| **Title** | Offline mode and Data Sync |
| --- | --- |
| **Context** | To improve the app's user experience, it must support offline use so that customers can browse products without wifi. This requires a program to sync with the server when online and a program to store data locally |
| **Decision** | The company will use SQL as the database for storing info, for the data sync, the company will implement a sync service that will trigger every time the device is connected to the internet. Offline changes will be added to a local queue to be updated |
| **Consequences** | Positives   * Allows users to browse the app and view their order history while being offline * Sql is efficient with structured data, therefore reducing the load on the network   Negatives   * Increased storage requirement on mobile device * Increased complexity due to data sync conflicts while being offline |

| **Title** | Push Notifications |
| --- | --- |
| **Context** | The app should be able to send notifications to users for updates on orders, new items, and limited-time offers. This requires a push notification service that runs on both IOS and Android |
| **Decision** | The company will use push owl for the notification service. It provides advanced user targeting features, along with a user-friendly UI, allowing the company to send notifications with ease. |
| **Consequences** | Positives   * PushOwl allows highly targeted push notifications, improving click rate and retention * Relatively easy integration   Negatives   * Requires a subscription at a larger scale, therefore increasing costs * Limited compatibility with analytical softwares |

| **Title** | Payment Gateways |
| --- | --- |
| **Context** | To ensure secure and convenient transactions, the mobile app should have multiple payment gateways that are handpicked by the team based on security, ease of use, and compatibility |
| **Decision** | The company will use Stripe and Paypal as their main payment gateway due to its amazing security and ease of use for both the devs and users |
| **Consequences** | Positives   * Both payment gateways are secure and trusted by users globally * Have a large payment method support   Negatives   * Higher fees compared to other regional gateways |

| **Title** | **User Behaviour Tracking and Loyalty Program** |
| --- | --- |
| **Status** | **Proposed** |
| **Context** | Building this feature to analyze the user behavior on our mobile app so that we can view detailed insights and metrics to improve the app’s performance, user experience, and the company’s revenue. |
| **Decision** | The app will store the product view and purchase history in an SQL database which will be used to create analytics and provide users with recommendations of loyalty programs that will improve user experience. |
| **Consequences** | **Positives:**   * Users will be provided with a customized list of products in their feed according to their history and purchases which they will likely be purchasing again. * Introduction to the loyalty system will improve user experience by providing loyal customers who have signed up for the loyalty program with discounts and coupons which the user alongside the company can benefit from. * System developers can benefit from the insights of the data from the users regarding their purchases and product view history so that they can push the products that are more in demand.   **Negatives:**   * Privacy Concerns may arise among users since the system will be tracking users’ behavior, including product views and purchases. Especially if the data is being collected without transparency. * Storing users’ behavior in an SQL database might be vulnerable if sensitive data is not properly encrypted. If a data breach happens then it might affect the company’s reputation. * Tracking users’ behavior throughout the app may cause performance overhead since we are storing those data in our SQL database, this is causing repetitive calls to the backend which can drastically put load on users’ devices. |

| **Title** | Product Image Storage and Optimizing Strategy |
| --- | --- |
| **Status** | Proposed |
| **Context** | Our mobile app will display product images varying in size and resolution in an optimal manner. To ensure proper performance and a seamless user experience, the development team needs to find better solutions for storing, optimizing, and displaying images efficiently. |
| **Decision** | For storing images, the development team will be using an Amazon S3 bucket to store the images.  To optimize image viewing we can compress the image slightly with very little to no effect on the image itself. Also, we can use lazy loading so that we load an image only when it is in view. We can also use caching technology to store images temporarily on users' end which regularly comes into view. |
| **Consequences** | **Positives:**   * Reduced loading time for the application and the images. * Enhanced user experience: The images load faster, and the user can use the application seamlessly without any obstruction. * Since the images are stored by compressing them the data usage is drastically decreased which is particularly beneficial for user with limited data plans on mobile phones.   **Negatives:**   * Using Amazon S3 and using algorithms to compress the images may increase the complexity of building the application. * The storage costs might still result in excess though the use of compression can reduce the size but storing a lot of images is definitely going to raise the cost. |