

# Digitale Kataloge als App

## Funktionalität

Die geplante Applikation soll ein ähnliches Aussehen und Auftreten haben wie der Apple eigene „iBooks-Store“. Es soll beim Aufrufen der App ein simpler Background angezeigt werden. Auf diesen Background kann man interaktive Kataloge der verschiedensten Anbieter laden (bspw. Elektronik, Mode, Möbelartikel usw.). Um die Kataloge downloaden/listen zu können befinden sich im Footer Bereich mehrere Buttons, die eine Vorauswahl und Kategorisierung der Kataloge übernehmen sollen.

In unserem Fall sind es „Mehr“, „Suche“, „Kategorien“.

In den gelisteten (HTML)-Katalogen kann man blättern und sich die verschiedensten Artikel ansehen. Darüber hinaus bei Gefallen auf eine Produkt-ID tippen, was eine Weiterleitung zum entsprechenden Shop mit dem passenden Produkt bewirkt.

Man hat auch die Möglichkeit ein Lesezeichen in dem Katalog zu setzen (auswählbar durch den Lesezeichen Button beim Aufrufen des Katalogs) um in dem gewünschten Katalog das entsprechende Produkt speichern und erneut ansehen zu können.

Der Footer Bereich der App ist in verschiedene Buttons aufgeteilt:

Mehr, Lesezeichen, Home, Suche, Kategorien.

Mehr: Beim klicken des Buttons wird eine neue Seite aufgerufen auf der – unter anderem – häufig angesehene Kataloge von anderen Nutzern angezeigt werden sollen.

Lesezeichen: Beim Aufrufen dieser Seite, werden die einzelnen Kataloge die man zuvor als Lesezeichen gespeichert hat angezeigt. Man kann die zuvor gespeicherten Produkte durch einen Klick aufrufen.

Home: Man wird zurück in die Übersicht geleitet, wo alle gelisteten Kataloge angezeigt werden.

Suche: Eine Suchfunktion, die es ermöglichen soll nach Herstellern und einzelnen Katalogtiteln zu suchen und sich die Ergebnisse anzeigen zu lassen.

Kategorien: Eine Auswahl an Kategorien soll es vereinfachen nach bestimmten Exemplaren filtern zu können. Beim Klick auf eine bestimmte Kategorie werden die darin zugeordneten Kataloge angezeigt.

Mit der Applikation wird hauptsächlich das Problem der intensiven Einzelsuche nach Artikeln, Katalogen oder Herstellern im Netz vereinfacht. Oftmals findet man keine vergleichbaren Produkte oder Angebote im Netz. Die Kategorisierung der vielen Kataloge soll dies ebenfalls lösen.

Kosteneinsparung gegenüber von Print Produkten.

In dem Fach „Corporate Publishing“ arbeiten wir derzeit an einem interaktiven Magazin für das iPad, das verschiedenste Artikel und Dienstleistungen bereitstellt. Eine Sammlung und Kategorisierung solcher Magazine und Kataloge in einer Applikation vereinfacht die Produktsuche.

Die Zielgruppe soll sich weitreichend über alle Altersgruppen erstrecken, die ein iPad bedienen können und Interesse am Blättern in Katalogen zeigen. Die Zielgruppe soll dadurch zum Lesen und zum Kauf von Produkten angeregt werden.

Mögliches Alter: 16-99 Jahre.

## Testing

Die im Wireframe enthaltene Suchfunktion soll bei den meisten Befragten tiefer in das System eingreifen und auch nach Fachwörtern, die in Artikeln von Magazinen enthalten sind suchen und diese anzeigen können.

Ebenfalls gewünscht waren Funktionen, die es ermöglichen die (HTML)-Magazine als .pdf am Rechner oder in Cloud Diensten sicher zu können.

Eine Notizfunktion soll enthalten sein, die neben den gewünschten Katalogen beispielsweise ein gemerktes Produkt notieren soll.

## A/B Test

-> Daraus ergibt sich für einen A/B Test die Möglichkeit in Zukunft unter dem Menüpunkt

„Einstellungen“ ein Anmeldefenster für die Cloud Dienste zur Verfügung stellen und

-> die Suchfunktion zu erweitern

-> Eine Implementierung von einfachen „Zurück Buttons“ sollte vorgenommen werden.

Mögliche Schwachpunkte in der UX würden sich negativ auf die Anzahl der Aufrufe der WebApp auswirken und negative Bewertungsberichte hervorrufen.

Zudem könnte der Umsatz zurückgehen.

## Key Performance Indikatoren

- Anzahl der Aufrufe der WebApp

- Anzahl der positiven Berichte über die App (bspw. im Entwicklerforum)

- Umsatz (pro Tag/Monat/Jahr), falls eine erweiterte App kostenpflichtig sein soll

- Antwortzeiten und Stabilität der WebApp

## Wire Frames (Beispiel)

