

Abstract (DE)

Die vorliegende Masterarbeit befasst sich mit dem Thema der plattformübergreifenden Entwicklungsansätze für mobile Anwendung für die beiden größten mobilen Plattformen Android und iOS sowie der Performanz-Analyse typischer UI-Interaktionen. Dazu werden die folgenden zwei Forschungsfragen bearbeitet: *Wie kann die Performanz von plattformübergreifenden Entwicklungsansätzen einheitlich gemessen werden?* sowie *Wie unterscheidet sich die Performanz von plattformübergreifenden Entwicklungsansätzen unter der Nutzung typischer UI-Interaktionen?* Um die Forschungsfragen zu beantworten, wurde am Anfang eine umfassende theoretische Einführung geben, um das notwendige Basiswissen für den praktischen Teil zu erarbeiten. Für den praktischen Teil wurden fünf mobile Anwendungen implementiert. Diese beinhalteten drei plattformübergreifende Anwendungen, welche mit React Native, Flutter und Ionic/Capacitor entwickelt wurden und zusätzlich, zwei native Anwendungen für Android und iOS. Danach wurde der Ressourcenverbrauch (CPU und RAM) während der Ausführung von drei typischen UI-Interaktionen sowohl auf einem Android- als auch einem iOS-Gerät gemessen. Zu den untersuchten UI-Interaktionen zählen das *Öffnen und Schließen des Navigation Drawers*, der *Übergang zwischen zwei Bildschirmen* sowie das *Scrollen durch eine virtuelle Liste*. Die Recherche zeigte, dass es noch kein Framework oder Tool gibt, welches Ressourcenmessungen sowohl auf einem Android- als auch einem iOS-Gerät einheitlich zulässt. Die Testdurchläufe wurden daher mit zwei unterschiedlichen Tools durchgeführt. Die Ergebnisse zeigen, dass die plattformübergreifend programmierten Anwendungen durchwegs mehr CPU in Anspruch nehmen, als die nativen Gegenstücke. Der Arbeitsspeicherverbrauch ist auf Android von der Interaktion abhängig und auf iOS benötigt die Ionic/Capacitor Anwendung gleich viel Speicherplatz wie die native Anwendung. Die React Native Anwendung verbraucht deutlich mehr.

Abstract (EN)

This master thesis deals with the topic of cross-platform development approaches for mobile applications for the two largest mobile platforms Android and iOS as well as performance analysis of typical UI interactions. The following two research questions are addressed: *How can the performance of cross-platform development approaches be measured uniformly* and *How does the performance of cross-platform development approaches differ when considering typical UI interactions?* To answer the research questions, a comprehensive theoretical introduction was given at the beginning to provide the necessary basic knowledge for the practical part. For the practical part five mobile applications were implemented. These included three cross-platform applications developed with React Native, Flutter and Ionic/Capacitor and additionally, two native applications for Android and iOS. Afterwards, the CPU usage and memory consumption were measured during the execution of three typical UI interactions on both Android and iOS devices. The analysed UI interactions include the *opening and closing of the Navigation Drawer*, the *transition between two screens*, and the *scrolling through a virtual list*. The research showed that there is no framework or tool that allows the measurement of resources on both Android and iOS devices in a consistent manner. The test runs were therefore performed using two different tools. The results show that the cross-platform applications consistently consume more CPU than their native counterparts. The memory consumption on Android depends on the interaction and on iOS the Ionic/Capacitor application requires the same amount of memory as the native application. The React Native application consumes significantly more.