# Entwicklung mobiler Applikationen

Die Entwicklung mobiler Applikationen lässt drei grundsätzlich verschiedene Herangehensweisen zu: die native, die webbasierte und die hybride Entwicklung. Jede Methode hat ihre Vor- und Nachteile, welche je nach Anforderungen mehr oder weniger stark gewichtet werden kann, sodass die „Best-Practice“ eine Einzelfallentscheidung darstellt.

Um die beste Herangehensweise für die Entwicklung der mobile SelfScanning-App zu ermitteln, werden in den folgenden Kapiteln zunächst die Anforderungen an die App selbst zusammengefasst und anschließend zusätzliche Anforderungen an die Entwicklungsmethode festgestellt. Davon ausgehend werden die drei Entwicklungskonzepte kurz vorgestellt und deren individuellen Vor- und Nachteile herausgestellt und gewichtet. Diese werden in einem abschließenden Fazit summiert, um die bestmögliche Entwicklungsmethode für das Projekt zu festzustellen.

## Ermittlung der Anforderungen

In der ersten Projektarbeit wurden bereits Anforderungen an ein mobile SelfScanning-System bei ALDI SÜD ermittelt. Diese werden im folgenden Kapitel „Fachliche Anforderungen“ nochmals erläutert und anschließend durch die technischen Anforderungen, die sich größtenteils aus den Rahmenbedingungen des Projekts ergeben, ergänzt. Beide Anforderungsgruppen werden abschließend in einer Liste stichwortartig zusammengefasst.

### Fachliche Anforderungen

Dem Kunden soll die Möglichkeit geboten werden, alle Artikel bereits während des Einkaufs in einer Filiale mithilfe seines Smartphones zu erfassen und in einer Liste zu speichern. Das Erfassen wird durch das Einscannen eines Barcodes auf der Artikelverpackung oder alternativ durch manuelles Suchen in einer Artikeldatenbank ermöglicht. Die Liste aller eingescannten Artikel wird bei der Bezahlung an das Kassensystem übertragen, sodass der Einkauf an einer herkömmlichen Kasse bezahlt werden kann. Das Auflegen der Artikel aufs Kassenband, anschließende Einscannen und erneute Einpacken an der Kasse entfällt somit.

Der virtuelle Warenkorb (sprich: die Liste auf dem Smartphone des Kunden, die alle Artikel enthält) soll die folgenden Einzelheiten anzeigen:

* Pro Artikel
  + EAN
  + Artikelbezeichnung
  + Verkaufspreis pro Stück
  + Stückzahl
  + Gesamtpreis
  + Mehrwertsteuersatz
  + ggf. Pfand das zu bezahlen ist
* Pro Einkauf
  + Filiale, in welcher der Einkauf stattfindet
  + Zeitpunkt, wann der Einkauf erstellt wurde
  + Anzahl der Artikel im Einkaufswagen
  + Gesamtsumme brutto
  + Gesamtsumme netto

Darüber hinaus gibt es folgende Anforderungen an die Artikel- und Preisinformationen, die dem Kunden angezeigt werden:

* Vollständigkeit

Es soll *ausnahmslos jeder Artikel erfasst werden können*. Hiervon nicht ausgenommen sind lose bzw. unverpackte Waren aus der O&G-Abteilung oder dem Backautomaten, Artikel deren Preis abhängig vom Gewicht ist (sog. Gewichtsartikel), sowie Pfandbons.  
Außerdem sollen auch die *jeweils in der Filiale gültigen Preise* angezeigt werden. Hiervon nicht ausgenommen sind Preise, die innerhalb der Filiale für einen bestimmten Zeitraum reduziert werden (z.B. Abverkauf von O&G kurz vor Feierabend).

* Aktualität der Daten

Die Artikel- und insbesondere die Preisinformationen, die dem Kunden angezeigt werden, sollen stets aktuell sein. Konkret heißt das, dass der Kunde immer *diejenigen Preise* angezeigt bekommt, welche *zum Zeitpunkt des Betretens der Filiale* gültig waren.

* Verfügbarkeit

Während des Einkaufs soll *weder eine Verbindung zum Internet noch eine Verbindung zum Kassensystem oder dem Filialnetzwerk erforderlich* sein.

