

Praxisprojekt

CrewRest

Janis Saritzoglou

Im Folgenden Praxisprojekt ist es Ziel eine Cross-Plattform Applikation für Windows 10, Android und iOS zu erstellen. Als Technologien soll Microsoft UWP und Xamarin zum Einsatz kommen. Somit soll aus einer gemeinsamen Codebasis eine Applikation, lauffähig auf verschiedenen Betriebssysteme erzeugt werden. Als Programmiersprache kommt C# zum Einsatz. Der fachliche Hintergrund ist es, ein System für die Mitarbeiter einer Luftfahrtgesellschaft zu entwickeln, um Urlaubsanträge oder allgemeiner: Anträge für Abwesenheiten erstellen zu können und dessen Status einsehen zu können. Bei Möglichkeit ist es erwünschenswert eine Art Kalenderübersicht in der Applikation anzuzeigen, welche die Abwesenheitslage aller Mitarbeiter darstellt, um mögliche Zeiträume für neue Abwesenheitsanträge sinnvoll anlegen zu können.

Inhaltsverzeichnis

1	Erstellung einer Cross-Plattform Applikation	4
2	Eingesetzte Hardware	5
3	Benötigte Software (APIs, Tools, IDEs, Emulatoren, Simulatoren)	6
4	Projektarchitektur	7
5	Wie funktioniert das native Deployment der unterschiedlichen Betriebssysteme	8
	Android	8
	Win10	8
	iOS	8
6	PCL	9
7	XAML vs. Code	10
8	Page Layouts	11
9	Implementierung	12
	SOAP	12
	Platformspezifisches Verhalten	12
	Bindings	12
	Updaten von Daten	12
	Anträge ListView	12
	Filtern der Liste	12
	Dynamisches hinzufügen von Komponenten	12
	Validierung von Eingaben	12
	DataTrigger	12
	EmbeddedResource	12
	Literatur	13

1 Erstellung einer Cross-Plattform Applikation

Als Entwicklungsumgebung für die Entwicklung der Apps für Android, iOS und Windows 10 kommt Visual Studio 2015 zum Einsatz. Alternativ kann auch Xamarin Studio eingesetzt werden. Bei der Erstellung eines neuen Cross-Plattform-Projekts, wird zwischen einer Blank App¹ Xamarin.Forms Portable und Xamarin.Forms Shared unterschieden (PETZOLD, 2016, S. 29).

¹Ein neues, leeres Projekt für mehrere Zielbetriebssysteme.

2 Eingesetzte Hardware

Zum Entwickeln und Testen der zu erstellenden Cross-Plattform-Applikation ist folgende Hardware zum Einsatz gekommen.

Tabelle 2.1: Hardware und Betriebssysteme

Gerät	OS	Version
iPhone 5	iOS	xxx
Asus Nexus 7	Android	xxx
Virtual Machine	Windows 10 Insider Preview	Build xxx
MacBook Pro	Mac OSX	xxx

3 Benötigte Software (APIs, Tools, IDEs, Emulatoren, Simulatoren)

Tabelle 3.1: Eingesetzte Software und APIs

Software	Art	Version	Zweck
Visual Studio 2015	IDE	2015 xxx	Entwicklung der Applikation
Android API	API	6.0 Marshmallow xxx	Benötigt für Deployment auf Android Platform
iOS API	API	xxx	Benötigt für Deployment auf iOS Platform



Abbildung 3.1: checkmark

4 Projektarchitektur

5 Wie funktioniert das native Deployment der unterschiedlichen Betriebssysteme

Android

Win10

iOS

6 PCL

7 XAML vs. Code

8 Page Layouts

9 Implementierung

```
public void lol(int i) {  
    private ImageSource bildquelle;  
}
```

SOAP

Platformspezifisches Verhalten

Bindings

Updaten von Daten

Anträge ListView

Filtern der Liste

Dynamisches hinzufügen von Komponenten

Validierung von Eingaben

DataTrigger

EmbeddedResource

Literatur

PETZOLD, CHARLES: *Creating mobile apps with xamarin.Forms: Cross-platform c# programming for iOS, android, and windows*. 1st Edition. Aufl. : Microsoft Press A Division of Microsoft Corporation One Microsoft Way Redmond, Washington 98052-6399, 2016
– Maßstab