# ADR 001 App Type

Status: [Accepted] Date: [2024-03-15]

## Context

The team is being asked by the retail company to develop a new mobile app that supports offline mode and push notifications.

## Options

* Native
* Web
* Hybrid

## Decision

The team has decided to focus on building a native app for the retail company.

## Rationale for Decision

* As web and hybrid app would require internet connection, it would be difficult to implement offline support, therefore the team has decided best to stick to building a mobile app.
* To implement push notifications into the app would require usage of a mobile app.

# ADR 002 UI Framework

Status: [Accepted] Date: [2024-03-15]

## Context

## Options

* React Native
* Ionic
* Native Script

## Decision

The team will use React Native as the UI framework.

## Rationale for Decision

* The team is much more familiar and experienced in app development using React Native.

# ADR 003 Backend Language

Status: [Accepted] Date: [2024-03-15]

## Context

The team is tasked with selecting a backend language for developing the server-side components of a new mobile app for a retail company. The app will facilitate various functionalities including user authentication, product management, order processing, and integration with third-party services.

## Options

* Kotlin
* Java
* Node.js

## Decision

After careful consideration and evaluation of various options, we have decided to use Node.js the backend language for the development of the mobile app.

## Rationale for Decision

* Performance and Scalability: Node.js is known for its event-driven, non-blocking I/O model, which makes it well-suited for handling asynchronous operations and high concurrency. This architecture allows for efficient handling of multiple client requests, ensuring optimal performance and scalability, which is crucial for a mobile app with potentially high user traffic.
* Developer Productivity: JavaScript is a widely-used and versatile language, familiar to many developers. Leveraging Node.js allows us to utilize JavaScript both on the frontend (with frameworks like React Native) and the backend, facilitating code reuse, easier collaboration among developers, and faster development cycles. Additionally, the extensive npm ecosystem provides a vast array of reusable packages and libraries, enabling rapid development of server-side functionalities.

# ADR 004 Permissions

Status: [Accepted] Date: [2024-03-15]

## Context

In the development of our mobile app for a retail company, we need to define and implement permissions management to control access to sensitive functionalities and data within the application.

## Decision

We will implement a role-based access control (RBAC) system for managing app permissions.

## Rationale for Decision

* Developer Productivity: RBAC with JWT authentication complements the developer productivity advantages of Node.js
* By validating JWT tokens on the server side, we can ensure that only authenticated and authorized users have access to protected resources

# ADR 005 Data Storage

Status: [Accepted] Date: [2024-03-15]

## Context

We are tasked with selecting a data storage that must meet the requirements for offline mode support, synchronization, scalability, performance, security, and compatibility with the app’s features.

## Decision

The team will utilize a combination of SQLite for offline mode support and synchronization and a cloud-based database such as Firebase Firestore for real-time data updates and scalability.

## Rationale for Decision

* Offline Mode Support: SQLite is a lightweight, embedded relational database that is well-suited for mobile applications, providing local storage capabilities for offline mode.
* Compatibility with Payment Gateways: SQLite and Firebase Firestore provide flexibility in integrating with various payment gateways, allowing secure and convenient transactions for customers.