Audit Report

We, Daniel Bernal, Daniel Gómez and Juan Pablo Martinez, are part of a company in charge of analyzing Android mobile applications reporting any technical issues found for the source code. For this audit, we are going to give a free analysis of Tachiyomi. Tachiyomi is a comprehensive manga reading application that offers a rich set of features, extensive customization options, and a vibrant community, making it a popular choice for manga enthusiasts seeking a versatile and user-friendly reading experience. For the following analysis we are going to focus on four main aspects: Eventual Connectivity Strategies, Caching Strategies, Memory Management Strategies and Multi-threading Strategies.

Tachiyomi effectively leverages Kotlin coroutines to manage asynchronous tasks, prioritizing user preferences and lifecycle management. The application also implements a fixed-size thread pool to handle concurrent tasks, ensuring seamless execution of network operations and image loading without compromising the user interface's responsiveness. Additionally, Tachiyomi integrates various strategies to support asynchronous network operations, crash reporting initialization, and notification channel configuration. With a focus on delivering smooth UI updates, the application uses asynchronous processes for image loading and display, ultimately enhancing the overall user experience. To further enhance Tachiyomi's threading and concurrency strategies, it is recommended to conduct regular performance testing and profiling, implement comprehensive error handling and exception management, introduce additional monitoring and logging mechanisms, and stay updated with evolving best practices and technological advancements in the field.

Tachiyomi exhibits a robust overall connectivity strategy, effectively balancing offline functionality and online content access. The application's use of caching and fetching strategies is commendable, ensuring efficient data retrieval and a smooth user experience. In the case of manga images, Tachiyomi wisely adopts a cache-first approach, minimizing unnecessary network requests and providing users with quick access to locally stored data. This is a prudent choice, as it reduces data usage and enhances the application's performance. Furthermore, the application demonstrates a "cache then network" approach when handling cover images, ensuring users can access cached images while allowing for background downloads when cover images are updated. These strategies enhance the user experience by maintaining a delicate balance between data efficiency and content access speed.

Despite its overall strong connectivity design, Tachiyomi does suffer from certain connectivity anti-patterns that warrant attention. The most prominent issue is the occurrence of "Stuck Progress Notifications" (Anti-pattern #2). In cases where the application was initially launched with an internet connection and then the internet is deactivated, Tachiyomi wrongly assumes a continuous internet connection. As a result, the app gets stuck on a progress notification, incorrectly believing that it still has an active connection. This can lead to user frustration and a less-than-optimal experience. Furthermore, the "Non-informative Message" anti-pattern (Anti-pattern #3) occurs under similar circumstances. The application, in the absence of network connectivity at startup, displays uninformative error messages. These messages can confuse users and hinder their ability to understand and resolve the issue.

To improve Tachiyomi's connectivity and rectify these anti-patterns, the application can implement a listener to monitor real-time connectivity changes. This solution will allow Tachiyomi to respond effectively to shifts in connectivity status. By monitoring network status and providing informative messages when necessary, the app can offer a smoother and more intuitive user experience. Implementing such a feature would not only enhance user satisfaction but also align with best practices in connectivity handling.

In summary, while Tachiyomi's overall use of connectivity patterns is strong, there is room for improvement in addressing the identified anti-patterns related to connectivity handling. By adopting the suggested solution of implementing connectivity listeners and providing informative messages, Tachiyomi can further enhance its reliability and responsiveness to connectivity changes, ensuring a more seamless user experience.