

Inhaltsverzeichnis

1	Einleitung	2
2	Windows 8	3
2.1	Neues Bedienkonzept	3
2.2	Windows 8 als hybrides System	3
2.3	Windows RT	3
2.4	Das Ökosystem	3
2.4.1	Microsoft Store	3
2.4.2	xBox	3
2.4.3	Windows Phone	3
3	Konzeption der App	4
3.1	Ziel der App	4
3.2	Nativ vs. Web	5
3.2.1	Native App	5
3.2.2	Webapp	6
3.2.3	Windows 8 App mit Webtechnologien	6
3.3	Navigationskonzept	6
3.3.1	Flaches System	6
3.3.2	Hierarchisches System	7
3.4	Menü und Features	8
4	App Entwicklung	9
4.1	Datenanbindung	9
4.2	Ressort Ansichten	10
4.3	Wichtiger Code	10
5	Evaluierung	11
5.1	Relation Bilder - Artikelinhalt	11
6	Fazit	12

1 Einleitung

2 Windows 8

Seit dem 26. Oktober 2012 ist Microsofts aktuellstes Betriebssystem „Windows 8“ für die breite Öffentlichkeit verfügbar. Dieses unterscheidet sich grundlegend von seinem Vorgänger „Windows 7“. Microsoft verfolgt mit Windows 8 den Ansatz, das gleiche Betriebssystem sowohl für Desktop- als auch für Tablet-Computer¹ zu verwenden. Es besitzt nach wie vor den bekannten Desktop von Windows 7. Äußerlich sind hier nur kleine Änderungen vorgenommen worden. So besitzt z.B. der Windows Explorer eine neues, kontextsensitives Ribbon-Menü². Neu ist hingegen das "Modern User Interface"(Modern UI, früher Metro), eine für Touch-Gesten optimierte Oberfläche. Obwohl die neue Oberfläche für Touch-Gesten optimiert ist, kann sie trotzdem auch mit Maus und Tastatur bedient werden. Es können, im Gegensatz zum Desktop, keine herkömmlichen Programme im Modern UI ausgeführt werden. In der Modern UI Oberfläche laufen nur Programme (Apps), die speziell für Windows 8 entwickelt wurden und über den Microsoft Store (siehe Sektion 2.4.1 auf Seite 3) erhältlich sind. Es soll nun ein etwas detaillierterer Blick auf die neue Oberfläche und dessen Eigenheiten geworfen werden.

2.1 Neues Bedienkonzept

2.2 Windows 8 als hybrides System

Windows 8 kann sowohl auf herkömmlichen Desktop Computern als auch auf Tablet-Computern eingesetzt werden.

2.3 Windows RT

2.4 Das Ökosystem

2.4.1 Microsoft Store

2.4.2 xBox

2.4.3 Windows Phone

¹Tablet-Computer: kleiner, flacher, tragbarer Computer mit einem Touchscreen. Es wird von nun an, der Einfachheit halber die englische Version „Tablet“ benutzt.

²Auch bekannt als Menüband oder Multifunktionsleiste.

3 Konzeption der App

In diesem Kapitel geht es darum, die Konzeption für eine Windows 8 Nachrichten App darzustellen und zu erläutern. Dabei gilt es darzulegen, warum und mit welchem Hintergrund Entscheidungen zu Gunsten der einen oder der anderen Möglichkeit ausfallen. Dazu müssen zunächst die Ziele der App ausgeführt werden. Anschließend muss sich entschieden werden mit welcher Technologie bzw. Programmiersprache gearbeitet werden soll. Zuletzt wird noch ein detaillierterer Blick auf das Herzstück der App geworfen. Das Navigationskonzept. Hierbei ist eine der wichtigsten Fragen: „Wie kann ich auch bei ggf. großen Datenmengen eine übersichtliche, intuitive Struktur schaffen die sich in das Gesamtkonzept von Windows 8 eingliedert?“

