
Technology Arts Sciences TH Köln

Exposé für eine Bachelorarbeit zum Thema:

**Vergleich der Cross-Plattform Frameworks React Native, Flutter und .NET
MAUI auf Basis einer Testanwendung im Hinblick auf ihre Performance unter
Android und Windows**

Vorgelegt von:

Manuel Bäcker
Tecklinghauser Weg 24
57462 Olpe
E-Mail: manuel.b.97@gmx.de

Matrikelnummer: 11117600
BA Allgemeine Informatik
Datum: 22.03.2023

Einleitung und Problemstellung

Das Angebot an Cross-Plattform Frameworks zur Erstellung von Anwendungen für verschiedene Plattformen auf einer einzelnen Code-Basis hat sich in den letzten Jahren deutlich vergrößert. Neben den älteren Technologien wie Ionic, Cordova, Electron oder Xamarin sind neue Frameworks wie bspw. React Native (Meta) und Flutter (Google) hinzugekommen. Zuletzt wurde mit .NET MAUI von Microsoft am 09.08.2022 eine Weiterentwicklung von Xamarin veröffentlicht. Flutter und React Native haben sich unter Mobile- und Webentwicklern als robuste und effiziente Lösungen bewährt, und sind dadurch sehr populär geworden [6]. Diese Arbeit hat einen Vergleich der drei Technologien Flutter, React Native und MAUI zum Ziel, um zu untersuchen, ob .NET MAUI eine leistungsstarke Alternative darstellt und ob die Performance gleichwertig oder sogar besser als die von React Native und Flutter ist. Im theoretischen Teil der Arbeit wird der Hintergrund sowie der aktuelle Stand aus dem Bereich Cross-Plattform Entwicklung und die zu untersuchenden Frameworks sowie deren zugrunde liegenden Programmiersprachen, Architekturen und Funktionsweisen erläutert. Im praktischen Teil der Arbeit werden die Technologien anschließend anhand verschiedener Metriken verglichen. Hierzu wird eine Testapplikation in Form eines Tierlexikons konzipiert und in allen der zu untersuchenden Frameworks implementiert. Die Applikation fungiert als Basis für den Vergleich verschiedener Metriken aus den Bereichen Performance und Entwicklungsaufwand und wird auf den Plattformen Android und Windows für die Messung der geplanten Metriken eingesetzt. Zudem wird der Einrichtungsprozess der Frameworks auf einem Windows-PC und die grundlegenden Konzepte, auf denen die Frameworks basieren verglichen, um eine Aussage zur Einarbeitungszeit und dem Schwierigkeitsgrad der Erlernbarkeit treffen zu können.

Thema und Fragestellung

Die Arbeit greift die Fragestellung auf, ob das Cross-Plattform Framework .NET MAUI von Microsoft eine leistungsstarke Alternative zu etablierten Frameworks wie React Native und Flutter darstellt. Zudem soll festgestellt werden, welches der drei Frameworks nach aktuellem Stand das performanteste ist und in verschiedenen, extra für diesen Zweck angefertigten Testszenarien innerhalb einer in allen zu untersuchenden Frameworks implementierten Testanwendung, die höchste Leistungsfähigkeit aufweist. Eine weitere Fragestellung ist, welches Framework die am einfachsten zu erlernenden Konzepte mit sich bringt und daher tendenziell am leichtesten für neue Entwickler zu erlernen ist. Zuletzt soll untersucht werden, welches Framework die geringste Anzahl an Codezeilen für die Implementierung einer funktional äquivalenten Anwendung benötigt. Da .NET MAUI noch sehr neu ist und nach ersten Recherchen keine Forschungsarbeiten existieren, die gezielt die Leistung und Performance des Frameworks im Vergleich zu etablierten Lösungen untersuchen, ist dies ein interessantes Forschungsvorhaben, um die Entscheidungsfindung bei der Projektplanung von Cross-Plattform-Applikationen zu unterstützen, und eine präzisere Aussage über die Features und Leistungsdaten des Frameworks treffen zu können. Zusammenfassend lässt hieraus folgende Leitfrage als Forschungsgegenstand ableiten: Weist .NET MAUI in gleichen Anwendungsszenarien eine gleichwertige oder sogar bessere Performance als React Native und Flutter auf?

