



Zweite Bachelorarbeit

# „Technische und anwendungsbezogene Probleme und Besonderheiten des mobilen Internets“

Ausgeführt am Fachhochschul-Bachelorstudiengang „Medientechnik“  
an der Fachhochschule St. Pölten

unter der Leitung von

FH-Prof. Dipl.-Ing. Grischa Schmiedl

ausgeführt von

Kerstin Blumenstein  
mto71008

St. Pölten am

## Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Bachelorarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Bachelorarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.
- diese Arbeit mit der vom Begutachter/von der Begutachterin beurteilten Arbeit übereinstimmt.

Ich räume hiermit der Fachhochschule St. Pölten das ausschließliche und räumlich unbeschränkte Werknutzungsrecht für alle Nutzungsarten an dieser Bachelorarbeit ein, und behalte das Recht, als Urheber dieses Werkes genannt zu werden.

---

Ort, Datum

---

Unterschrift

## Kurzfassung

Laut Statistik Austria besaßen bereits 2008 92,5 % der österreichischen Haushalte ein Mobiltelefon. Davon wurden - bei steigender Tendenz - 48 % der Geräte als prinzipiell „internet-tauglich“ eingestuft. Parallel dazu werden speziell in Österreich die Tarife zur Benutzung des mobilen Internets immer kostengünstiger. Diese Fakten zeigen deutlich, dass mit dem mobilen Internet immer mehr Menschen zu erreichen sind, und es deshalb ein relevantes Thema im Bereich der interaktiven Medien darstellt. In diesem Zusammenhang ist es bei der Erstellung mobiler Webseiten wichtig zu betrachten, welche Probleme bei der Nutzung des Internets mit einem mobilen Endgerät für den User entstehen und wie diese Probleme zu umgehen bzw. zu beheben sind. Diese Fragen liegen dieser Arbeit als Forschungsfragen zu Grunde.

Es werden dabei vier Problemfelder genauer betrachtet und auf einer Auswahl an aktuellen Smartphones analysiert: Das Eingabeproblem, das Darstellungsproblem, das Vollständigkeitsproblem und das Limitationsproblem. Es wird gezeigt, dass Applikationen und Websites für die mobile Nutzung den im Folgenden aufgeführten Regeln folgen sollten:

- reduzierte Userinteraktion über die Tastatur
- günstig gestaltete Bedienelemente
- Layout auf die am Gerät vorhandenen Bildschirme abstimmen
- Inhalte an die mobilen Belange anpassen
- Nutzung von Technologien und Frameworks kritisch abwägen

## Inhaltsverzeichnis

Ehrenwörtliche Erklärung .....	II
Kurzfassung .....	III
Inhaltsverzeichnis .....	IV
1. Die Verbreitung des mobilen Internets.....	1
1.1. Gegenstand und Ziel der Arbeit.....	1
1.2. Aufbau der Arbeit .....	2
2. Das mobile Internet und die mobilen Geräte .....	3
2.1. Das mobile Endgerät.....	3
2.1.1. Mobiltelefone.....	3
2.1.2. Feature Phones.....	3
2.1.3. Smartphones.....	4
2.1.4. Non-Phone Geräte.....	4
2.2. Das mobile Internet .....	5
2.2.1. Internetzugriff via Browser.....	5
2.2.1.1. Desktop Web.....	5
2.2.1.2. Mobiles Web .....	5
2.2.1.3. Desktop Web - mobile ready .....	6
2.2.2. Internetzugriff via Applikation .....	6
2.3. Die Testgeräte.....	7
3. Probleme bei der Verwendung des mobilen Internets .....	9
3.1 Das Eingabeproblem .....	9
3.1.1. Das Problem im Detail.....	9
3.1.2. Das Problem in der Praxis.....	10
3.1.3. Best Practices.....	11
3.2. Das Darstellungsproblem.....	15
3.2.1. Das Problem im Detail.....	15
3.2.2. Das Problem in der Praxis.....	16
3.2.3. Best Practices.....	18
3.3. Das Vollständigkeitsproblem .....	20
3.3.1. Das Problem im Detail.....	20
3.3.2. Das Problem in der Praxis.....	21
3.3.3. Best Practices.....	23
3.4. Das Limitationsproblem.....	25
3.4.1. Das Problem im Detail.....	25
3.4.2. Das Problem in der Praxis.....	27
3.4.3. Best Practices.....	28
4. Lösungsansätze in der Praxis .....	30
4.1. Der Fotoupload.....	30
4.2. Ein Video einbinden .....	31
4.3. Der Login .....	32
5. Die zukünftige Bedeutung des mobilen Internets.....	36
5.1. Ergebnisse der Arbeit.....	36
5.2. Das mobile Internet in der Zukunft.....	36
Verzeichnisse .....	38
Quellen.....	38
Abbildungen.....	43
Tabellen.....	43
Listings.....	43

### 1. Die Verbreitung des mobilen Internets

Die Verbreitung von internetfähigen Mobiltelefonen hat in letzten Jahren zugenommen. Mittlerweile benutzen die Menschen weltweit mehr internetfähige mobile Endgeräte als Desktop-Rechner (vgl. Bieh 2008, 159f).

Nach Statistik Austria besaßen bereits 2008 92,5 % der österreichischen Haushalte ein Mobiltelefon. Davon wurden 48 % der Geräte als prinzipiell „internettauglich“ eingestuft. Wobei als internetfähig in dieser Studie alle Geräte mit WAP, GRPS und UMTS galten (vgl. STATISTIK AUSTRIA - IKT-Einsatz in Haushalten 2008, 43). Diese Tendenz ist stark steigend, da in der Regel Kunden alle zwei Jahre ein neues aktuelles Gerät kaufen.

Gleichzeitig werden speziell in Österreich die Tarife zur Nutzung des mobilen Internets immer kostengünstiger. Als ein Beispiel sei der Provider Bob angeführt, der das 1 GB Datenpaket für 4 Euro im Monat anbietet (vgl. bob datenpaket 1gb 2010).

Zudem veröffentlichte der Analyst Gartner die Top10 der mobilen Technologien, auf die 2010 und 2011 geachtet werden sollte, darunter das mobile Web. Gartner prognostiziert, dass im Jahr 2010 85 % der Mobiltelefone mit einem mobilen Browser unterschiedlicher Qualität ausgestattet sind. In Westeuropa werden etwa 60 % aller Mobiltelefone Smartphones sein. Diese besitzen einen hochentwickelten Browser und die Möglichkeit normale HTML-Seiten darzustellen. (vgl. Gartner Outlines 10 Mobile Technologies to Watch in 2010 and 2011 2010)

#### 1.1. Gegenstand und Ziel der Arbeit

Diese Fakten zeigen deutlich, dass mit dem mobilen Internet immer mehr Menschen zu erreichen sind. Es stellt deshalb ein relevantes Thema im Bereich der interaktiven Medien dar. In diesem Zusammenhang ist es bei der Erstellung mobiler Webseiten wichtig zu wissen, welche Probleme bei der Verwendung des Internets auf Mobiltelefonen auftreten. Im Weiteren ist es von Bedeutung, welche Strategien und Techniken existieren, um diesen Problemfeldern optimal zu begegnen. Diese beiden Fragen liegen der Arbeit als Forschungsfragen zu Grunde. Es sollen Antworten dazu gefunden werden, was der Entwickler bei der Erstellung von Webseiten, die für mobile Endgeräte bestimmt sind, berücksichtigen muss. Dabei wird im Speziellen auf technische und anwendungsbezogene Einschränkungen und Chancen sowie allgemein auf die Besonderheiten des mobilen Internets eingegangen. Während eines Experteninterviews mit FH-Prof. Dipl.-Ing. Grischa Schmiedl (2010) kristallisierten sich folgende Problemfelder heraus:

- Das Eingabeproblem: insbesondere bei tastaturlosen Geräten (Touchscreen-Geräten)
- Das Darstellungsproblem: bspw. Screengröße, Kontrast, Farbdarstellung
- Das Vollständigkeitsproblem: Vollständigkeit versus Übersichtlichkeit
- Das Limitationsproblem: in Bezug auf unterstützte Inhalte (bspw. Flash)

Diese Problemfelder sollen genauer betrachtet werden. Sie werden mit Hilfe einer Literaturrecherche ermittelt, aufgearbeitet, beschrieben und in einem Test mit einer Auswahl von aktuellen Smartphones analysiert. Durch das Erarbeiten bzw. Finden von Best Practices werden Möglichkeiten beschrieben, wie diesen Problemen und Chancen begegnet werden kann. Zum Abschluss wird im Sinne eines Proof of Concepts anhand von Beispielszenarien gezeigt, wie es möglich ist mit dieser Problematik umzugehen.

Aufgrund der aktuellen Marktsituation (vgl. Top 9 Mobile Browsers in Austria from May 09 to Apr 10 | StatCounter Global Stats 2010) und der Prognose von Gartner (vgl. Gartner Outlines 10 Mobile Technologies to Watch in 2010 and 2011 2010) betrachtet diese Arbeit die Probleme in Bezug auf Smartphones.

## 1. Die Verbreitung des mobilen Internets

---

### **1.2. Aufbau der Arbeit**

Im Mittelpunkt des zweiten Kapitels steht die Entwicklung des mobilen Internets und der entsprechenden mobilen Geräte. Gegenstand von Kapitel drei ist das Beschreiben, Analysieren und Beheben der Probleme, die im Zusammenhang mit der Benutzung des Internets über mobile Endgeräte entstehen. Im vierten Kapitel werden anhand der gefundenen Best Practices Beispielszenarien entwickelt. Den Abschluss bildet unter anderem die Betrachtung der Frage, wie bedeutsam die Entwicklung von mobilen Webapplikationen sein wird.

## 2. Das mobile Internet und die mobilen Geräte

### 2. Das mobile Internet und die mobilen Geräte

#### 2.1. Das mobile Endgerät

Ein mobiles Endgerät ist tragbar, persönlich, immer dabei, einfach und schnell zu nutzen und hat eine beliebige Form von Netzwerkanbindung (vgl. Firtman 2010, 4f).

Mobile Endgeräte werden in Mobiltelefone, Feature Phones, Smartphones, Non-Phones und Tablet-PC, Netbooks & Notebooks untergliedert. Tablet-PC, Netbooks und Notebooks besitzen ein Desktop-Betriebssystem mit einem Desktop-Browser (vgl. Firtman 2010, 6ff; Fling 2009, 6ff). Sie sind daher eher desktopverwandte als mobile Endgeräte und deshalb nicht Gegenstand dieser Arbeit.

##### 2.1.1. Mobiltelefone

Mit einem Mobiltelefon kann der Benutzer Anrufe tätigen und SMS schreiben. Es hat keinen Browser und keine Internetverbindung. Diese einfachen Mobiltelefone wurden überwiegend in den Jahre 1973 bis 1998 entwickelt.



Abbildung 1: Mobiltelefon - Nokia 1100 (Quelle: Nokia Schweiz - Nokia 1110i - Produkte 2010)

##### 2.1.2. Feature Phones

Feature Phones sind der Entwicklung nach in die Jahre 1998 bis 2008 einzuordnen. Sie besitzen durch die Einführung des GPRS-Standards zur Datenübertragung Internetzugang, MP3-Player und Kamera.



Abbildung 2: Feature Phone - Sony Ericsson K800i (Quelle: Sony Ericsson - Products - Mobile phones - Overview - K800i 2010)

## 2. Das mobile Internet und die mobilen Geräte

---

### 2.1.3. Smartphones

Die Smartphone-Ära begann im Jahr 2002. Smartphones besitzen ein multitaskingfähiges Betriebssystem, einen Browser, der Desktop-Websites vollständig anzeigt, sowie Wireless LAN (WLAN, WiFi), 3G, MP3-Player und oftmals viele der folgenden Features: GPS-Funktionen, digitaler Kompass, Kamera mit Videofunktion, TV out, Bluetooth, Touchscreen, Beschleunigungsmesser.



