In developing the Transaction Tracker app using Flutter for the frontend and Spring Boot for the backend, several challenges arose, particularly in ensuring seamless communication between the two components. One major challenge was configuring the API endpoint for the Flutter app, especially when testing on different devices. To address this, I implemented a configuration menu that allows users to easily input the IP address and port number of the Spring Boot server, making it adaptable for both local and remote testing. Additionally, managing state and error handling during API calls proved to be complex. We utilized GetX for state management, which simplified the process of managing the app's state and provided reactive programming features. This ensured that users received immediate feedback on their actions, such as successful logins and transaction submissions, thereby enhancing the overall user experience. Furthermore, to enable internet access for API calls, I included the necessary permission in the Android manifest: `<uses-permission android:name="android.permission.INTERNET" />`. Through these solutions, I created a robust and user-friendly application that effectively meets its intended functionality.