3.1 Ziel der App

Das Ziel der zu erstellenden App ist, den Inhalt der ZEIT ONLINE Website auf ansprechende Art und Weise in einer Windows 8 Applikation darzustellen. Das Layout und die Darstellung der Inhalte soll sich an das sogenannte „Look and Feel“ von Windows 8 anpassen und sich daran orientieren. Der Fokus bei den Inhalten liegt auf den Artikeln selbst und den jeweiligen Aufmacher bzw. Teaser Bildern. Das heißt andere Inhalte wie z.B. Bildergalerien, Infografiken, Blogartikel oder Quizze werden von der App nicht erfasst und nicht dargestellt. Hintergrund ist, die App möglichst einfach zu halten da es in der Fragestellung um die Relation zwischen den Aufmacherbildern und dem dahinter liegenden Artikeltext geht. Die App erhebt in dieser Hinsicht keinen Anspruch darauf, die gesamten redaktionellen Inhalte von ZEIT ONLINE darzustellen, sondern versteht sich eher als explorative Applikation im Sinne der Fragestellung.

Der User soll die Möglichkeit haben die standardmäßig vorhandenen Artikeltitel auszublenken um so, wenn gewünscht, einen rein visuellen Eindruck der Artikelbilder zu bekommen. Hierzu soll einen Schalter in der Menüleiste am unteren Bildrand geben. Wenn der User sich im Artikel befindet soll er die Möglichkeit haben die Schriftgröße in gewissen Maß selbst zu bestimmen, da die App eventuell auf Monitoren mit verschieden großen Auflösungen ausgeführt werden wird oder die Sehkraft des Users nicht mehr ausreicht um eine normal große Schrift zu erkennen und zu lesen. So wird in den Artikeln ein gewisses Maß an Barrierefreiheit gewährleistet.

Das übergeordnete Ziel der App ist es, eine rudimentäre Nachrichten Applikation für Windows 8 zu erstellen. Gleichzeitig soll eine Umgebung geschaffen werden, die es erlaubt Untersuchungen anzustellen, in wie weit es möglich ist allein durch das Betrachten der Aufmacherbilder auf den Inhalt der jeweiligen Artikel zu schließen (siehe Sektion 5.1 auf Seite 11). Außerdem soll die App dazu dienen, zu erforschen, wie eine Nachrichten Applikation in der Modern UI Oberfläche von Windows 8 erstellt wird und welche design- als auch funktionstechnischen Vorgaben von Microsofts vorhan-

den sind. Sprich, wie eine App mit Nachrichteninhalten nach Microsofts Sichtweise auszusehen hat.

3.2 Nativ vs. Web

Vor dem Entwickeln einer Windows 8 App muss sich entschieden werden mit welcher Technologie bzw. mit welcher Programmiersprache entwickelt werden soll. Die Rede ist hier von einer App die in der Modern UI Oberfläche von Windows 8 läuft und für diese konzipiert ist. Das heißt es handelt sich nicht um eine klassische .NET oder WIN32 Anwendung für Windows. Um eine Modern UI App zu entwickeln, müssen zwei Dinge zwingend vorhanden sein. Zum einen Windows 8 selbst und zum anderen wird die neueste Version von Visual Studio, Visual Studio 11³, benötigt. Visual Studio steht in der Express Version für Windows 8 kostenlos zur Verfügung. Des weiteren muss sich zwischen der nativen Umsetzung und der Implementierung mit Webtechnologien entschieden werden. Es soll zunächst erläutert werden was die beiden Begriffe bedeuten und in welcher Weise und welchem Zusammenhang sie bei der Entwicklung einer Windows 8 Modern UI App üblicherweise gebraucht werden.