Abbildung 3: Smartphone - iPhone (Quelle: Apple - iPhone - Mobiltelefon, iPod und Internetgerät. 2010)

### 2.1.4. Non-Phone Geräte

Non-Phone Geräte sind keine Telefone, dennoch tragbar, persönlich, immer dabei, einfach und schnell zu nutzen. Sie besitzen eine Netzwerkverbindung und können oft, wie es beim iPad (Abb. 4) oder iPod Touch der Fall ist, in Sachen Funktionalität mit Smartphones mithalten. eBook-Reader gehören ebenfalls in diese Kategorie.



Abbildung 4: Non-Phone Gerät - iPad (Quelle: Apple 2010)



## 2. Das mobile Internet und die mobilen Geräte

### 2.2. Das mobile Internet

Das World Wide Web Consortium (W3C) definiert das mobile Internet als den Zugang zum Web mit einem mobilen Endgerät. Darunter sind die gleichen Webseiten zu verstehen, wie sie auch von einem Desktop-Computer erreichbar sind.

Abbildung 5 zeigt die Alternativen, wie mobile Geräte auf das Internet zugreifen. Das Gerät kann mit Hilfe des mobilen Browsers auf Desktop-Websites, auf Website, die speziell für mobile Endgeräte entwickelt wurden (Mobiles Web) und auf Desktop-Website, bei denen mobile Geräte in der Entwicklung Berücksichtigung fanden (Desktop Web - mobile ready), zugreifen. Zudem ist es möglich, dass eine Applikation auf das Internet zugreift, um Daten anzufordern.

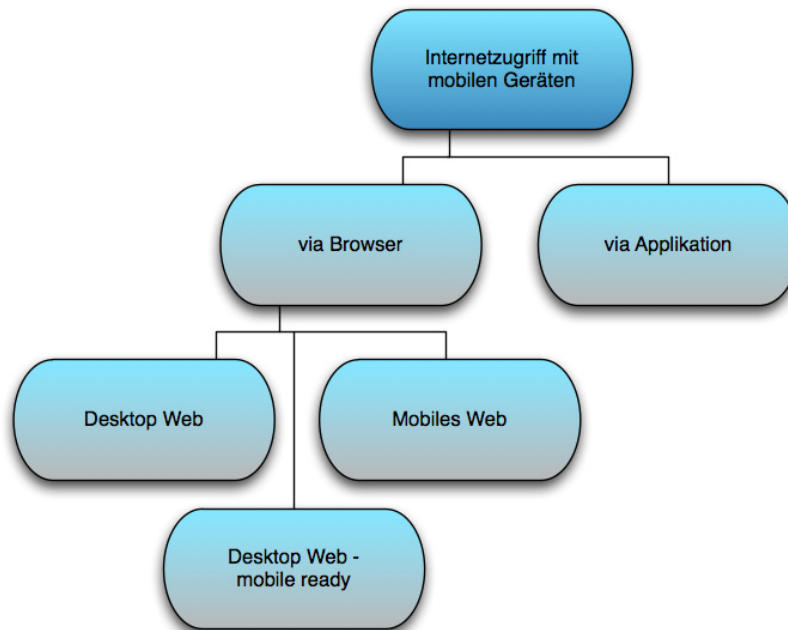


Abbildung 5: Internetzugriff mit mobilen Geräten (Quelle Kaikkonen 2008, 2; Schmiidl 2010)

#### 2.2.1. Internetzugriff via Browser

##### 2.2.1.1. Desktop Web

Desktop-Websites werden mit HTML entwickelt. Diese Seiten sind für die Benutzung mit einem Desktop-Computer vorgesehen. Mobile Geräte wurden bei der Entwicklung nicht berücksichtigt. Wird eine solche Website im mobilen Browser aufgerufen, bekommt der Benutzer den gleichen Inhalt wie am Desktop-Browser angezeigt. Das Layout dieser Seiten kann auf mobilen Geräten ungünstig aussehen. (vgl. Kaikkonen 2008, 2f)

##### 2.2.1.2. Mobiles Web

Eine Website für das mobile Web kann mit speziellen mobilen Auszeichnungssprachen (bspw. HDML, WML, cHTML) oder HTML geschrieben sein. Der Benutzer erkennt keinen Unterschied. Kaikkonen (2008, 2f) behauptet, dass eine Website genau dann für mobile Geräte entwickelt wurde, wenn spezielle Auszeichnungssprachen verwendet werden. Seitenstruktur und Inhalt differieren zur Desktopwebsite.

Dem muss jedoch widersprochen werden. Websites für mobile Endgeräte, die ein spezielles Design und angepasste Seitenstruktur haben, können nach neuen Erkenntnissen auch mit HTML geschrieben werden. Zudem müssen Seitenstruktur und Inhalt nicht differieren.

## 2. Das mobile Internet und die mobilen Geräte

---

### 2.2.1.3. Desktop Web - mobile ready

Desktop-Websites, bei deren Entwicklung mobile Geräte berücksichtigt wurden, sind dem Bereich Desktop Web - mobile ready zuzuordnen (vgl. Schmiedl 2010). Diese Websites werden mit HTML entwickelt. Sie sind in Sachen Seitenstruktur, Inhalt und Design sowohl für Desktop-Rechner also auch für mobile Geräte angepasst.

### 2.2.2. Internetzugriff via Applikation

Greift eine Anwendung, die auf einem mobilen Gerät ausgeführt wird, auf das Internet zu, so können Inhalte hoch- und heruntergeladen werden. Diese Anwendungen werden als native Applikationen bezeichnet. Sie können vom Internet heruntergeladen und auf dem Gerät installiert werden (vgl. Kaikkonen 2008, 3). Hierbei handelt es sich bspw. um Java- oder iPhone-Applikationen.

In dieser Arbeit wird der Internetzugriff via Applikationen nicht betrachtet. Der Schwerpunkt liegt auf dem Internetzugriff via Browser. Aufgrund der HTML-Fähigkeit der Browser von Smartphones ist dabei die aktuelle (XHTML 1.1) und zukünftige (HTML 5) HTML-Version (vgl. XHTML 1.1 - Module-based XHTML 2010; HTML5 2010) von Interesse.

## 2. Das mobile Internet und die mobilen Geräte

### 2.3. Die Testgeräte

Um die genannten Problemfelder (vgl. Kapitel 1.1) in der Praxis zu analysieren, kommen Testgeräte zu Einsatz. Diese wurden anhand aktueller Marktstatistiken von StatCounter Global Stats ausgewählt. In Abbildung 6 sind die neun am häufigsten benutzten mobilen Browser dargestellt. StatCounter ermittelt die Daten aus über 3 Millionen Websites und über 5 Milliarden Seitenzugriffen im Monat.

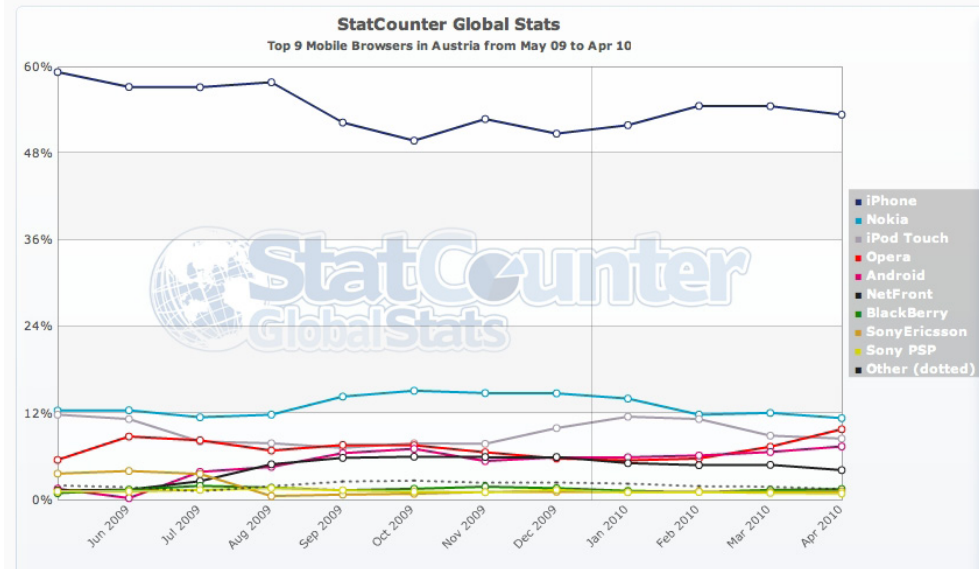


Abbildung 6: Top 9 der mobilen Browser von Mai 2009 bis April 2010 (Quelle: Top 9 Mobile Browsers in Austria from May 09 to Apr 10 | StatCounter Global Stats 2010)

Die Top 4 der mobilen Browser in Österreich sind demnach der Mobile Safari (integriert im iPhone und iPod Touch), die Browser von Nokia, Opera und der Android Browser. Die Wahl der Smartphones fiel deshalb auf das iPhone 3GS (Mobile Safari) und das HTC Legend (Android Browser). Mit dem Nokia N900 fällt die Wahl zusätzlich auf einen Vertreter mit dem linuxbasierten Betriebssystem Maemo mit dem zugehörigen Browser. Als Vertreter des Opera Browsers fiel die Wahl auf die aktuellste Version des Opera Mobile: Opera Mobile 10. Das zugehörige Testgerät ist das Sony Ericsson XPERIA X1 mit Windows Mobile 6.5 Professional.

## 2. Das mobile Internet und die mobilen Geräte

In Tabelle 1 sind die relevanten technischen Daten der Testgeräte aufgeführt.

	iPhone 3GS	HTC Legend	Nokia N900	Sony Ericsson XPERIA X1
Betriebssystem	iPhone OS 3.1.3	Android 2.1 Eclair	Maemo 5	Windows Mobile 6.5 Professional
Browser	Mobile Safari / WebKit	Android Browser / WebKit 3.1	Maemo Browser	Opera Mobile 10
Maße (H x B x T in mm)	115,5 x 62,1 x 12,3	112 x 56,3 x 11,5	110,9 x 59,8 x 18 (19,55)	110 x 53 x 16,7
Displaygröße / Diagonale	3,5 Zoll / 8,89 cm	3,2 Zoll / 8,1 cm	3,5 Zoll / 8,9 cm	3 Zoll / 7,62 cm
Auflösung in Pixel / PPI / Farben	480 x 320 / 163 ppi / 16,7 Mio	320 x 480 / 180 ppi / 16,7 Mio	800 x 480 / 267 ppi / 65,536	800 x 480 / 310 ppi / 65,536
Display-Typ	TFT	AMOLED	TFT	TFT
Touchscreen	Multitouch, kapazitiv	Multitouch, kapazitiv	resistiv	resistiv
Tastatur	-	-	QWERTZ - Quer	QWERTZ - Quer

**Tabelle 1: Technische Daten der Testgeräte (Quelle: Apple - iPhone - Technische Daten 2010; Apple iPhone 3GS - Technische Daten - Datenblatt 2010; HTC Legend - Technische Daten 2010; HTC Legend - Technische Daten - Datenblatt 2010; Nokia N900 - Technische Daten - Datenblatt 2010; Nokia Deutschland - Nokia N900 - Technische Daten 2010; Sony Ericsson XPERIA X1 - Technische Daten - Datenblatt 2010)**

### 3. Probleme bei der Verwendung des mobilen Internets

## 3. Probleme bei der Verwendung des mobilen Internets

### 3.1. Das Eingabeproblem

#### 3.1.1. Das Problem im Detail

Das Eingabeproblem beschäftigt sich mit dem Problem der eingeschränkten Bedienbarkeit eines Smartphones.