Leitfrage und logisches Gerüst

Die Arbeit soll primär die Fragestellung beantworten, ob .NET MAUI im Vergleich zu Flutter und React Native eine leistungsstarke Alternative bei der Entwicklung von Cross-Plattform Applikationen darstellt und ob die Performance des Frameworks gleichwertig oder besser ist als die von React Native und Flutter. Auf Basis der Literaturrecherche werden bereits vorhandene Forschungsergebnisse erfasst um erste Erkenntnisse zu diesem Themenbereich zusammenfassen und analysieren zu können. Am Ende der Arbeit kann dadurch beantwortet werden, ob sich diese Ergebnisse tendenziell bestätigen lassen oder stark abweichen. Hieraus kann ebenfalls abgeleitet werden, ob React Native und Flutter durch ihre stetige Weiterentwicklung signifikante Verbesserungen in ihrer Performance im Vergleich zu älteren Forschungsergebnissen aufweisen. Zudem wird der Einrichtungsprozess sowie die Architekturen und Konzepte, auf denen die Frameworks basieren, verglichen. Hieraus kann zusätzlich abgeleitet werden, welches der Frameworks sich am besten für weniger erfahrene Entwickler eignet, welche Konzepte leicht zu verstehen sind und welches Framework sich am schnellsten in den Entwicklungsprozess von Entwicklerteams beliebiger Größe integrieren lässt. Um dieses Vorhaben durchführen zu können, wird eine Testanwendung in allen drei Frameworks erstellt und auf Basis dieser Anwendung und wiederum auf ihr basierenden Testszenarien verschiedene Metriken wie Arbeitsspeichernutzung, Startzeit der Anwendungen und Größe der jeweiligen Installationspakete verglichen. Alle zu untersuchenden Kriterien und Metriken werden genauer im Kapitel „Vergleichskriterien und Auswertung“ beschrieben. Aus den Forschungsfragen sowie der Planung und des Vorgehens ergibt sich folgende vorläufige Gliederung als Grundgerüst für die Arbeit, welche auf der nächsten Seite dargestellt wird.

Vorläufige Gliederung

- Abstract/Einleitung
 - Grundlagen
 - Cross Plattform Entwicklung
 - Vor- und Nachteile
 - Aktuelle Marktsituation
 - React Native
 - Flutter
 - .NET MAUI
 - Konzeption der Testapplikation
 - Funktionsumfang
 - Vergleichskriterien
 - Views/Pages
 - Serverkomponente
 - Testszenarien
 - Umsetzung
 - React Native Implementierung
 - Flutter Implementierung
 - .NET MAUI Implementierung
 - Framework Vergleich
 - Verwendete Hardware
 - Unterstützte Plattformen
 - Unterstützte Hardware Features
 - Vergleich der Codezeilen
 - Größe des Installations-Pakets
 - Größe der installierten Anwendung
 - Startzeit der Anwendung
 - Wiederaufnahmezeit nach Deaktivierung
 - Ressourcennutzung nach Anwendungsstart
 - RAM-Nutzung
 - CPU-Nutzung
 - Rendering Performance
 - Frame Times
 - Bilder pro Sekunde (FPS)
 - FPS-Stabilität bei Animationen und Übergängen
 - Testszenario 1 (Http Request und Listenansicht Initialisierung)
 - Komplettierungszeit
 - RAM-Nutzung
 - CPU-Nutzung
 - Testszenario 2 (Embedded Camera View öffnen und ein Bild speichern)
 - Komplettierungszeit
 - RAM-Nutzung
 - CPU-Nutzung
 - Evaluierungsmatrix
 - Fazit
 - Ausblick
-