In einigen Fällen führt das Erfassen von Artikeln mithilfe des Smartphones zu Ausnahmefällen. Diese werden im Folgenden erläutert:

* Um den Preis eines Gewichtsartikels bestimmen zu können, muss er zunächst gewogen werden. Da in der Filiale keine für den Kunden zugänglichen Waagen vorhanden sind, werden Gewichtsartikel nach wie vor an der Kasse gewogen. Hierfür soll, sobald sich ein Gewichtsartikel im Einkaufswagen des Kunden befinden, ein *Hinweis* erscheinen, mit der Bitte, *den Artikel an der Kasse zum Wiegen bereitzuhalten*.
* Ebenso soll beim Erfassen von *altersbeschränkten Artikeln ein Hinweis* *erscheinen*, dass die Volljährigkeit des Kunden an der Kasse zunächst kontrolliert werden muss.
* Die Bonsumme, die der Kunde auf dem Smartphone angezeigt bekommt, könnte von der Summe abweichen, die er tatsächlich zu zahlen hat. Dies könnte einer Preisveränderung während des Einkaufs oder einer internen Falschberechnung zu Schulden sein. In beiden Fällen soll die Differenz ermittelt und der Kunde darauf aufmerksam gemacht werden.

### Technische Anforderungen

Das Einscannen eines neuen Barcodes erfordert den Zugriff auf die Kamera des Smartphones. Die Zugriffsberechtigung soll einmalig bei der Erstinstallation eingefordert werden, sodass neue Artikel schnellstmöglich hinzugefügt werden können.

Zur Synchronisation von Artikel- und Preisinformationen wird eine Internetverbindung benötigt. Die initiale Datenbereitstellung soll aufgrund der Menge lediglich per WLAN erfolgen. Die nachfolgenden, regelmäßigen Aktualisierungen können per WLAN oder mobilem Internet erfolgen. Für beide Varianten soll die Zugriffsberechtigung einmalig bei der Erstinstallation eingefordert werden, sodass Daten ohne Zustimmung des Benutzers aktualisiert werden können.

Alle Daten sollen auch dann verfügbar sein, wenn keine Verbindung zum Internet besteht. Hierfür wird Zugriff zu einem permanenten Datenspeicher benötigt, dessen Berechtigung bei der Erstinstallation eingefordert werden soll.

Eine Datenbank, welche die Daten, sowie eine Schnittstelle zur Synchronisierung der Daten bereitstellt, ist vorhanden und ist ebenso wie zusätzliche Kassenfunktionen (z.B. Wiegen von Gewichtsartikeln oder Hinweis zur Alterskontrolle) nicht Teil dieses Projektes. Lediglich beim Entwurf und Implementierung der Schnittstelle muss darauf geachtet werden, die erforderlichen Daten bereitzustellen.

Da die mobile Applikation lediglich als Prototyp implementiert werden soll, spielen Anforderungen an Performance und User-Interface zunächst eine nebensächliche Rolle. Viel wichtiger ist der Aspekt, nach kurzer Zeit erste Entwicklungsergebnisse zu erhalten und die gewonnenen Erfahrungen, gemäß dem Evolutionären Vorgehensmodell, in die nächste Iteration der Entwurfsphase einfließen zu lassen.

### Zusammenfassung

Im vorherigen Kapitel wurden die Anforderungen an das Projekt nochmals näher beschrieben. Die folgende Liste gibt nochmals einen Überblick in Stichworten, um die Umsetzung der Anforderungen am Ende des Projekt überprüfen zu können.

**Fachliche Anforderungen**

* Einkauf mithilfe des Smartphones durchführen
  + Einkauf eröffnen
  + Artikel scannen / suchen
  + Einkauf abschließen / bezahlen
* Verfügbarkeit von Artikel- und Preisinformationen
  + Aktuell -> alle gültigen Preise bis zum Zeitpunkt des Betretens der Filiale vorhanden
  + Vollständig -> jeder Artikel soll erfasst werden können. Speziell O&G, Gewichtsartikel, Pfandbons und Backwaren aus dem Automat
  + Offline verfügbar -> voll funktionsfähig auch ohne WLAN und ohne mobiles Internet
* Einkauf soll an herkömmlicher Kasse bezahlt werden
* Ausnahmefälle
  + Gewichtsartikel -> Hinweis
  + Altersbegrenzte Artikel -> Hinweis
  + Summe der vom Kunden eingescannten Artikel stimmt mit tatsächlicher Summe der Artikel nicht überein -> wahrscheinlich Preisänderung während Einkauf -> „stiller Alarm“ zur Protokollierung und späterer Auswertung