Das Gerät besitzt mindestens einen Bildschirm. Ist dieser ein Touchscreen, so kann es die einzige Möglichkeit sein mit dem Telefon zu interagieren. In den meisten Fällen besitzt das Gerät allerdings mindestens eine physikalische Taste. Zudem haben einige Geräte eine Hardware-QWERTZ-Tastatur.

Ein Hauptproblem bei der Eingabe stellt die Fingerbedienung bei Touchscreen-Geräten dar. Der Benutzer kann mit dem Finger nicht so präzise auf ein Element drücken, wie es mit der Maus an einem Desktop-Computer der Fall ist. Das Element ist meist kleiner als der Finger (fat finger problem). Außerdem verdeckt der Finger beim Drücken auf ein Element das Element selbst. Dies wird als occlusion problem bezeichnet. (vgl. Wigdor u. a. 2007, 209)

Die momentan am Markt erhältlichen Geräte mit Touchscreen können in zwei Kategorien unterteilt werden: Geräte mit resistiven und Geräte mit kapazitiven Touchscreens. Resistive Touchscreens lassen eine Interaktion mit allen körperlichen Gegenständen zu. Dies können bspw. Stifte oder Finger sind. Kapazitive Touchscreens können nur mit Fingern bedient werden. (vgl. Carbonell und European Research Consortium for Informatics and Mathematics. 2003) HTC macht diesen Nachteil mit einem Stift für kapazitive Displays wett (vgl. Fingerstift: HTC meldet Stylus für kapazitive Displays zum Patent an - PC-WELT 2009).

Während die resistiven Touchscreens einen Genauigkeitsvorteil bei der Bedienung mit Stiften haben, bieten die kapazitiven Touchscreens die Möglichkeit mit mehreren Fingern zu interagieren (Multitouch). Lee & Zhai (2009, 311f) verglichen in einem Experiment die Eingabe mit Stift und die Eingabe mit Fingern. Die Ergebnisse zeigen, dass die Eingabe mit Stift langsamer als die Fingereingabe ist. Die Fehlerrate ist bei der Fingereingabe jedoch höher.

Die Fehlerrate und Geschwindigkeit bei der Eingabe wird außerdem von der Weite der Tastatur beeinflusst. Touchscreen-Geräte bieten die Möglichkeit, über eine Software-Tastatur Zeichen einzugeben. Je nach Ausrichtung des Gerätes ändert sich die Weite der Tastatur. Hochformat bedingt eine enge Tastatur (Abb. 7), Querformat lässt eine breitere Tastatur zu (Abb. 8). Dies wirkt sich direkt auf die Eingabegeschwindigkeit (vgl. Lee und Zhai 2009, 315f) und somit auf die Bereitschaft, mit der der User eine Eingabe tätigt, aus.

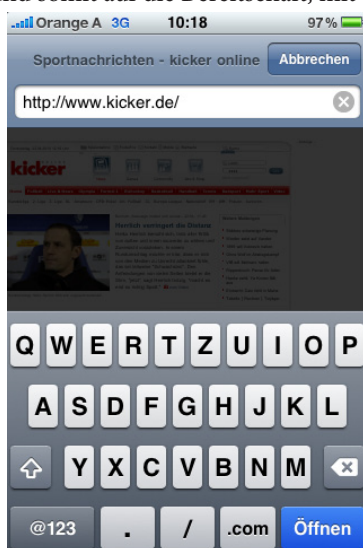


Abbildung 7: iPhone Tastatur Hochformat



Abbildung 8: iPhone Tastatur Querformat

### 3. Probleme bei der Verwendung des mobilen Internets

Ein taktiles Feedback verbessert zusätzlich die Bedienung des Gerätes in Bezug auf Fehlerfreiheit und Eingabezeit (vgl. Hoggan, Brewster, und Johnston 2008, 1576ff). Das HTC Legend besitzt taktiles Feedback. Der User spürt den Tastendruck durch Vibration.

Die Bereitschaft zum Tippen hängt weiterhin von der Situation ab, in der das mobile Internet und damit das mobile Gerät verwendet wird. Karlson, Bederson & Contreras-Vidal (2006, 2f) untersuchten, wann die Benutzer das Mobiltelefon ein- oder zweihändig bedienen. In Abbildung 9 wird deutlich, dass im Gehen die einhändige Bedienung signifikant bevorzugt wird. Im Sitzen kommen vermehrt beide Hände zum Einsatz.

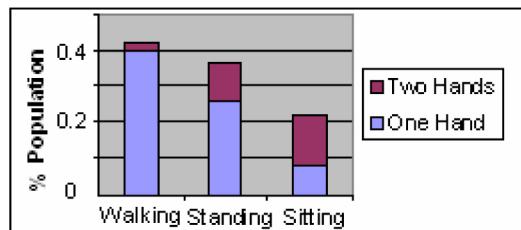


Abbildung 9: Nutzung von ein- oder zweihändiger Eingabe (Quelle: Karlson, Bederson & Contreras-Vidal: 2)

Basierend auf einer einhändigen Bedienung, stellt sich die Frage, ob die Position der zu berührenden Elemente einen Einfluss auf die Treffgenauigkeit hat. Diesem Problem widmeten sich Perry & Hourcade (2008) und Karlson, Bederson & Contreras-Vidal (2006). Die Untersuchungen zeigen, dass nicht alle Bereiche des Bildschirms gleichwertig bedient werden können. Folgende Faktoren beeinflussen die Fingerfertigkeit und die Reichweite des Daumens, der bei der Ein-Finger-Bedienung zumeist zum Einsatz kommt:

- Haltung des Gerätes
- Anatomie der Hand
- Ergonomie des Gerätes

Nach Perry & Hourcade (2008, 61) haben Rechtshänder eine eingeschränkte Daumenführung in Nord-Südost Richtung. In einer früheren Studie von Parhi, Karlson & Bederson (2006, 207ff) stellte sich bereits heraus, dass die Position der Elemente mehr Einfluss auf eine erfolgreiche Berührung hat als die Tastengröße.

#### 3.1.2. Das Problem in der Praxis

Eine Situation des Eingabe-Problems wird an dieser Stelle genauer betrachtet: Der User-Login.

Soll sich der Benutzer in einer Webapplikation einloggen, so muss er mindestens Usernamen und Passwort eingeben. Gerade bei der Eingabe des Passwortes kommt es zu Problemen, wenn anstelle der Zeichen nur Sterne oder Punkte zu sehen sind.

Passwörter wurden ursprünglich maskiert, damit sensible Daten nicht durch einen Zuseher beim Eintippen ausgespäht werden konnten. In der mobilen Welt ist die Situation anders als am Desktop-Rechner. Bei der limitierten Bildschirm- und Schriftgröße ist es sehr schwierig für eine andere Person zu sehen, was der Benutzer auf seinem Smartphone eintippt.

Hinzu kommt, dass das Tippen insbesondere von Sonderzeichen selbst bzw. gerade auf einem Gerät mit Hardware-Tastatur schwer ist. Wenn die Zeichen durch Sterne oder Punkte maskiert werden, kann der Benutzer nicht sicher sein, was er tippt. Es sei denn, die Zeichen werden für eine kurze Zeit angezeigt, bevor sie durch Sterne oder Punkte maskiert werden. (vgl. Firtman 2010, 134)

### 3. Probleme bei der Verwendung des mobilen Internets

Ein Test des `input`-Typs `password` im Testbrowser zeigt, dass drei von vier Browsern sich so verhalten (Abb. 10). Lediglich der Maemo Browser maskiert sofort.



Abbildung 10: `<input type="password" />` im Mobile Safari

Weitere Situationen, in denen das Eingabe-Problem relevant ist, sind Formulareingaben und das Eingeben von Webadressen.

#### 3.1.3. Best Practices

Die Begegnung mit dem Eingabe-Problem kann recht einfach gehandhabt werden: Benutzereingaben sollten, wann immer es geht, ganz unterbunden oder zumindest auf das Nötigste reduziert werden.

Eine denkbare Alternative zur formularbasierten Benutzer-Authentifizierung sind Clientzertifikate, die am Gerät installiert werden können. Bei dieser Art der Authentifizierung fordert der Server ein gültiges Zertifikat vom Browser an (vgl. Vorgehensweise: Einrichten von Clientzertifikaten 2004).

Auch eine Cookie-Lösung ist möglich. Ist der Benutzer einmal eingeloggt, wird er durch den Cookie wiedererkannt und die Anzahl der Login-Eingaben somit reduziert. Der Entwickler sollte sich jedoch nicht auf die Verfügbarkeit von Cookies verlassen. Alle Testgeräte unterstützen Cookies. Allerdings kann der Benutzer diese Funktion eigenständig deaktivieren. Deshalb sollte eine Prüfung der Verfügbarkeit von Cookies erfolgen. (vgl. Mobile Web Application Best Practices 2010)

Da es nicht immer möglich ist, Eingaben ganz wegzulassen, gibt es verschiedene Ansätze zur Erleichterung des Eingabevorgangs für den Benutzer. Beim User-Login sollten die Zeichen bei Passwordeingaben lesbar sein. Optional könnte der User die Auswahl haben, ob die Zeichen maskiert oder im Klartext zu sehen sind. Diese Aussagen werden durch Schmiedl, Seidl & Temper (2010, 2) und Firtman (2010, 134) gestützt. Sofern ein Entwickler doch den maskierten Typ bevorzugt, sollte er den Input auf Zahlen beschränken (vgl. Firtman 2010, 134). Dadurch hat der User eine geringere Fehlerquote weil bspw. Sonderzeichen sowie Groß- und Kleinschreibung wegfallen.

Zudem ist eine Location-gestützte Autovervollständigung beim Login an mobilen Geräten denkbar. Die Benutzerinformationen sind mit Locations verknüpft. Tippt der Benutzer seinen Usernamen, werden anhand seiner aktuellen GPS-Position per AJAX passende Vorschläge zur Autovervollständigung gegeben.

Die Autovervollständigung sollte so oft wie möglich eingesetzt werden. Eine Alternative sind ausgewählte Vorgaben: Was wird am meisten gewählt? Was ist am wahrscheinlichsten? Die Eingabefelder sollten mit der passenden Standard-Eingabemethode belegt sein bspw. Ziffern bei Postleitzahlen. Auswahllisten und Radiobuttons sollten anstelle von Freitext-Eingaben benutzt werden. (vgl. Bieh 2008, 216f)

### 3. Probleme bei der Verwendung des mobilen Internets

Sofern in Websites für mobile Geräte Formulare eingesetzt werden, sollten diese auf das Wesentliche beschränkt sein. Pflichtfelder sollen gekennzeichnet werden.

Wenn der Benutzer ein Input-Feld zur Eingabe auswählt, kann es passieren, dass er ein dazugehöriges Label, was vor oder über dem Inputfeld steht, nicht mehr sieht. Um dies zu umgehen, werden Platzhalter eingesetzt. Aufgrund des Platzproblems können diese Platzhalter auch als Ersatz für Labels verwendet werden. Platzhalter können mittels zweier Varianten erstellt werden:

- HTML5 bietet das Attribut `placeholder` im `input`-Tag (vgl. HTML5 2010):  
`<input type="text" name="zip" placeholder="Postleitzahl" />`
- Sobald der Benutzer die Eingabe beginnt, wird mit Javascript das `value`-Attribut im `input`-Tag gelöscht.

Problematisch ist hierbei, dass HTML5 derzeit kein offizieller Standard ist und daher nicht von allen Browsern unterstützt wird. Von den getesteten Browsern unterstützen nur Mobile Safari und Android Browser das `placeholder`-Attribut. Deshalb ist es wichtig zu prüfen, ob `placeholder` unterstützt wird, um gegebenenfalls die Javascript-Variante zu nutzen.



The image shows three text input fields stacked vertically. The top field contains the placeholder text "Name". The middle field is empty with a vertical cursor on the left side. The bottom field contains the placeholder text "Postleitzahl".