3.2.1 Native App

Der heutige Begriff „native App“ unterscheidet sich in einigen Punkten von der früheren oder ursprünglichen Verwendung des Begriffs. Früher sprach man von einer nativen App wenn direkt auf die Ressourcen der Maschine zugegriffen wurde, wie z.B. Maschinencode der direkt von der CPU ausgeführt wird. Heute wird eine App oftmals schon als nativ deklariert, wenn es sich nicht um eine Webapp handelt. Eine sinnige Definition liegt irgendwo zwischen diesen beiden Varianten. Eine App ist dann nativ, wenn sie Geräte-, Betriebssystem- oder Laufzeitumgebungsabhängig ist. Das heisst, sie ist für ein spezielles Gerät entwickelt und kann nur auf diesem ausgeführt werden. Sie kann dabei alle Geräte- oder Betriebssystemspezifischen Funktionen nutzen, es ist jedoch egal wie nah an der Hardware tatsächlich programmiert wurde (O'Brian, 2013).

In Visual Studio können native Windows 8 Apps u.a. mit den Programmiersprachen C++, C# oder Visual Basic erstellt werden. Für das Design bzw. das Aussehen der App wird die Markupsprache - Extensible Application Markup Language (XAML) - verwendet. Es gibt zwei Möglichkeiten wie das XAML erstellt werden kann. Es kann entweder von Hand geschrieben werden oder man lässt es sich automatisch generieren. Zum automatischen Generieren lassen sich per Drag & Drop Elemente wie z.B. Buttons und andere Schaltflächen aus einer Werkzeugpalette direkt auf die „App-Leinwand“ ziehen, sowie verschieben oder nach Belieben anordnen.

³Visual Studio 2011 war die aktuellste Version beim Erstellen dieser Arbeit.

3.2.2 Webapp

3.2.3 Windows 8 App mit Webtechnologien

3.3 Navigationskonzept

Damit der User ein für ihn angenehmes und flüssiges Nutzungserlebnis hat, ist es notwendig sich über das Navigationskonzept Gedanken zu machen, gerade wenn es sich um eine App handelt, in der es viele verschiedene Inhaltsbereiche gibt. Der User soll sich möglichst intuitiv durch die Inhalte bewegen können. Außerdem muss dem User auf der Einstiegsebene der ein guter Überblick über die vorhandenen Inhalte gegeben werden, damit er von dort aus zielgerichtet weiter navigieren kann, ohne lange suchen zu müssen. Microsoft nennt in seinen Richtlinien für Windows Store Apps (Modern UI Apps) grundsätzlich zwei Möglichkeiten wie die Navigation umgesetzt werden kann.

3.3.1 Flaches System

Das flache Navigationssystem wird in vielen Windows Store Apps verwendet, häufig in Spielen, Browsern oder in Apps zum Erstellen von Dokumenten. Es zeichnet sich dadurch aus, dass sich die Inhalte auf der gleichen hierarchischen Ebene befinden. Dieses System eignet sich wenn ein schneller Wechsel zwischen wenigen Seiten oder Registerkarten möglich sein soll (Microsoft, 2013).



Abbildung 1: Flaches Navigationssystem (Microsoft, 2013)

Abbildung 1 zeigt wie das flache Navigationssystem funktioniert. Am oberen Rand befindet sich eine, nicht immer sichtbare Navigationsleiste mit den verschiedenen Registerkarten oder Seiten. Die Leiste wird angezeigt, wenn der User vom unteren oder

oberen Bildrand streift (siehe Sektion 2.1). Wenn der User die Seite wechseln möchte geschieht dies direkt über die Navigationsleiste, d.h. es gibt meist keinen permanenten Zurück-Button. Ein flaches Navigationssystem eignet sich nicht unbedingt für eine Nachrichten App wie die, die in dieser Arbeit konzipiert und umgesetzt ist, weil es zu viele Bereiche (Ressorts) gibt, welche sich zudem sehr ähneln. Hier ist es angebracht ein hierarchisches Navigationssystem zu verwenden.