**Technische Anforderungen**

* Uneingeschränkter Zugriff auf Gerätekomponenten
  + Kamera (Artikelscan)
  + Internetverbindung (Synchronisation)
  + Datenspeicher (Datenhaltung)
* Artikel- und Preisdatenbank ist vorhanden und bietet Schnittstelle zur Synchronisierung mit Smartphone
* Schnittstelle Smartphone -> Kasse um Einkauf zu bezahlen
  + Kennzeichnung von Gewichtsartikeln
  + Kennzeichnung von altersbeschränkten Artikeln
  + Kennzeichnung der Gesamtsumme, um mit tatsächlicher Summe abzugleichen
* Kurze Entwicklungszeit, steile Lernkurve

## Unterschiedliche Herangehensweisen zur App-Entwicklung

In den vorherigen Kapiteln wurden die Anforderungen an das Projekt und die mobile Applikation aufgezählt und erläutert. Davon ausgehend soll nun der geeignetste Ansatz zur Entwicklung einer App ermittelt und angewendet werden. Hierfür werden die drei Alternativen zunächst beschrieben und deren Vor- und Nachteile herausgestellt.

### Native App-Entwicklung

Als „native Apps“ werden eigenständige Anwendungen bezeichnet, die speziell für ein konkretes Betriebssystem implementiert werden und letztendlich nur auf diesem Betriebssystem installations- und lauffähig sind. Auf diese Weise können die Systemressourcen optimal ausgereizt und mithilfe entsprechender Grundkonzepte des jeweiligen Betriebssystems (Multithreading, Speicherzugriff, Grafikrendering) eine hohe Performance erzielt werden. Dieser Aspekt macht sich besonders bei komplexen Anwendungen bemerkbar. Darüber hinaus ermöglichen native Anwendungen den Zugriff auf Gerätekomponenten, wie z.B. Kamera, GPS-Sensor oder NFC-Chip.

Neben den eben genannten Vorteilen auf technischer Seite, gibt es noch einen weiteren, marketingtechnischen Aspekt: native Apps werden über den offiziellen App-Store des jeweiligen Herstellers vertrieben. Potenziellen Nutzer können die App also in gewohnter Art und Weise finden, Bewertungen lesen und die App gegebenenfalls direkt auf dem Smartphone installieren. Nach der Installation befindet sich eine Verknüpfung auf dem Homescreen des Nutzers, sodass die App jederzeit verfügbar steht.

Wobei dieser Aspekt gleichzeitig auch als Nachteil zu erwähnen gilt: vor der Veröffentlichung im App-Store muss die Anwendung zunächst ein aufwendiger Freigabeprozess durchlaufen werden. Die App kann jederzeit vom Store-Betreiber abgelehnt oder nachträglich aus dem Store entfernt werden. Außerdem ist für die offizielle Entwicklung einer nativen App im Falle von Apple eine kostenpflichtige Lizenz sowie ein Mac Voraussetzung.

Hinzu kommt die aufwendige Einarbeitung für unerfahrene Entwickler. Selbst bei guter Kenntnis der zu implementierenden Sprache ist eine sorgfältige Einarbeitung in das jeweilige SDK unverzichtbar.

* Eigenständige Anwendungen
* für bestimmtes OS entwickelt
* letztendlich nur auf diesem OS installations- und lauffähig
* => Ressourcen können optimal ausgenutzt werden
  + => hohe Performance möglich
  + => komplexe Applikationen möglich
* => Schnittstellen zur Hardware direkt zugänglich
* Für den Benutzer über den gewohnten Vertriebskanal „App-Store“ zugänglich
* Nach Installation Icon auf dem Homescreen -> schneller Zugriff
* Gebühren um native Apps offiziell entwickeln zu dürfen
* Aufwendiger Freigabeprozess durch den Store-Betreiber
* Einarbeitung in Herstellerspezifisches SDK sehr aufwändig

### Entwicklung einer Web-App

Eine Webanwendung wird im Gegensatz zur nativen App nicht eigenständig, sondern mithilfe eines Browsers auf dem Zielsystem ausgeführt. Sie kann durch manuelles Eingeben der URL oder Aufrufen eines Lesezeichens geöffnet werden. Im Regelfall wird keine Verknüpfung auf dem Homescreen erstellt. Eine Webanwendung unterscheidet sich also nur geringfügig von einer herkömmlichen Webseite. Die größten Unterschiede liegen in der für mobile Geräte optimierten Bedienoberfläche und dem reduzierten Funktions- und/oder Informationsumfang.