Abbildung 11: Verhalten von Mobile Safari beim `input`-Attribut `placeholder`

User-Inputs müssen so viel und so früh wie möglich (bereits bei der Eingabe) validiert werden. Die maximale Länge sollte über das Attribut `maxlength` festgelegt sein. Ein weiteres Hilfsmittel ist das Einblenden eines Zeichenzählers beim Tippen. (vgl. Firtman 2010, 136)

Um dem Benutzer der Website weitergehende Tipparbeit zu ersparen, sollten Telefonnummern und Email-Adressen für die vorgesehene Verwendung verlinkt werden.

Das W3C (vgl. Mobile Web Application Best Practices 2010) empfiehlt für Ein-Klick-Anrufe die Nutzung von `tel:<Telefonnummer>`. Dieser Wert wurde vom Japanischen Mi-Mode Standard kopiert und wird i-Mode Format genannt.

```
<a href="tel:[<Telefonnummer>]">Anrufen</a>
```

Um automatisch den Email-Client am Gerät aufzurufen, wird der Wert `mailto` des Attributs `href` im Tag `a` gesetzt.

```
<a href="mailto:[<Email-Adresse>][?Parameter]">Email schreiben</a>
```

Zudem können die Werte `sms[to]:` und `mms[to]:` in einigen Browsern genutzt werden, um den SMS- bzw. MMS-Client zu öffnen.



### 3. Probleme bei der Verwendung des mobilen Internets

Die Ergebnisse des Tests dieser Werte sind aus Tabelle 2 ersichtlich. Die Werte `tel:` und `mailto:` werden von allen getesteten Browsern verstanden. Der Android Browser zeigt bei den Werten `sms[to]:` und `mms[to]:` eine Fehlermeldung.

Browser	<code>tel:</code>	<code>mailto:</code>	<code>sms[to]:</code>	<code>mms[to]:</code>
Mobile Safari	ja	ja	nur sms:	-
Android Browser	ja	ja	Fehler: Prozess unerwartet beendet	Fehler: Prozess unerwartet beendet
Maemo Browser	ja	ja	nur sms:	nur sms:
Opera Mobile	ja	ja	nur sms:	ja

Tabelle 2: Kompatibilität der Werte `tel:`, `mailto:`, `sms[to]:`, `mms[to]:`

Bei der Benutzung eines mobilen Browsers ist die erste Hürde die Eingabe der Webadresse. In vielen Fällen sind diese lang und kryptisch. Webadressen von Einstiegsseiten sollten möglichst kurz gehalten werden. Diese Meinung vertreten auch Schmiedl, Seidl & Temper (2010, 2). Dadurch wird die Fehleranfälligkeit bei der Eingabe deutlich reduziert. Zudem kann der Webserver so konfiguriert werden, dass der Zugriff ohne Subdomain als Teil der Webadresse möglich ist. Der Benutzer gibt nun nicht mehr `www.beispiel.at/index.html` ein sondern `beispiel.at` (vgl. Bieh 2008, 173f).

2D-Barcodes (bspw. QR-Tags), Farb-Barcodes, RFID-Tags oder Bluetooth sind weitere Möglichkeiten, um dem Benutzer das Tippen der URL zu erleichtern bzw. zu ersparen (vgl. Bieh 2008, 173f).



Abbildung 12: QR-Code der URL <http://medieninformatik.fhstp.ac.at>



Abbildung 13: 8farbiger Barcode (Quelle: High Capacity Color Barcode Technology - Microsoft Research 2010)

Ein weiterer wichtiger Bestandteil zur Lösung des Eingabe-Problems ist die Größe der Elemente. Diese sollte im Designprozess starke Beachtung finden. Sie ist verantwortlich für das occlusion problem und bestimmt, wie komfortabel ein Element erreicht werden kann. Perry & Hourcade (2008, 64) schlussfolgern aus ihren Untersuchungen, dass nur die größte getestete Elementgröße von 11,5 mm eine Treffgenauigkeit von über 95 % mit sich bringt.

### 3. Probleme bei der Verwendung des mobilen Internets

Für die Zukunft wird vermutlich auch die Spracheingabe ein relevantes Thema im Bereich der Interaktion mit mobilen Geräten. Vorreiter auf diesem Gebiet ist die Firma Google, die mit ihrer nativen Google Mobile App bspw. einen Suchvorgang mit Hilfe der Stimme implementiert (vgl. Google Mobile App for your phone 2010). Auch die visuelle Eingabe mittels Kamera wird von Google forciert. Mit Google Goggles wurde ebenfalls eine native Applikation entwickelt, die durch Augmented Reality per Kamera Informationen zu verschiedensten Gegenständen liefert von der Sehenswürdigkeit über Bücher bis hin zur Umwandlung von Adressen und Telefonnummern auf Visitenkarten in Kontakte (vgl. Google Goggles for Android 2010).

Einen anderen Weg geht iConji. Die Firma Over the Sun, LLC möchte mit dieser Applikation die Kommunikation zwischen verschiedenen Nationen fördern und Sprachbarrieren abbauen. Die Kommunikation geschieht ausschließlich über Icons. Bei dieser vorerst nativen Anwendung wird gleichzeitig die Tipparbeit verringert. (vgl. iConji. Connecting the world. 2010)



Abbildung 14: iConji Applikation (Quelle: iConji. Connecting the world. 2010)

### 3. Probleme bei der Verwendung des mobilen Internets

#### 3.2. Das Darstellungsproblem

##### 3.2.1. Das Problem im Detail

Dem Darstellungsproblem liegt die Problematik der beschränkten Darstellungsmöglichkeiten auf mobilen Geräten zugrunde. Es beschäftigt sich vor allem mit der Frage, wie die Inhalte in mobilen Browsern dargestellt werden können.

Die Bildschirme der mobilen Geräte besitzen unterschiedlichste Größen. Während bei Desktop-Computern Bildschirmgrößen ab 17 Zoll üblich sind, verfügen mobile Displays meist nur über 1,5, 2,3 oder 3 Zoll. Die Auflösung eines Desktop-Bildschirms beträgt gewöhnlich mindestens 1024 x 768 Pixel. Die entsprechende Auflösung mobiler Geräte liegt bei einem Viertel oder der Hälfte der Desktop-Auflösung. Es gibt keine Standards bei mobilen Bildschirmauflösungen. Von 128 x 128 Pixel bis 800 x 600 Pixel sind verschiedene Auflösungsvarianten möglich. Die meist genutzte Auflösung ist 240 x 320 Pixel, was QVGA und somit einem Viertel der VGA-Auflösung von 1280 x 960 Pixel entspricht (vgl. Firtman 2010, 9f).

Zudem beeinflussen die möglichen darstellbaren Farben das Darstellungsproblem. 2009 schrieb Fling (2009, 125f), dass fast das gesamte Spektrum an Farben zur Verfügung steht. Die in den einzelnen Geräten benutzten Farbtiefen unterscheiden sich jedoch erheblich. Einige Beispiele können Tabelle 3 entnommen werden.

Farbtiefe	Unterstützte Farben	Beispielgeräte
12-bit	4.096	Nokia 6800
16-bit	65.536	Nokia N900, Sony Ericsson XPERIA X1, Blackberry Bold 9000
18-bit	262.144	Samsung Alias, Sony Ericsson TM506
24-bit	16,7 Mio.	iPhone 3GS, HTC Legend, Nokia N97

Tabelle 3: Farbtiefen bei mobilen Geräten (Quelle: Fling 2009, 125, leicht modifiziert)

Ein weiterer Teil des Problems ist die Darstellung von Schriften.

Wie von Desktop-Browsern bekannt, gibt es auch im Bereich der mobilen Browser nur sehr beschränkte Möglichkeiten zur Nutzung von Schriften. Traditionell konnte bei mobilen Geräten nur eine Schriftart genutzt werden: die Geräteschrift. Das Design wurde nur durch die Schriftgröße beeinflusst. Einige Betriebssysteme boten nur drei mögliche Font-Größen. Wurde die typische Definition mit Pixel genutzt, konnte 12 px und 14 px die gleiche tatsächliche Font-Größe bei der Darstellung sein. Hochauflösende Bildschirme erlauben heute eine größere Fontauswahl und sehr kleine Schriftgrößen (vgl. Fling 2009, 129f; Firtman 2010, 161; Schmiedl, Seidl, und Temper 2010, 2)

Eigene Fonts können allerdings nicht eingebettet werden. Auch dynamische Bildersetzungs-techniken wie FIR (Fahrner Image Replacement) oder sIFR (Scalable Inman Flash Replacement) kommen wegen mangelnder Unterstützung seitens der Geräte nicht in Frage. Bei diesen Techniken kann Text durch JavaScript und Flash oder von einer serverseitigen Scriptsprache durch dynamisch erzeugte Bilder ersetzt werden. (vgl. Bieh 2008, 72f)

### 3. Probleme bei der Verwendung des mobilen Internets

#### 3.2.2. Das Problem in der Praxis

Zwei der vier Testgeräte besitzen eine Auflösung von 320 x 480 Pixel (iPhone 3GS, HTC Legend) und zwei 480 x 800 Pixel (Nokia N900, Sony Ericsson XPERIA X1), allerdings bei unterschiedlichen Bildschirmgrößen. Das iPhone 3GS hat mit dem Nokia N900 den größten Bildschirm. Dieser ist 3,5 Zoll groß. Das HTC Legend verfügt über einen 3,2 Zoll großen und das Sony Ericsson über einen 3 Zoll großen Bildschirm. In Tabelle 4 sind die Pixeldichten der Testgeräte aufgelistet.

Testgerät	Pixeldichte
iPhone 3GS	163 ppi
HTC Legend	180 ppi
Nokia N900	267 ppi
Sony Ericsson XPERIA X1	310 ppi

Tabelle 4: Pixeldichte der Testgeräte

Da mobile Geräte unterschiedliche Pixeldichten aufweisen, kann es bei einer fixen Bestimmung der Position oder Größe eines Elementes via Pixelangabe zu Problemen kommen. Abbildung 15 veranschaulicht den Sachverhalt. Während auf einem Desktop-Monitor 72px 1 Zoll entspricht, sind es auf einem iPhone 3GS-Bildschirm (163 ppi) nur 0,4 Zoll. Auf dem Display des Sony Ericsson XPERIA X1 (310 ppi) ergeben 72 Pixel lediglich 0,2 Zoll.

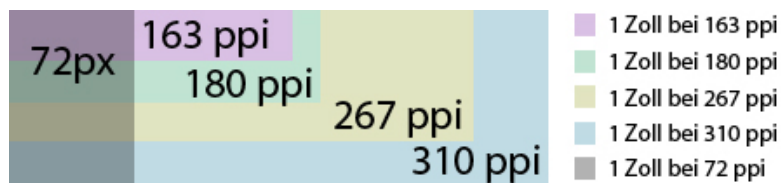


Abbildung 15: PPI-Problem – 72 Pixel im Verhältnis zu 1 Zoll bei unterschiedlichen Pixeldichten

Zudem ist die Farbdarstellung der Testgeräte unterschiedlich. Das iPhone und das Legend können 16,7 Millionen Farben darstellen, die beiden anderen nur 65.536 Farben.

Ein ausführlicher Vergleich der Desktop-Version der New York Times mit der mobilen Version der New York Times soll aufzeigen, wie das Problem der Darstellung von Inhalten in der Praxis gelöst wird.

### 3. Probleme bei der Verwendung des mobilen Internets

Die Desktop-Version der New York Times (U.S. Edition) (siehe Abb. 16) ist unter <http://www.nytimes.com> zu erreichen. Sie ist sechsspaltig aufgebaut und hat eine Länge von 3380 Pixel. Der User muss an einem Display mit einer Auflösung von 1440 x 900 Pixel vier Mal scrollen, um zum Ende der Startseite zu gelangen. Alle Neuigkeiten und wichtigen Informationen inklusive Werbung, Videos, Bilder und Login sind auf der Startseite untergebracht. In der linken Spalte sind sämtliche Kategorien der Seite aufgelistet. 17 Artikel sind mit einem Anrisstext auf der Startseite vertreten.