3.3.2 Hierarchisches System

Die meisten Windows Store-App benutzen ein hierarchisches Navigationssystem. Microsoft nennt es Hubnavigationsmuster. Es eignet sich um große Inhaltssammlungen zu ordnen und sie für User benutzerfreundlich auf zu bereiten. Der Schlüssel dazu ist die Unterteilung des Inhalts in verschiedene Detailebenen (Microsoft, 2013).

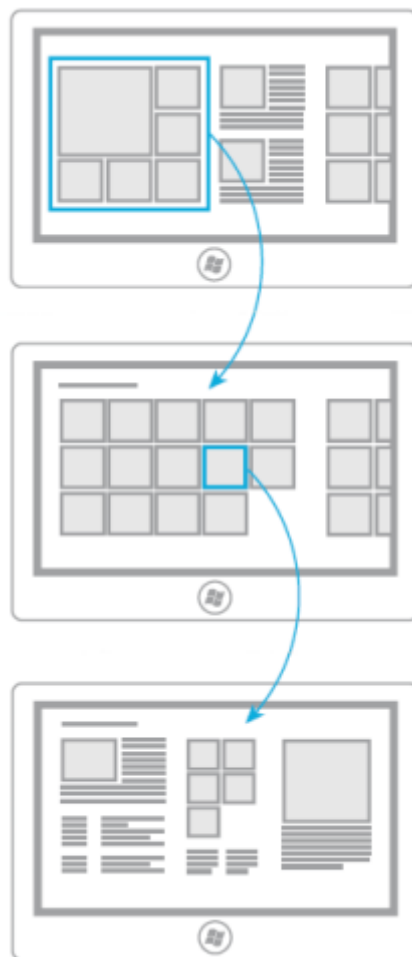


Abbildung 2: Hierarchisches Navigationssystem (Microsoft, 2013)

Das Schema eines hierarchischen Systems ist in Abbildung 2 dargestellt. Der Einstiegspunkt in die App ist die sogenannte „Hubseite“ hier ganz oben in der Abbildung zu sehen. Auf der Hubebene werden aus den vielen großen Bereichen der App jeweils ein kleiner Teil gesammelt und dargestellt, sodass sich der User vorstellen

kann was ihn im jeweiligen Bereich erwartet. Die Anordnung, in welcher der Auszug der Inhalte dargestellt ist muss nicht für jeden Bereich gleich sein. Es kann das Nutzungserlebnis sogar verbessern und vielfältiger machen wenn unterschiedliche Darstellungen (z.B. in Höhe, Breite oder Anzahl der Objekte) für die Bereiche genutzt werden. Die Hubansicht wird horizontal gescrollt, d.h. weiterer Inhalt befindet sich hinter dem rechten Rand und kann entweder durch das Scrollrad der Maus oder auf einem Tablet, durch das swipen nach links in das Sichtfeld des Users gebracht werden.

Die mittlere Darstellung in Abbildung 2 zeigt die zweite Detailstufe des hierarchischen Systems. Hier werden alle Elemente eines Bereichs dargestellt. In diesem Fall werden hier alle Artikel aus einem Ressort aufgelistet.

Im unteren Bereich der Abbildung ist schließlich die letzte Detailstufe zu sehen. Es handelt sich hierbei um den eigentlichen Inhalt (z.B. einen Artikel). Es ist ebenfalls möglich von der Hubansicht direkt in die Detailansicht eines Elements zu wechseln.

3.4 Menü und Features

4 Umsetzung der App

4.1 Datenanbindung

Es soll zunächst geklärt werden wie sich der Inhalt der ZEIT ONLINE Website darstellt und wie er generiert wird. Die redaktionellen Inhalte werden fast ausschließlich über ein hauseigenes Content Management System (CMS) erstellt und gepflegt. Das CMS generiert aus den Inhalten eine Extensible Markup Language (XML)-Struktur. Aus dem XML wiederum wird anschließend mit Hilfe von Extensible Stylesheet Language Transformations (XSLT) in einer zweistufigen Transformation das fertige Hypertext Markup Language (HTML) erstellt. Dies ist ein grober Überblick wie sich die Webseite von ZEIT ONLINE zusammensetzt.