Aktueller Stand der Forschung

Es existieren einige Arbeiten, welche die Frameworks Flutter und React Native in ihrer Performance vergleichen. Zu .NET MAUI wurde allerdings noch sehr wenig veröffentlicht und auf Basis erster Recherchen noch keine Forschungsarbeiten, welche die Performance des Frameworks genauer untersuchen. Mah'dar (2021) thematisiert in seiner Arbeit aktuelle Entwicklungsansätze im Bereich der mobilen Entwicklung und vergleicht .NET MAUI in Bezug auf seine Features mit seinem Vorgänger Xamarin.Forms und beschreibt die Neuerungen des Frameworks [1]. Der Einblick von Mah'dar kann durch dieses Forschungsvorhaben gerade in Bezug auf die Performance und den Vergleich mit etablierten Frameworks wie React Native und Flutter sehr gut ergänzt werden. Húdek (2022) beschäftigt sich in seiner Arbeit mit der Anfertigung von Übungsaufgaben innerhalb des Frameworks, um Studenten einen Einblick in die Anwendungsentwicklung mit .NET MAUI zu ermöglichen [2]. Kurfürst (2022) entwickelt innerhalb seiner Arbeit eine Applikation für Investments mit dem Framework .NET MAUI. Daraus geht hervor, dass obwohl MAUI sich zum damaligen Zeitpunkt noch in der Preview-Phase befunden hat, es als nützliches Framework zur Entwicklung von Cross-Plattform Anwendungen zu bewerten ist und als Resultat eine funktionale Applikation entstanden ist, die mit einigen weiteren Anpassungen und Features veröffentlicht und im produktiven Umfeld genutzt werden kann [3]. Wu (2018) hat einen Performancevergleich zwischen Flutter und React Native durchgeführt, auf den sich weitere Arbeiten in diesem Bereich stützen [4]. Hier können wichtige Ergebnisse referenziert werden, um zu untersuchen, wie sich Flutter und React Native im Vergleich zu einer aktuellen Untersuchung von ihrer Performance verhalten, und ob sich die Ergebnisse von Wu bestätigen lassen, oder diese durch die stetige Weiterentwicklung der Frameworks eine Abweichung aufweisen. Jagiello (2019) vergleicht ebenfalls die Frameworks Flutter und React Native und stellt dabei eine Abweichung zu den Ergebnissen von Wu fest, die er mit den Hintergrundaktivitäten des verwendeten Betriebssystems begründet [5]. Es ergibt sich im Bereich der Untersuchung der Leistung von .NET MAUI daher eine Forschungslücke, die mit dieser Arbeit aufgegriffen werden soll und dabei gleichzeitig Ergebnisse zum aktuellen Stand der beiden anderen Cross-Plattform Frameworks liefert.

Untersuchungsansatz

Um die Frameworks optimal vergleichen zu können, wird eine Testapplikation in Form eines Tierlexikons in allen der drei Frameworks implementiert und auf Basis dieser Anwendung verschiedene Testszenarien durchgeführt. Zudem wird eine kleine Serverkomponente implementiert, welche eine Rest-Schnittstelle bereitstellt, durch die Daten über verschiedene Tiere mit dem HTTP-Protokoll im JSON-Datenformat abgerufen werden können. Eine Anwendung dieser Form wird gewählt, da ihre grundlegenden Funktionalitäten wie der Abruf von Daten über eine Rest-Schnittstelle einen in vielen im produktiven Umfeld eingesetzten Apps verwendeten Anwendungsfall abbilden, und daher als möglichst repräsentativ für die Messung von Leistungsdaten zu klassifizieren ist. Zudem werden bei dieser Form von Anwendung grundlegende und häufig genutzte Komponenten für Benutzeroberflächen wie bspw. Listenansichten verwendet, um die Performance auch im Hinblick auf die Nutzerinteraktion möglichst präzise bewerten zu können. Um die Performance und weitere Metriken präzise erfassen zu können, werden diverse Profiling- und Entwicklertools, welche Bestandteil der Framework-SDKs und Entwicklungsumgebungen sind, benutzt. Außerdem ist geplant, sofern möglich bereits auf dem Betriebssystem vorhandene Tools wie Task-Manager o.ä. zur Sammlung von Leistungsdaten zu verwenden um dadurch den Overhead und damit den Einfluss von Debug-Builds bei der Messung von Werten zu minimieren. Einen weiteren Bereich der Testapplikation stellt der Zugriff auf die plattformspezifischen Hardware Schnittstellen und Funktionen dar. Hierbei werden bspw. Systeminformationen ausgelesen oder auf Hardware-Komponenten wie Kamera und Sensorik zugegriffen. Dabei werden Performance und die Verfügbarkeit für das jeweilige Framework