Abbildung 16: New York Times Desktop-Version  
(Quelle: The New York Times - Breaking News, World News & Multimedia 2010)

Die mobile Version der New York Times (U.S. Edition) ist zu erreichen unter <http://mobile.nytimes.com/main?edt=us> und in den Abbildungen 17 und 18 zu sehen. Sie ist überwiegend einspaltig aufgebaut und hat eine Länge von 2118 Pixel. Der User muss auf einem Bildschirm mit einer Auflösung von 320 x 480 Pixel fünf Mal scrollen, um zum Ende der Seite zu gelangen.

Ohne Scrollen erreichbar sind Wetter- und Börsennews (mit weiterführendem Link), die Suchfunktion und ein Link auf eine Liste aller Kategorien (Sections) der Website. Die Anrisstexte der zwei aktuellsten Artikel sind ebenfalls ohne Scrollen lesbar. In der Version, die mit dem Mobilien Safari und dem Android Browser angezeigt wird, sind neben den beiden Anrisstexten dazugehörige Bilder positioniert (siehe Abb. 17 = Mobile Version #1). Im Opera Mobile sind lediglich Titel und Anrisstexte zu lesen (siehe Abb. 18 = Mobile Version #2). Zudem fehlen in dieser Version die Suche, die Möglichkeit alle Kategorien anzeigen zu lassen und die Verlinkung zu den Aktienkursen. Der Maemo Browser zeigt die mobile Version der New York Times auf Grund eines XML-Verarbeitungsfehlers nicht an.

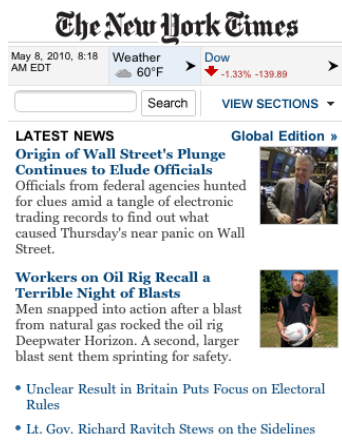


Abbildung 17: New York Times Mobil #1  
(Quelle: NYT Mobile 2010)



Abbildung 18: New York Times Mobil #2  
(Quelle: NYT Mobile 2010)

### 3. Probleme bei der Verwendung des mobilen Internets

Während in der mobilen Version #1 immer der gesamte Artikel auf einer Seite Platz findet, wird in der mobilen Version #2 der Artikel in mehrere Unterseiten gegliedert. In beiden mobilen Versionen wird ein Artikel einspaltig dargestellt. Ein Bild ist unterhalb des Titels platziert. Am Ende des Artikels ist der untere Teil der Startseite integriert. Im Gegensatz zur mobilen Version #1 können die Artikel auf der Desktop-Version in mehrere Seiten unterteilt werden. Die einspaltige Darstellung wird auch hier bevorzugt.

In der mobilen Version #1 wird die Schriftart („Georgia“) der Desktop-Website verwendet. Außerdem sind Links innerhalb des Artikeltextes vorhanden. Diese sind Blau (#004276) hervorgehoben und im Gegensatz zur Desktop-Version nicht unterstrichen. Bei der mobilen Version #2 kommt eine Systemschrift zum Einsatz. Innerhalb des Textes sind keine Links vorhanden.

Während auf der Startseite der Desktop-Website der New York Times alle möglichen relevanten Informationen dargestellt werden, ist das Design der mobilen Version wesentlich schlanker.

Die Entscheidung, welche Version der mobilen Website auf einem Gerät angezeigt wird, wird nicht aufgrund der Bildschirmauflösung getroffen. Dies lässt sich aus der höheren Auflösung (480 x 800 Pixel) des Sony Ericsson XPERIA X1 in Verbindung mit dem Browser Opera Mobile schließen. Es ist anzunehmen, dass diese Entscheidung vom benutzten Browser abhängt.

Bei den auf der Website gewählten Farben ist nur ein geringfügiger Unterschied zwischen den Versionen zu erkennen (sowohl Desktop vs. Mobil als auch Mobil vs. Mobil). Alle Versionen benutzen die Hintergrundfarbe Weiß (#FFFFFF). Als Paragraph-Farbe wird ein Dunkelgrau (#333333) und als Linkfarbe ein Blau (#004276) verwendet. Um Blöcke abzusetzen wird in der Desktop-Version Hellgelb, -blau oder -grau genutzt. In den mobilen Versionen wurde dafür lediglich hellgrau gewählt. Im Allgemeinen wird sehr sparsam mit Farben umgegangen. Bei der Gestaltung wurde der Bunt-Unbunt-Kontrast verwendet. Dieser basiert auf der Kombination einer Farbe mit Grau, Schwarz oder einer weniger gesättigten Farben (vgl. Boehringer, Buehler, und Schlaich 2003, 153).

#### **3.2.3. Best Practices**

Der Gestalter einer mobilen Website muss die Bildschirmauflösung der verschiedenen Geräte nicht berücksichtigen, sofern er folgende Punkte beachtet:

- Festsetzung des Viewports auf die eigentliche Screengröße
- Elemente nicht absolut definieren

Browser wie Webkit (bspw. Android Browser und Mobile Safari) erstellen einen virtuellen Viewport von 980 Pixel. Um den Viewport auf die eigentliche Screengröße zu setzen, wird folgender HTML-Tag verwendet (vgl. Fling 2009, 229):

```
<meta name="viewport" content="width=device-width" />
```

Die Größe und Position der Elemente sollten nicht absolut definiert werden. Relative Angaben wie Prozent (%) und em sind zu bevorzugen. (vgl. Bieh 2008, 68; SELFHTML: Style-sheets / CSS-Formate definieren / Maßeinheiten, Farbangaben und Wertzuweisung 2007)

Für Schriften empfiehlt Firtmann (2010, 161) relative Konstanten einzusetzen: xx-small, x-small, smaller, small, medium, large, larger, x-large, xx-large. Die Standard-Fontgröße (medium) ist die optimale Größe für einen normalen Textabsatz.

### 3. Probleme bei der Verwendung des mobilen Internets

Tabelle 5 zeigt Geräte mit Font-Unterstützung und die jeweils zur Verfügung stehenden Fonts.

Browser / Plattform	Nutzbare Fonts
iPhone / Mobile Safari	American Typewriter American Typewriter Condensed Arial Arial Rounded MT Bold Courier New Georgia Helvetica Marker Felt Times New Roman Kuba Trebuchet MS Verdana Zapfino
Android / Android Browser	Droid
WebOS / WebOS Browser	Arial Coconut Verdana
Symbian / Symbian Browser	S6o Sans

**Tabelle 5: Font-Unterstützung bei mobilen Geräten (Quelle: Firtman 2010, 160)**

Auf Grund der geringen Screenbreite und der zumeist vertikalen Ausrichtung wird eine mobile Webseite als ein vertikales Scrolldokument bezeichnet. Firtman (2010, 54) und Schmiedl et al. (2010, 2) unterstützen diese Aussage. Vertikales Scrollen an einem mobilen Endgerät stellt kein Problem dar. Horizontales Scrollen, das durch fixe Breiten und Positionierungen verursacht wird, ist jedoch problematisch.

Das typische Zwei- oder Drei-Spalten-Layout ist nicht brauchbar. Empfohlen wird das Ein-Spalten-Layout, welches auch bei der mobilen Version der New York Times zum Einsatz kommt (vgl. Kapitel 3.2.2.).

Zudem sollten die Inhalte zur besseren Übersicht gegliedert und Trennlinien genutzt werden. Jedes mobile Webdokument sollte folgende Bereiche besitzen: Header, Hauptnavigation, Inhalt, Nebennavigation und Footer. (vgl. Bieh 2008, 68)

Für den Landscape-Modus eines Mobiltelefons (Querformat) wird von Firtman (2010) empfohlen, eine alternative Organisation der Inhalte anzubieten, bei der die Hauptnavigation in einer rechten Spalte positioniert sein kann.

Mobile Endgeräte werden nicht wie Desktop-Rechner an einem Ort verwendet, der meist stabile Lichtverhältnisse aufweist, sondern oft bei Tageslicht und sogar direkter Sonneneinstrahlung. Dabei kommen Spiegelungen häufig vor. Somit ist die angezeigte mobile Website schwieriger zu erkennen als bei Dunkelheit oder diffusem Licht. Aus diesem Grund ist zu beachten, dass die Kombination aus Vorder- und Hintergrundfarben ausreichend Kontrast bietet. (vgl. Bieh 2008, 70 und 191)

Um eine Webanwendung auf unterschiedlichen Beleuchtungsarten zu testen, bietet Adobe seit der Adobe Creative Suite 3 Device Central an (vgl. Device Central 2006). Dieses Programm ist eine Testumgebung für mobile Anwendungen. Es beinhaltet eine Online-Bibliothek mit Geräteprofilen. Innerhalb der Software ist es möglich, das Verhalten der mobilen Anwendung unter verschiedenen Anzeigebedingungen zu simulieren. (vgl. Mobiltelefone, Entwicklung für Mobiltelefone | Adobe Device Central CS5 2010)

### 3. Probleme bei der Verwendung des mobilen Internets

#### **3.3. Das Vollständigkeitsproblem**

##### **3.3.1. Das Problem im Detail**

Das Vollständigkeitsproblem befasst sich mit der Entscheidung, was auf der mobilen Version einer Website dargestellt werden soll. Dabei steht die Vollständigkeit meist in Konflikt mit der Übersichtlichkeit. Benötigt der mobile User andere Informationen als der Desktop-User? Müssen dieselben Informationen anders aufbereitet oder strukturiert sein?

Das mobile Internet wird in verschiedenen Situationen genutzt. Kaikkonen (2008, 6) fand in ihren Untersuchungen heraus, dass die Menschen zu Hause oft das mobile Internet nutzen, wenn niemand da ist. Im Speziellen wird es zu Hause an Orten genutzt, an denen es nicht möglich ist, mit einem PC ins Internet zu gehen, bspw. auf der Couch im Wohnzimmer. Weitere relevante Situationen zur Nutzung des mobilen Internets sind:

- unterwegs
- mit öffentlichen Verkehrsmitteln fahrend oder auf diese wartend
- auf Arbeit oder im Büro in Pausen oder als zweiten Task

Die meisten Menschen sind beim Surfen mit dem mobilen Gerät allein. Sobald eine andere Person hinzukommt, wird die Aktivität beendet und eine Interaktion mit der Person angefangen. Ist das mobile Internet in einer Gruppe im Einsatz, wird es meist dazu verwendet, in einer Diskussion aufkommende Themen zu prüfen.

Die User des mobilen Internets führen zumeist folgende Aktivitäten durch (vgl. Kaikkonen 2008, 5f; Schmiedl u. a. 2009, 21f):

- Suche nach Telefonnummern und Adressen sowie anderen relevanten Informationen bspw. Öffnungszeiten
- Informationen abfragen mit Hilfe von Enzyklopädien und Wörterbüchern
- Nachrichten, Wetter und Sportnews lesen
- Fahrplanauskünfte abrufen
- anstellungsspezifische Informationen abrufen
- Interaktionen auf Community-Plattformen
- Emails lesen
- Bilder ansehen

In weiterer Folge fand Kaikkonen (2008, 5f) heraus, dass die Desktop-Version einer Website genutzt wird, wenn der User nach Informationen sucht, die nur auf der Desktop-Version zu finden sind. Gleichzeitig weiß der User, wo er die Informationen findet. Die mobile Version wird für bestimmte Aufgaben oder zum reinen Vergnügen bevorzugt.

