

Scenario One

This document offers the ADR decisions for the first scenario.

ADR 1 - Native/Web/Hybrid App

Decision - Native Mobile App

Developing a native app would be the best course of action due its performance, reliability, and the ability to handle complex functionalities like offline mode and push notifications. Native apps are suited for handling large data sets, aligning with the retail company's requirement for managing multiple products and orders. Native apps also offer a seamless and a responsive user experience which enhances the experience for customers and administrators. Native apps are able to store data locally which is essential for the implementation of offline mode per the retail company's request. Data would be synced with servers once an internet connection is established.

ADR 2 - UI Framework

Decision - ReactNative

Using ReactNative allows us to write a single codebase for both iOS and Android platforms, reducing development time compared to if we created separate codebases. ReactNative offers native performance and a multitude of UI components which allows for high quality user experience. Lastly, ReactNative has a large community which provides developers with access to multiple resources and support. ReactNative also offers seamless integration with native APIs and functionalities like push notifications and offline mode.

ADR 3 - Backend Language

Decision - Node.js

Node.js allows us to handle high traffic that would come from the retail app. Due to the fact that it is built on event-driven architecture, the app would be able to manage multiple requests without being slowed. This is crucial to ensure the experience of customers is seamless and efficient. Node.js uses JavaScript, a language well known in development, meaning developers would have a reduced learning curve when creating the app opposed to other languages.

ADR 4 - Permissions

Decision - Fine Grained Permissions Management

Implementing fine-grained permissions management enhances user trust and creates a more transparent experience. By only requesting the permissions we truly need and doing so at appropriate times, we show our users that we respect their privacy and are committed to protecting their personal information. This transparency helps build a positive relationship with our customers, as they can feel confident that their data is being handled responsibly. Additionally, managing permissions in this way ensures compliance with privacy regulations by minimizing unnecessary data access. This not only protects the company from potential legal issues but also aligns with our commitment to ethical practices. By being judicious with permission requests, we create a user-friendly app that prioritizes user consent and security, ultimately fostering greater trust and loyalty among our users.

ADR 5 - Data Storage

Decision - Local/Cloud Storage

Using a combination of local and cloud storage ensures that the app can provide a seamless offline experience while also managing large amounts of data efficiently. Local storage allows users to browse products and view their order history even when they are offline, fulfilling one of the key requirements of the retail company. Once an internet connection is available, the app can synchronize data with the cloud, ensuring that all information is up-to-date. Cloud storage, on the other hand, is ideal for storing large datasets such as product images and details. It provides scalability and flexibility, allowing the app to handle a growing amount of data without performance degradation. This approach also ensures data redundancy and security, as cloud storage solutions typically include built-in data protection and backup mechanisms.

ADR 6 - Additional Frameworks/Technology Stacks

Push Notifications - Firebase Cloud Messaging

Payment - Stripe

Firebase Cloud Messaging is selected for push notifications due to its reliability and ease of integration with both iOS and Android platforms. It allows us to send targeted notifications to users, keeping them informed about order updates, new product arrivals, and exclusive offers. Firebase Cloud Messaging provides robust features for

managing notifications, including scheduling, segmentation, and analytics, which help in maintaining high user engagement. Stripe is chosen for payment processing because of its strong security features and user-friendly interface. Stripe supports multiple payment methods and currencies, which is essential for a retail app that aims to expand its customer base internationally. Its comprehensive API and documentation make integration straightforward, ensuring seamless and efficient transactions for users. Additionally, Stripe's advanced fraud detection mechanisms help protect both the company and its customers from potential security threats.