verglichen um zu analysieren, ob jedes der Frameworks den gleichen Funktionsumfang bietet. Zudem kann hierdurch der Aufwand der Implementierung dieser Features für jedes Framework analysiert und verglichen werden.

Vergleichskriterien und Auswertung

Für den Vergleich der Frameworks im Hinblick auf ihre Performance werden folgende Kriterien verwendet. Alle Metriken werden auf Basis von in der Testanwendung geschaffenen Testszenarien gemessen und ausgewertet. Bisher sind zwei Szenarien geplant, zum einen die Beschaffung von Daten über die Serverkomponente via HTTP-Protokoll und die Einspeisung dieser Daten in eine Listenansicht. Zum anderen soll ein Szenario geschaffen werden, indem aus der Testanwendung heraus die native Kamera-Applikation des Betriebssystems ausgeführt und ein Foto gespeichert wird. Zunächst wird die Ressourcennutzung untersucht, hierbei werden CPU-Auslastung und Arbeitsspeicher-Nutzung über den Zeitraum der ausgeführten Testszenarien aufgezeichnet und verglichen. Des Weiteren erfolgt eine Analyse der Rendering Performance um zu vergleichen, wie flüssig bspw. Animationen dargestellt werden können und um zu analysieren, ob Differenzen bestehen und sich diese merkbar auf die Nutzererfahrung auswirken. Hierbei werden über den Zeitraum der Testszenarien die gerenderten Bilder pro Sekunde (FPS) sowie die Bildzeit analysiert. Zwei weitere essentielle Vergleichskriterien sind die Start- und Wiederaufnahmezeit der Anwendungen. Diese Zeitwerte haben großen Einfluss auf die Nutzererfahrung, da Sie diejenigen Performance-Merkmale sind, die ein Nutzer bei der Interaktion mit einer Anwendung als erstes wahrnimmt. Zudem werden quantitative Metriken wie die Größe des Installationspakets für die jeweilige Plattform, die Größe der installierten Anwendung und die jeweils unterstützten Plattformen verglichen. Die Größe der Anwendung hat ebenfalls einen Effekt auf die Nutzererfahrung und der Vergleich der unterstützten Plattformen ermöglicht es eine Aussage darüber zu treffen, wie flexibel ein Entwicklerteam bei der Veröffentlichung einer Anwendung auf verschiedenen Plattformen sein kann und ob das Framework es ermöglicht, die Anwendung auch im Browser laufen zu lassen. Außerdem soll die Anzahl an Codezeilen für alle Anwendungen verglichen werden, um eine Aussage zum Entwicklungsaufwand treffen zu können. Zuletzt wird die Unterstützung von Hardwarefeatures untersucht. Diese Analyse ist gerade auf der Android Plattform sinnvoll und gibt an, ob alle Frameworks den gleichen Funktionsumfang bei der Nutzung von nativen Features wie Kamera, Standortdiensten, Geräteinformationen und Sensorik bieten. Die Kombination aller dieser Metriken stellt einen umfangreichen Vergleich der Frameworks dar und abschließend sollen alle Punkte gewichtet und in eine Vergleichsmatrix eingetragen werden. So kann für jedes Framework eine abschließende Bewertungspunktzahl erstellt werden.

Erwartete Ergebnisse

Da .NET MAUI noch relativ neu ist wird gerade im Bereich der Performance ein Rückstand zu den etablierten Frameworks React Native und Flutter erwartet. Interessant ist hier zu analysieren, wie groß der Unterschied im Detail ist und ob sich dieser stark auf die Nutzererfahrung auswirkt.