Das mobile Web wird demnach in anderen Situationen genutzt als das Web am Desktop-Computer. Ein User wird mit einem Mobiltelefon keine komplexen Recherchen durchführen. Die meisten Benutzer wollen keine langen Texte auf dem Gerät lesen. Das erfordert nicht nur eine Optimierung der Darstellung der Inhalte für die mobile Nutzung, sondern auch die Optimierung der Inhalte selbst. Dies wird von Alby (2008, 33) bestätigt.



### 3. Probleme bei der Verwendung des mobilen Internets

#### 3.3.2. Das Problem in der Praxis

Für den Praxistest werden die Inhalte der Desktop- und der mobilen Version der Website derStandard.at verglichen.

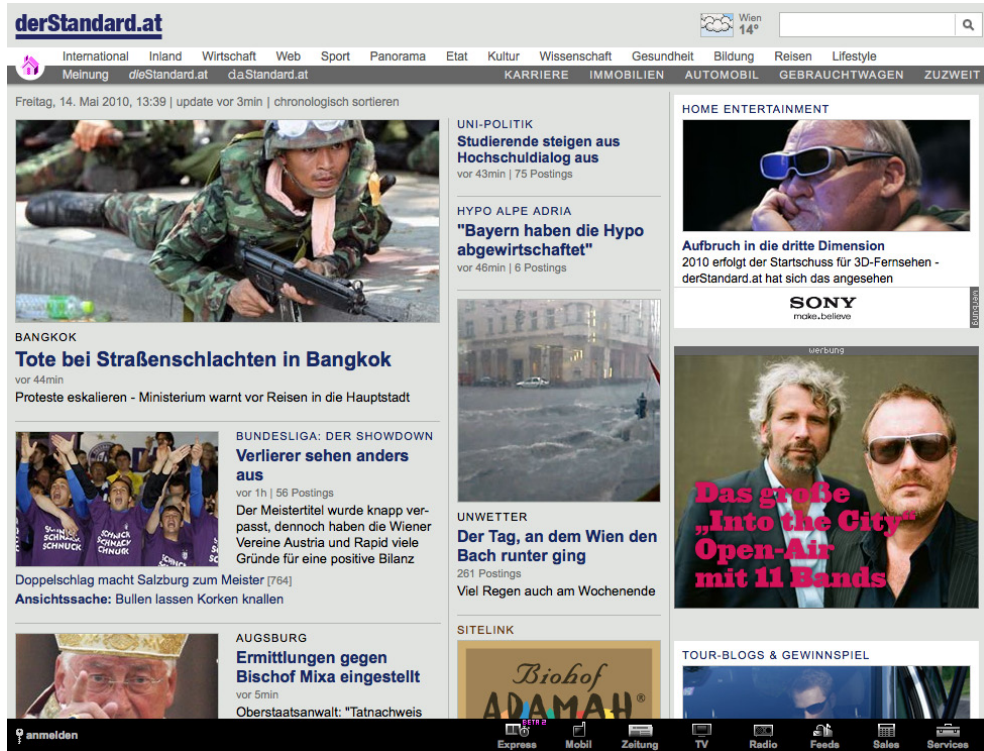


Abbildung 19: derStandard.at Desktop-Version (Quelle: derStandard.at 2010)

Im Header-Bereich der Desktop-Version (<http://www.derstandard.at>, siehe Abb. 19) sind die Suche, das Wetter und die Hauptnavigation mit Links zu allen Kategorien positioniert.

Im Folgenden ist die Startseite der Desktop-Version in mehrere Bereiche unterteilt. Im ersten Bereich werden aktuelle Meldungen angezeigt. Hier finden sich neun Artikel. Davon sind sieben Artikel mit Bildern und sechs mit Anrisstexten versehen. Zudem ist die Werbung, die in der rechten Spalte positioniert ist, dominant. Zu einem Artikel sind Videos verlinkt. Abgeteilt wird dieser Bereich durch einen Society-Block mit Links zu den Themen Kochen, Reisen, Motorrad, Sudoku uvm. Im Weiteren folgen zu jeder Kategorie der Website die Top-Meldungen zumeist mit Bild und Anrisstext.

### 3. Probleme bei der Verwendung des mobilen Internets



Abbildung 20: derStandard.at Desktop-Version Artikelseite (Quelle: Tote bei Straßenschlachten in Bangkok - Thailand - derStandard.at › International 2010)

Ein Artikel auf der Desktop-Version (siehe Abb. 20) ist nicht in mehrere Seiten unterteilt. Der gewählte Artikel „Tote bei Straßenschlachten in Bangkok“ ist mit zwei Bildern (jeweils mit Bildunterschrift) ausgestattet, in sechs Absätze unterteilt und mit zwei Zwischenüberschriften versehen. Am Ende des Artikels werden alle Leserkommentare angezeigt. In der rechten Spalte werden dem Leser mit dem Thema verwandte Artikel mit Bild und Anrisstext angezeigt.



Abbildung 21: derStandard.at Mobil (Quelle: derStandard.at Mobile 2010)



Abbildung 22: derStandard Mobil Artikelseite (Quelle: Tote bei Straßenschlachten in Bangkok - Thailand - derStandard.at › International Mobile 2010)

### 3. Probleme bei der Verwendung des mobilen Internets

Auf der Startseite der mobilen Version (<http://mobile.derstandard.at>, siehe Abb. 21) sind neun Artikel zu sehen. Alle Artikel sind mit Titel und kurzem Anrisstext versehen. Im Header- und Footer-Bereich sind Links zu Kategorien positioniert. Zudem gibt es im Footer einen Link zur Vollversion (Desktop-Version) und die Suchfunktion.

Auf der mobilen Version wird der gesamte Artikel einspaltig auf einer Seite angezeigt (siehe Abb. 22). Eine Anzeige der Bilder erfolgt nicht. Es gibt jedoch einen Hinweis darauf, dass zwei Bilder vorhanden sind.

Der Artikel ist wie bei der Desktop-Version in sechs Absätze und zwei Zwischenüberschriften untergliedert. Die Leserkommentare werden nicht direkt angezeigt. Der User hat die Möglichkeit, diese über Links zu erreichen. Auf die Anzeige von verwandten Artikeln wird verzichtet.

Breadcrumbs „derStandard.at > International > Asien-Pazifik > Thailand“ weisen sowohl in der Desktop- als auch in der mobilen Version die Position des Users innerhalb der Website aus.

Es befinden sich keine Werbung und keine Videos auf der mobilen Version. Auch auf Bilder wird verzichtet, es sei denn der User ruft diese explizit auf.

Die Anrisstexte und Titel der Desktop- und der mobilen Version sind gleich lang. Zur selben Abrufzeit sind die Artikel des aktuellen Bereiches der Desktop-Version auch auf der Startseite der mobilen Version zu sehen.

#### **3.3.3. Best Practices**

Eine Herangehensweise an das Vollständigkeitsproblem beschreibt Firtman (2010, 50f) mit der 80/20 Regel. 80 % der Desktop-Version interessieren den mobilen User nicht. Beim Entwickeln einer mobilen Website kommt es somit auf die 20 % an, die den User interessieren. Um zu diesen 20 % zu gelangen, müssen Anwendungsszenarien erstellt und nach Wichtigkeit priorisiert werden. Jedes Szenario soll in nicht mehr als drei Klicks oder drei Seiten realisiert werden. Unterhalb der Startseite sollten nicht mehr als drei Hauptsektionen existieren. Wenn mehr benötigt werden, ist die mobile Webseite in mehrere mobile Seiten zu unterteilen.

Grundsätzlich sollte die mobile Seitenarchitektur so einfach, wie möglich gehalten sein (vgl. Fling 2009, 93). Wie in Kapitel 3.3.2 bei derStandard.at zu sehen, wird auf alles Unwichtige wie Bilder, Videos, Werbung oder die Postings der Leser verzichtet. Der Benutzer hat allerdings die Möglichkeit bestimmte Elemente, wie Bilder und Leserkommentare, explizit aufzurufen. Wichtig dabei ist eine ausgewogene Struktur. Der User benötigt nur wenige Links auf einer Seite, damit er die gesuchte Seite mittels weniger Klicks erreicht. Dabei sollte der Entwickler eine ausgewogene Balance zwischen einer großen Anzahl an Navigationslinks und der Notwendigkeit durch viele Links zu navigieren finden.

Die Relevanz der bereitstehenden Informationen ist zu berücksichtigen. Häufig abgerufene Informationen sollten leichter zugänglich sein. Seltener abgerufene Seiten können auch auf Unterseiten zu finden sein. Sinnvoll ist eine Priorisierung der Links nach Popularität. Außerdem ist eine Limitierung auf zehn Links pro Seite ratsam. (vgl. Bieh 2008, 175; Fling 2009, 179)

### 3. Probleme bei der Verwendung des mobilen Internets

Jede Website sollte einen Titel haben, damit sich der User besser orientieren kann. Dieser sollte zwischen vier und acht Worte lang sein. Welcher Browser welche Titel-Länge unterstützt, ist Tabelle 6 zu entnehmen. (vgl. Firtman 2010, 101f)

Browser / Plattform	Darstellbare Titellänge (title Attribut)
iPhone / Mobile Safari	40 Zeichen in Portrait (Hochformat) und 60 in Landscape (Querformat), wird versteckt, wenn der User scrollt und in Webapplikationen
Android / Android Browser	15 Zeichen nach dem Domainnamen in 1.0 und 1.5 / keine Unterstützung nach 2.0
Nokia Series 40	13 Zeichen in 2nd Edition / 20 Zeichen in 3rd Edition / keine Unterstützung in 5th und 6th Edition
Palm Web OS	keine Unterstützung
BlackBerry	15 bis 25 Zeichen hängt von Screenbreite ab
NetFront	keine Unterstützung
Myriad	15 Zeichen
Internet Explorer	keine Unterstützung
Motorola	15 Zeichen
Opera Mobile	hängt von Screengröße ab, zwischen 20 und 60 Zeichen
Opera Mini	hängt von Screengröße ab, zwischen 20 und 60 Zeichen

**Tabelle 6: Darstellbare Titellänge der Browser (Quelle: Firtman 2010, 101f)**

Ein weiteres wichtiges Element einer mobilen Website ist der Link zur Desktop-Version. Dieser sollte auf jeder Seite vorhanden sein, um dem User die Wahl der Version zu ermöglichen (vgl. Schmiedl, Seidl, und Temper 2010, 2). Der Link kann beispielsweise wie bei derStandard.at im Footer positioniert sein.

In der Literatur wird das Konzept von derStandard.at die vollständigen Artikel auf einer Seite anzuzeigen als unvorteilhaft angesehen. Alby (2008, 33) ist der Meinung, dass die meisten Benutzer keine langen Texte auf das Handy laden und durchscrollen wollen. Bieh (2008, 184) bekräftigt diese Meinung und möchte den Inhalt auf das beschränken, was der Nutzer abgefragt hat. „Seiten wie Spiegel.mobi oder Kicker.mobi kürzen Texte meist auf ca. 2000 Zeichen, ehe ein Benutzer eine Seite weiterblättern muss.“ (Bieh 2008, 186)

Der Praxistest mit derStandard.at zeigt in Kapitel 3.3.2, dass die Darstellung des gesamten Artikels auf einer Seite auf modernen Smartphones keine erkennbaren Probleme mit sich bringt. Ein weiterer Test mit der Website spiegel.mobi (vgl. Datenschutz bei Facebook: Wie die Privatsphäre erodiert - SPIEGEL ONLINE - Nachrichten - Netzwelt 2010) zeigt, dass bei Aufruf eines Artikels ebenfalls der vollständige Artikel auf einer Seite mit über 8900 Zeichen angezeigt wird. Die Meinung, dass Text nicht auf mehrere Seiten aufgesplittet werden muss, unterstützen auch Schmiedl, Seidl & Temper (2010, 2).