Der limitierende Faktor einer Web-App ist der jeweilige Browser, in dem die App ausgeführt wird. Funktionen, die innerhalb des Browsers nicht zur Verfügung stehen, können von der App nicht verwendet werden. So gibt es beispielsweise keine Möglichkeit, den Nutzer einer Webanwendung über anstehende Updates oder Erinnerungen per Push-Notification zu kontaktieren.

Dank moderner Webtechnologien wie HTML5 stehen allerdings zahlreiche Möglichkeiten für den Zugriff auf Systemkomponenten zur Verfügung. Leider muss der Anwender diesen Zugriff jedes Mal explizit gewähren. Darüber hinaus ermöglicht das sog. DOM-Storage eine permanente lokale Datenspeicherung bis zu 5 bis 10MB – je nach verwendetem Browser. Die zusätzliche Integration zahlreich vorhandener JavaScript-Frameworks ermöglicht dem Entwickler auch komplexe Applikationen schnell und einfach zu entwickeln.

Eine Webanwendung muss lediglich einmal implementiert werden und ist dank der einheitlichen Webstandards auf allen Betriebssystemen lauffähig. Zur Veröffentlichung ist weder ein Freigabeprozess noch eine Lizenz notwendig. Es genügt ein Webserver auf dem die Anwendung läuft.

### Hybride App-Entwicklung

Hybride Apps vereinen die Eigenheiten beider Entwicklungsalternativen: sie werden mithilfe von HTML, CSS und JavaScript implementiert und anschließend in eine native App integriert. Die native App besitzt als einzige Komponente einen InApp-Browser. In diesem eingebetteten Browser wird die ursprünglich entwickelte Webanwendung gerendert.

Auf diese Weise werden die Vorteile beider Welten kombiniert: mithilfe der herkömmlichen Webtechnologien lassen sich Anwendungen relativ unkompliziert entwickeln, sind auf verschiedenen Betriebssystemen lauffähig und bieten dank der „Wrapper-App“ trotzdem den vollen Funktionsumfang einer nativen App. Für den Nutzer sind hybride Apps von nativen kaum zu unterscheiden, da sie wie gewohnt über den App-Store heruntergeladen, installiert und über ein Icon aufgerufen werden. Lediglich in den Bedienelementen lässt sich gegebenenfalls ein Unterschied zu nativen Apps erkennen.

## Fazit

Im ersten Teil dieses Kapitels wurden nochmals die Anforderungen an die App beschrieben. Dabei wurde ersichtlich, dass besonders die Kommunikation mit externen Systemen (Artikel- und Preisdatenbank sowie Kassensystem in der Filiale), sowie die Speicherung großer Datenmengen (Offline-Verfügbarkeit aller Daten) zwei wichtige Anforderungen darstellen.

Der anschließende Vergleich der drei Herangehensweisen verdeutlichte, dass die Wahl der Entwicklungsmethode weitreichende Auswirkungen auf die Implementierung und dessen Ergebnis haben kann. So bieten native Apps selbst bei komplexen Anwendungen eine hohe Performance, der nötige Entwicklungsaufwand ist besonders für unerfahrene Entwickler jedoch äußerst hoch. Web-Apps lassen sich dagegen innerhalb kürzester Zeit mithilfe gängiger Webtechnologien entwickeln und ohne weitere Umstände (weder Lizenz noch Freigabeprozess) veröffentlichen. Dank zahlreich vorhandener Frameworks lassen sich auch komplexere Anwendungen umsetzen. Der Funktionsumfang des Browsers begrenzt den Funktionsumfang jedoch deutlich, sodass eine Web-App für dieses Projekt nicht in Frage kommt (lediglich 5-10MB Datenspeicher, keine Echtzeit-Updates über Push-Notifications). Die hybride Variante scheint die Vorteile beider Alternativen zu vereinen.

Langfristig wäre aufgrund der hohen Datenmenge und der damit einhergehenden Rechenlast eine native Entwicklung sicherlich die optimale Lösung für dieses Projekt. Vor dem Hintergrund des prototypischen Charakters und der zeitlichen Begrenzung wird die hybride App-Entwicklung der nativen jedoch vorgezogen.