Für die Windows 8 App wird direkt das XML verwendet, welches vom CMS generiert wird. Für die App werden zwei verschiedene Seitentypen der Webseite benötigt um die gewünschten Informationen darzustellen. Zum einen die verschiedenen „Centerpages“ und zum anderen die Artikelansicht. Centerpages sind die Hauptseiten der jeweiligen Ressorts (z.B. die Hauptseite des Politikressorts), sowie die Startseite mit den wichtigsten und aktuellsten Meldungen. Auf den Centerpages befinden sich die Teaserelemente für die verschiedenen Artikel. Die Teaserelemente bestehen meist aus einem Bild und einem kurzen, prägnanten Text, welcher kurz erläutert worum es in dem dahinter liegenden Artikel geht.

```
1 <block href="http://xml.zeit.de/digital/internet/2013-08/fablab-open-
   hardware" year="2013" issue="34" ressort="Digital" author="Tilman Baumgä
   rtel" contenttype="article" publication-date="" expires="" date-last-
   modified="2013-08-14T12:58:40+00:00" date-first-released="2013-08-14T09
   :57:43.627551+00:00" date-last-published="2013-08-14T12
   :59:39.691370+00:00" last-semantic-change="2013-08-14T09
   :56:40.185797+00:00">
2 <supertitle>Open Hardware</supertitle>
3 <title>Fab Labs, die Maschinen-Bibliotheken</title>
4 <text>
5     3-D-Drucker, CNC-Fräsen oder Lasercutter - mit solchen Maschinen
       sollen Bastler in Fab Labs experimentieren. Immer mehr solcher
       Werkstätten entstehen nun in aller Welt.
6 </text>
7 <description>
8     3-D-Drucker, CNC-Fräsen oder Lasercutter - mit solchen Maschinen
       sollen Bastler in Fab Labs experimentieren. Immer mehr solcher
       Werkstätten entstehen nun in aller Welt.
9 </description>
10 <byline/>
11 <image alt="MakerBot Replicator 2" align="left" title="MakerBot
    Replicator 2" base-id="http://xml.zeit.de/digital/internet/2013-08/
    makerbot-cebit-hannover/" type="jpg" publication-date="" expires="">
```

```

12     <bu>
13         Ein MakerBot Replicator 2 auf der diesjährigen Cebit in Hannover
14     </bu>
15     <copyright>© REUTERS/Fabrizio Bensch</copyright>
16 </image>
17 </block>

```

Listing 1: Das XML eines Teaserelements

Das XML von einem Teaserelement ist in Listing 1 dargestellt. Es werden jedoch nicht alle Informationen aus dem XML verwendet. Die verwendeten Informationen sind das „date-first-released“ (Zeile 1), der „title“ (Zeile 3), sowie die „description“ (Zeile 7) und das „image“ (Zeile 11). Das XML für die Artikelansicht ist ähnlich aufgebaut, nur ist hier zusätzlich noch der Artikeltext (Inhalt) in Form von Paragraphen-Tags (<p>) enthalten. Dies ist ein konkreter Einblick wie die rohen Daten aussehen, welche für die App verwendet werden. Wie die Daten im Detail verarbeitet werden wird in Sektion 4.3 auf Seite 10 näher erläutert.

4.2 Ressort Ansichten

Wenn die App startet befindet sich der User in der Startansicht. Hier werden ihm für alle Hauptressorts die ersten sechs Artikel angezeigt, in der Reihenfolge wie sie auch auf der Webseite zu finden sind.

