Socket.IO is key for those live updates during order tracking. We're using the Google Maps API for reliable location services, and we're sorting out various Payment Gateway SDKs to give users secure payment options.