### 3. Probleme bei der Verwendung des mobilen Internets

#### 3.4. Das Limitationsproblem

##### 3.4.1. Das Problem im Detail

Das Limitationsproblem befasst sich mit technischen Einschränkungen der mobilen Geräte.

Verschiedene mobile Geräte bedeuten unterschiedliche Funktionen. Selbst Geräte eines Herstellers interpretieren Befehle unterschiedlich. Standards haben sich im Bereich des mobilen Webs bisher nicht so etabliert wie im Desktop Web. Es ist viel Zusatzarbeit nötig, um die jeweiligen Fähigkeiten einer Plattform angemessen zu berücksichtigen. Der Benutzer des mobilen Internets hat oft keine Wahl zwischen verschiedenen Browsern, so wie es am Desktop-Rechner der Fall ist. (vgl. Bieh 2008, 157; Alby 2008, 66) Dem Entwickler bleibt bei der Programmierung die Wahl zwischen der Findung des kleinsten gemeinsamen Nenners oder einer aufwändigen Anpassung an die unterschiedlichen Geräteklassen.

Wird vom mobilen Web gesprochen, sollte die genutzte Übertragungstechnik nicht außer Acht gelassen werden. Das mobile Gerät ist grundsätzlich kabellos mit dem Internet via Sendemasten oder WLAN-Router verbunden. Vor allem bei Ortsänderungen sind mobile Verbindungen als nicht stabil einzustufen. Das Sendesignal der Sendemasten ist nicht stark genug und die Verbindung reißt ab. Die neuesten Smartphones bauen in der Regel mit UMTS oder WLAN die Verbindung ins Internet auf. Damit können im UMTS-Bereich Download-Geschwindigkeiten bis zu 14,4 Mbit/s erreicht werden (vgl. Alby 2008, 26; Bieh 2008, 159). In der Realität bietet bspw. AON via HSDPA 7,2 Mbit/s. Im Vergleich dazu steht eine normale festnetzgebundene Breitbandverbindung mit 8 Mbit/s (best effort) (vgl. aon Breitband-Duo 2009, 2). Tatsächlich erreichen sowohl die Breitbandverbindung als auch die mobile Verbindung diese Geschwindigkeiten zumeist nicht. Die Zeitschrift Konsument testete die Downloadgeschwindigkeiten der mobilen Internetverbindungen. Wie Abbildung 23 zu entnehmen ist, erreichte die mobile AON-Verbindung bei diesem Test effektiv 5,76 Mbit/s.

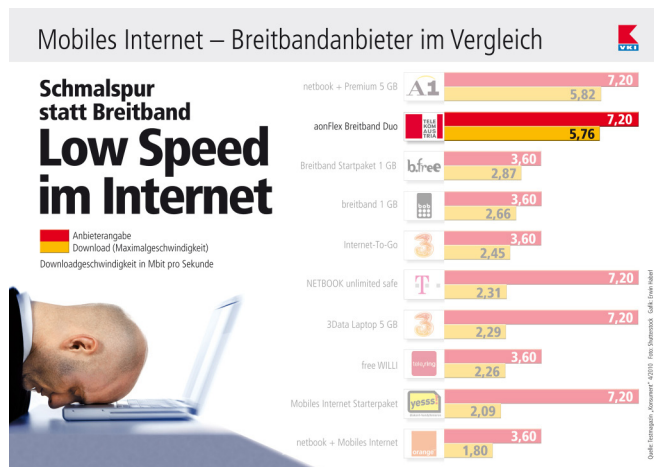


Abbildung 23: Downloadgeschwindigkeiten mobiler Internetverbindungen  
(Quelle: KONSUMENT.AT - Mobiles Internet 2010; Mobiles Breitband-Internet:  
Langsam und teuer - Mobilfunk - derStandard.at › Web 2010; leicht modifiziert)

Ein viel diskutiertes Limitationsproblem im mobilen Bereich ist die Flash-Technologie. In Desktop-Browsern kann der Flash Player von Adobe als defacto-Standard für Animationen und Videos angesehen werden. Der Flash Player 8 hatte im September 2009 eine Verbreitung von mehr als 99,5 % (vgl. Firtman 2010, 144ff). In der mobilen Welt verhält es sich anders. Momentan gibt es bei Adobe zwei Entwicklungsrichtungen für mobile Geräte (vgl. Adobe Labs - Adobe Flash Player 10.1 2010; Adobe - Flash Lite 2010):

- Flash Lite
- Flash Player 10

### 3. Probleme bei der Verwendung des mobilen Internets

Flash Lite ist die Flash Player Alternative für Feature Phones und Smartphones vor 2010. Den Flash Player 10 gibt es bisher für Android, Symbian, Palm Pre und Windows Mobile. Auf dem iPhone ist Flash derzeit nicht verfügbar.

Flash Lite kann in Menüs, Hintergründen, Spielen, Applikationen und im Browser benutzt werden. Die Unterschiede zwischen Flash Lite 1.0 und 4.0 sind sehr groß, deshalb wird zur Nutzung von Flash im mobilen Web nur geraten, wenn mit einer kontrollierten Auswahl an Geräten gearbeitet wird und diese getestet werden können. (vgl. Firtman 2010, 144ff)

Ein weiteres Teilproblem ist die JavaScript-Unterstützung. Mittlerweile verstehen alle modernen mobilen Browser JavaScript. Bei der Unterstützung des HTTP-Request-Objektes (AJAX) sind Unterschiede zu beachten, diese sind Tabelle 6 zu entnehmen.

Browser / Plattform	JavaScript	AJAX
iPhone / Mobile Safari	ja	ja
Android / Android Browser	ja	ja
Symbian / S60	ja	ja seit 3rd Edition
Nokia Series 40	ja	nein vor 6th Edition
Palm Web OS	ja	ja
BlackBerry	ja ab 3.8	nein vor 4.6
NetFront (Sony Ericsson und andere)	ja	nein vor 3.5
Internet Explorer	ja	ja seit Windows Mobile 5
Motorola Internet Explorer	ja	nein
Opera Mobile	ja	ja seit 8.0
Opera Mini	ja, aber alles nach onload wird am Server ausgeführt, was einen Postback generiert	ja seit 3.0

Tabelle 7: JavaScript-Unterstützung (Quelle: Firtman 2010, 190, 231f; leicht modifiziert)

### 3. Probleme bei der Verwendung des mobilen Internets

#### 3.4.2. Das Problem in der Praxis

In der Praxis soll an dieser Stelle auf das Filesystem-Problem geblickt werden.

Der Typ `file` im Tag `input` ist im mobilen Web-Standard nicht enthalten. Zwei der vier Testbrowser (Opera Mobile und Maemo Browser, siehe Abb. 24) interpretieren den HTML-Tag dennoch richtig und öffnen den Dateibrowser. Es öffnet sich ein Popup-Fenster und der User kann im lokalen öffentlichen Speicher oder einer Speicherkarte eine Datei auswählen. Mobile Safari und Android Browser interpretieren diesen Tag nicht (siehe Abb. 25).

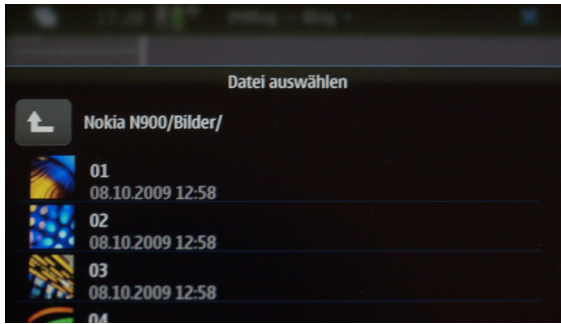


Abbildung 24: Maemo Browser - `<input type="file" />`

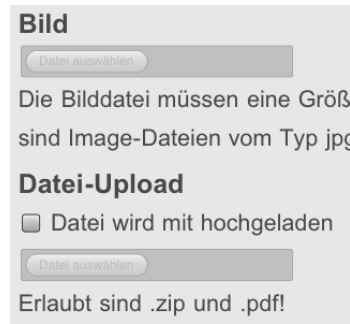


Abbildung 25: Mobile Safari - `<input type="file" />`

Zudem erlauben einige mobile Browser kein Dokumenten-Download. Es sei denn, es ist ein für das jeweilige Dokument kompatibler Reader installiert. Wie mobile Browser mit Datei-downloads umgehen, ist Tabelle 7 zu entnehmen.

Browser / Plattform	Unknown File	Adobe PDF	Microsoft Office
iPhone / Mobile Safari	nein	ja: PDF Reader	ja: 2007 & 2010 (teilweise) Word, Excel, Powerpoint Reader
Android / Android Browser	ja	ja, aber User muss einen kompatiblen Reader oder Editor installiert haben	
Symbian / S60	ja	ja, aber User muss einen kompatiblen Reader oder Editor installiert haben	
Nokia Series 40	ja	ja, aber kein Reader da; nur Download	
Palm Web OS	ja	ja, aber User muss einen kompatiblen Reader oder Editor installiert haben	
BlackBerry	ja	ja	ja: Word, Excel, Powerpoint 2007
NetFront	ja	nur Download	
Internet Explorer	ja	benötigt Reader	ja
Motorola Internet Browser	nein	nein	nein
Opera Mobile	ja	Download / Öffnen hängt vom Gerät ab	
Opera Mini	ja, wenn Gerät File API von Java ME unterstützt	Download / Öffnen hängt vom Gerät ab	

Tabelle 8: Download-Kompatibilität (Quelle Firtman 2010, 126f; leicht modifiziert)

### 3. Probleme bei der Verwendung des mobilen Internets

#### 3.4.3. Best Practices

Will der Entwickler sicher gehen, dass ein Gerät den gewählten Befehl interpretieren kann, muss er die Kompatibilität des Befehls prüfen. Dazu bietet die WURFL-Datenbank dem Entwickler die Möglichkeit herauszufinden, was das Gerät des Users kann. WURFL baut auf das User Agent Profile auf. Somit kann dem Gerät das übertragen werden, was es interpretieren kann. (vgl. Passani 2010)

Dennoch sollten auf `no script`-Tags, deren Inhalte angezeigt werden, wenn der Browser bspw. kein JavaScript versteht, nicht verzichtet werden. Der User könnte JavaScript im Browser deaktiviert haben.

Als Alternative zu Flash bietet sich der zukünftige W3C-Standard HTML5 an (vgl. HTML5 2010). Die verantwortliche Arbeitsgruppe WHATWG (Web Hypertext Application Technology Working Group) prognostiziert die Fertigstellung von HTML5 für 2012 (vgl. FAQ - WHATWG Wiki 2010). Bereits jetzt sind einige Tags in den Browsern integriert. Das `video`-Tag bietet sich als Flash-Ersatz für die Anzeige von Videos an.

In den Testbrowsern Mobile Safari und Android Browser wird dieser Tag erkannt (siehe Abb. 26 und 27). Während der Mobile Safari H.264 kodierte Videos im Quicktime- oder MPEG4-Container abspielt, benötigt der Android Browser explizit ein H.264-Video im MPEG4-Container, kodiert im Baseline-Profil. Opera Mobile und der Maemo-Browser verstehen den HTML5 `video`-Tag nicht (siehe Abb. 28).