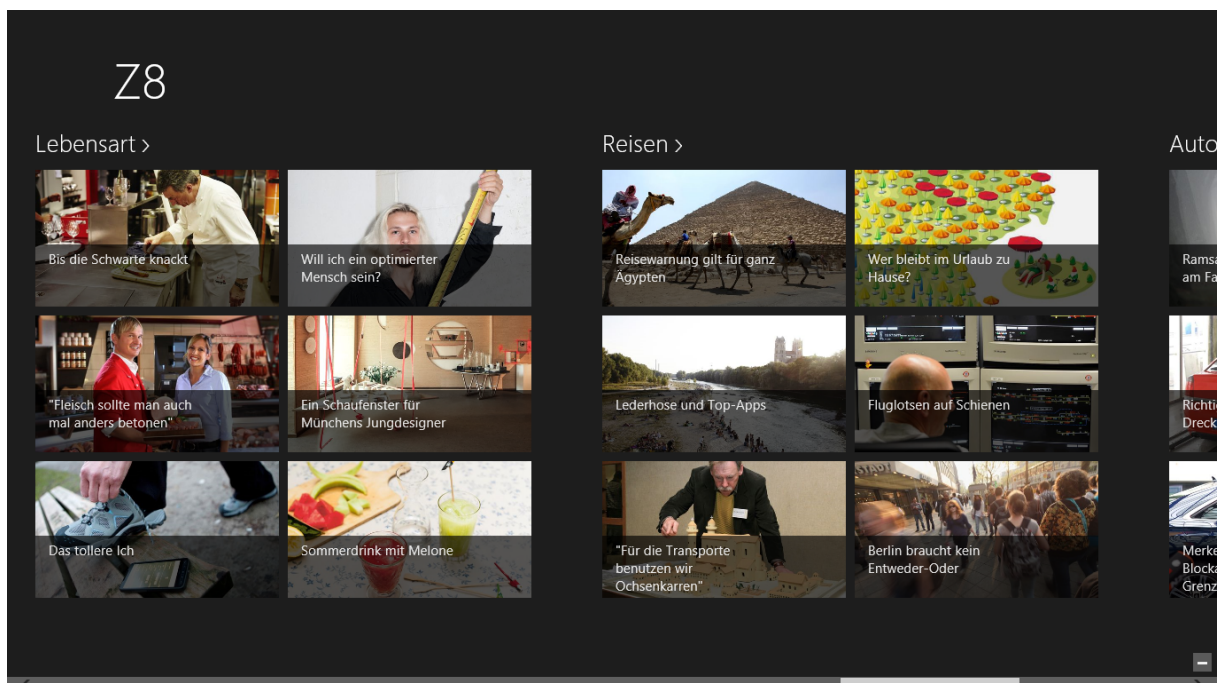


Abbildung 3: Ressortübersicht in Startansicht

4.3 Wichtiger Code

5 Evaluierung

5.1 Relation Bilder - Artikelinhalt

6 Fazit

Abkürzungsverzeichnis

XAML	Extensible Application Markup Language	5
CMS	Content Management System	9
XML	Extensible Markup Language.....	9
XSLT	Extensible Stylesheet Language Transformations	9
HTML	Hypertext Markup Language.....	9

Literatur

[Microsoft 2013] MICROSOFT: *Navigationsdesign für Windows Store-Apps*. <http://msdn.microsoft.com/de-de/library/windows/apps/hh761500.aspx>.
Version: 2013. – zuletzt geprüft: 17.08.2013

[O'Brian 2013] O'BRIAN, Tim: *Web, native, and déjà vu*. <http://channel9.msdn.com/Blogs/Vector/Web-native-and-dj-vu>. Version: Januar 2013. – zuletzt geprüft: 29.07.2013

Abbildungsverzeichnis

1	Flaches Navigationssystem (Microsoft, 2013)	6
2	Hierarchisches Navigationssystem (Microsoft, 2013)	7
3	Ressortübersicht in Startansicht	10