Die Einrichtung von .NET MAUI gestaltet sich gerade für unerfahrene Entwickler möglicherweise wesentlich einfacher, da alle notwendigen Tools bei der Installation von Visual Studio direkt mit installiert und konfiguriert werden.

Die Konzepte von .NET MAUI sind gerade für unerfahrene Entwickler und Neueinsteiger komplizierter zu erlernen im Vergleich zu Flutter und React Native.

Alle drei Frameworks liefern ähnliche Ergebnisse bei der Nutzung von Systemressourcen (CPU, RAM) in den angefertigten Testszenarien.

Alle Frameworks haben die gleiche Anzahl an Features für die Plattformintegration. Hierzu zählen bspw. Der Zugriff auf Hardwarekomponenten wie Kamera und Sensoren wie auch der Zugriff auf diverse Features des zugrunde liegenden Betriebssystems.

Alle Frameworks liefern ähnliche Werte bei der Anzahl von Codezeilen für die Implementierung der Testapplikation auf, sodass allgemein gesagt werden kann, dass keines der Frameworks einen signifikanten Mehraufwand bei der Erstellung von Anwendungen aufweist.

React und Flutter weisen im Vergleich zu vorherigen Untersuchungen aufgrund ihrer stetigen Weiterentwicklung seit Veröffentlichung eine signifikante Verbesserung der Performance auf.

React und Flutter sind beide ungefähr gleich schnell zu erlernen, da sie ähnliche Konzepte verwenden.

Zeitplan

Dauer: Neun Wochen (10.04.2023 – 11.06.2023)

Woche 1 (10.04 – 16.04): Literaturrecherche, Literaturauswertung und Aufbau der Gliederung

Woche 2 (17.04 – 23.04): Konzepterarbeitung für die Testapplikation und Implementierung der Serverkomponente

Woche 3 (24.04 – 30.04): Einrichtung von .NET MAUI und Implementierung der Testapplikation

Woche 4 (01.05 – 07.05): Einrichtung von Flutter und Implementierung der Testapplikation

Woche 5 (08.05 – 14.05): Einrichtung von React Native und Implementierung der Testapplikation

Woche 6 (15.05 – 21.05): Auswahl von Profiling-Tools sowie Messung und Vergleich der geplanten Metriken

Woche 7 (22.05 – 28.05): Anfertigung von Diagrammen und Tabellen auf Basis der erhobenen Daten

Woche 8 (29.05 – 04.06): Anfertigung der Rohfassung der Arbeit auf Basis vorher angefertigter Fragmente

Woche 9 (05.06 – 11.06): Finalisierung der Arbeit und Abgabe

Literaturverzeichnis

- [1] "Comparison of the .NET MAUI and Xamarin.Forms Frameworks" - Lukáš Maňár, Tomas Bata University in Zlín, 2022, Von <http://digilib.k.utb.cz/handle/10563/51282>
- [2] "Development of Study Materials in Maui Framework for the Course Application Frameworks" - Radomír Húdek, Tomas Bata University in Zlín, 2022, Von <http://digilib.k.utb.cz/handle/10563/51090>
- [3] "Property Management Tool for Small Investors" - Jan Kurfürst, Tomas Bata University in Zlín, 2022, Von <http://digilib.k.utb.cz/handle/10563/51841>
- [4] „React Native vs Flutter, cross-plattform mobile application frameworks” – Wenhao Wu, Metropolia University of Applied Sciences, 2018, Von <https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf?sequence=1>
- [5] "Performance Comparison between React Native and Flutter" – Jakub Jagiello, Umeå University, 2019, Von <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1349917&dswid=3243>
- [6] "Choosing the right framework for Android development: which mobile development frameworks are chosen and why?" Seite 27 – Ashwin Banwarie, University of Amsterdam, 2022, Von https://ictinstitute.nl/wp-content/uploads/2022/08/Master_Thesis_A.Banwarie_FINAL.pdf
-