Abbildung 26: `<video>` im Mobile Safari und Android Browser #1

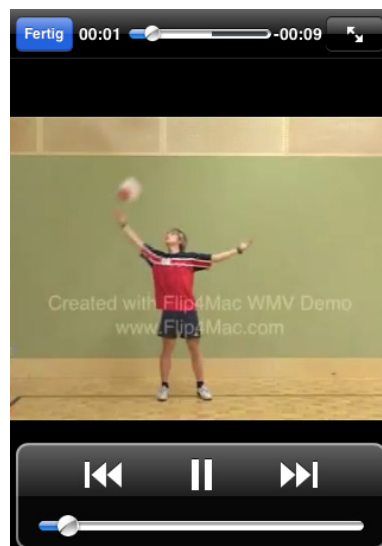


Abbildung 27: `<video>` im Mobile Safari und Android Browser #2

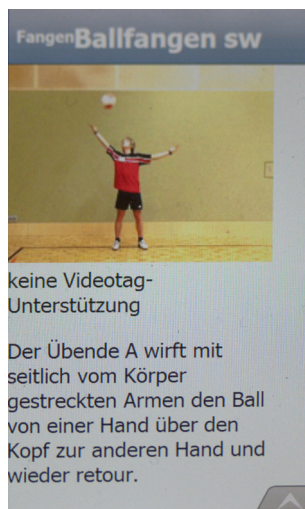


Abbildung 28: `<video>` im Opera Mobile und Maemo Browser



### 3. Probleme bei der Verwendung des mobilen Internets

Eine weitere Flash-Alternative beruht auf SVG. Tobias Schneider (2010) entwickelte ein Open Source Flash Runtime, das Javascript und SVG benutzt. Das SWF muss dabei allerdings als Flash Player 1 oder 2 exportiert werden (aktuell Flash Player 10), womit viele Features der aktuelleren Versionen nicht interpretiert werden.

Eine Vermutung, warum bisher speziell Mobile Safari und der Android Browser den Dateiupload nicht erlauben, ist, dass der Zugriff auf den internen Speicher des Gerätes als Sicherheitsproblem angesehen wird. Das W3C arbeitet im Zusammenhang mit HTML5 an einer neuen FileAPI. Diese lässt hoffen, dass der `input`-Typ `file` in zukünftigen mobilen Browserversionen integriert wird. (vgl. File API 2009) Momentan sollte der Dateiupload vermieden werden. Ein Mail2App Gateway könnte ein möglicher Workaround bspw. für Fotouploads sein (vgl. Schmiedl, Seidl, und Temper 2010, 2). Doch für den User ist diese Variante nicht komfortabel. Sowohl beim HTC Legend als auch beim iPhone 3GS ist es nicht möglich, aus dem Email-Client heraus eine Email mit Fotoanhang zu versenden. Der User muss das Foto in der Fotoapplikation auswählen, um es per Email versenden zu können. Somit hilft ihm eine `mailto`:-Verlinkung in der Webapplikation wenig. Er muss sowohl den Betreff als auch die Email-Adresse eingeben oder die Adresse im Adressbuch gespeichert haben.

Image Exchange bietet in einer Betaversion einen Mobile Image Exchange. Der User sendet eine Email mit Fotoanhang an Image Exchange und bekommt einen Link zum Foto zurück. Diesen Link kann er in der entsprechenden Website integrieren. (Image Exchange - Free Image Hosting & Photo Sharing 2010) Der Nachteil dieser Variante ist, dass die Bilder nicht auf dem eigenen Server sondern bei Image Exchange gespeichert werden.

Speziell für das iPhone ist ein iPhone Foto Picker in der Entwicklung. Möchte ein iPhone-Nutzer in einer Webapplikation ein Foto hochladen, wird er, sofern die Webanwendung dies unterstützt, auf eine native Applikation, die auf dem iPhone installiert sein muss, weitergeleitet und kann aus dieser das Foto hochladen. Im Anschluss wird auf eine von der Webapplikation vorgegebene URL weitergeleitet. Die funktionstüchtige Anwendung liegt als Open Source vor, ist allerdings bisher nicht im App-Store erhältlich. (vgl. README - iphone-photo-picker - Project Hosting on Google Code 2009)

Lange und andauernde Datenübertragungen sollten begrenzt werden. In diesem Zusammenhang sollte auf Autorefresh verzichtet werden. Es sei denn, der Benutzer wird darüber benachrichtigt und stimmt zu. Ein Autorefresh kann über die `meta`-Attribute `http-equiv` und `content` bestimmt werden:

```
<meta http-equiv="refresh" content="0" />
```

Zudem sollten Caching-Informationen genutzt werden, um unnötige Reloads zu vermeiden:

```
<meta http-equiv="Cache-Control" content="max-age=300" />
```

Dabei ist zu beachten, dass nicht alle Geräte Cache-Control unterstützen (vgl. Fling 2009, 175f).

## 4. Lösungsansätze in der Praxis

### 4.1. Der Fotoupload

Das Limitationsproblem beinhaltet das Problem des Dateiuploads (vgl. Kapitel 3.3.2). Im Praxistest stellte sich heraus, dass Mobile Safari und der Android Browser keinen Dateiupload in Webapplikationen unterstützen. Um diesen Benutzern die Möglichkeit zu geben, Bilder bspw. in eine Galerie hochzuladen, wird ein PHP-Skript entwickelt, welches die Emails eines bestimmten Accounts ausliest und das Bild aus dem Email-Anhang anzeigt. In Listing 1 ist dieses Skript zu sehen.

```
1  <?php
2  // Mailbox öffnen
3  $mailbox = imap_open("{[IMAP-Server]:143/imap/notls}INBOX", "[USERNAME]",
4  "[PASSWORT]") or die("Can't connect IMAP-Server!");
5  // Mails nach Kriterien suchen
6  $mailArray = imap_search($mailbox, "FLAGGED");
7
8  foreach($mailArray as $mailA){
9      // Struktur der Mail auslesen
10     $mail = imap_fetchstructure($mailbox, $mailA);
11     // Teile der Mail zählen
12     $partsInMail = count($mail->parts);
13     if($partsInMail >= 2){
14         // Lese den ersten Anhang
15         $inhalt = imap_fetchbody($mailbox, $mailA, 2);
16         header('Content-type: image/jpeg');
17         echo base64_decode($inhalt);
18     }
19 }
20 ?>
```

Listing 1: PHP-Skript zum Abrufen von Emails und darstellen des Bildanhanges

Das PHP-Skript verwendet die PHP-IMAP-Bibliothek. Der User bekommt eine Email-Adresse mitgeteilt, an die er Bilder als Email-Anhang senden kann.

In Zeile 3 wird mit `imap_open` die Mailbox geöffnet. Diese Funktion muss mit den entsprechenden Werten für die Parameter [IMAP-Server], [USERNAME] und [PASSWORT] befüllt werden. Anschließend (Zeile 5) wird im Postfach eine Suche durchgeführt. In diesem Beispiel wird nach dem FLAG gesucht, der bei markierten Emails gesetzt wird. Weitere mögliche Suchparameter der Funktion `imap_search` sind bspw. Absender (FROM), neue Nachrichten (NEW) oder Betreff (SUBJECT) (vgl. PHP: `imap_search` - Manual 2010).

In einer ForEach-Schleife werden die Suchergebnisse durchlaufen. Die Struktur und die Teile der jeweiligen Email werden ermittelt (Zeile 9 und 11). In der Test-Email befindet sich ein Bild im Anhang. In diesem Fall ist dies der zweite Teil der Email. In Zeile 14 wird das Bild mit der Funktion `imap_fetchbody` aus der Email extrahiert und angezeigt. Die Variable `$inhalt` (Zeile 14) enthält das Bild in Base64-Kodierung. Der Inhalt dieser Variable könnte bspw. mit dem Absender der Mail in eine Datenbank geschrieben werden, um ihn in anderer Form wieder zu verwenden.

Weitere denkbare Alternativen, um allen mobilen Usern den Bildupload zu ermöglichen, ist die Integration von Fotoalben anderer Anbieter. Flickr bspw. bietet für das iPhone eine native Applikation, von der aus der User Bilder direkt in seinen Account laden kann. Für alle anderen internetfähigen mobilen Geräte bietet Flickr die Möglichkeit, an eine userbezogene Adresse eine Email mit dem Bild als Anhang zu senden. Diese wird automatisch dem jeweiligen Account zugeordnet. (vgl. Flickr® für Mobiltelefone 2010)

Um diese Bilder in die eigene Webapplikation zu integrieren, könnte der User seinen Flickr-Account hinterlegen. Anschließend kann ihm mit Hilfe der Flickr-API und eines Skriptes die Möglichkeit gegeben werden Bilder auszuwählen, die für die eigene Webanwendung von Interesse sind.

## 4.2. Ein Video einbinden

Am Desktop-Computer kann Flash momentan als defacto-Standard zum Einbinden von Videos betrachtet werden. Wie bereits in Kapitel 3.3 geschildert, ist Flash auf mobilen Geräten nicht verlässlich vorhanden. Um ein Video auf allen vier Testgeräten über eine Webanwendung anzeigen zu können, ist eine Unterscheidung der Geräte notwendig. Auf Basis von HTML, PHP und JavaScript wird in Listing 2 ein möglicher Umgang mit Videoeinbettungen aufgezeigt.

```

1 <!DOCTYPE html>
2 <html>
3 <?php
4 $user_agent=$_SERVER['HTTP_USER_AGENT'];
5 $video=true;
6 $maemo=false;
7 // Mobiles Gerät erkennen
8 if(preg_match('/maemo/i',$user_agent) || preg_match('/(iris|3g-t|windows ce|opera mobi|windows ce;
  smartphone;|windows ce; iemobile)/i',$user_agent)){
9     $video=false;
10    if(preg_match('/maemo/i',$user_agent)){
11        $maemo = true;
12    }
13 }??
14 <head>
15     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
16     <meta name="viewport" content="width=device-width" />
17     <title>Video abspielen</title>
18     <script>
19         function playVideo(){
20             var myVideo=document.getElementsByTagName('video')[0];
21             myVideo.play();
22         }
23     </script>
24 </head>
25 <body>
26 <? if($video){ ??
27 <!-- iPhone & Android -->
28     
29     <div id="iPAnd">
30         <video src="video/1101_250_baselineprofile.mp4" width=1 height=1></video>
31     </div>
32 <? }else if($maemo){ ??
33 <!-- Nokia Maemo -->
34     <div id="Maemo">
35         <object>
36             <param name="movie" value="player-viral.swf" />
37             <param name="allowscriptaccess" value="always" />
38             <param name="flashvars" value="file=video/1101.flv&image=img/1101.jpg" />
39             <embed type="application/x-shockwave-flash" src="player-viral.swf"
  allowscriptaccess="always" allowfullscreen="true" flashvars=
  "file=video/1101.flv&image=img/1101.jpg" width="360" height="288"
40             />
41         </object>
42     </div>
43 <? }else{ ??
44 <!-- Opera Mobile / Windows -->
45     <div id="Win">
46         <embed src="video/1101.wmv" width="360" height="288"/>
47     </div>
48 <? } ??
49 </body>
50 </html>

```

Listing 2: Code zum Einbinden von Videos

Das Video wird in drei Formaten benötigt:

- iPhone & HTC Legend: H.264 kodierte MP4 im Baseline-Profile
- Nokia N900: H.246 kodierte FLV
- Sony Ericsson XPERIA X1: WMV

Mit Hilfe von `$_SERVER` wird der Useragent ausgelesen (Zeile 3). Über die Funktion `preg_match` werden bestimmte Fälle überprüft und herausgefunden, um welches der vier Testgeräte es sich handelt. Je nach aufrufendem Gerät werden die beiden flag-Variablen `$video` und `$maemo` gesetzt. In Abhängigkeit von diesen booleschen Variablen werden für Android und iPhone die Zeilen 28 bis 31 angezeigt. An dieser Stelle wird der HTML5 `video`-Tag zur Einbettung des Videos genutzt. Da der Android Browser das `poster`-Attribut im `video`-Tag ignoriert, wird ein Workaround über JavaScript integriert (Zeile 19 bis 22), der das Video beim Berühren des Bildes abspielt. Mit dem `poster`-Attribut ist es möglich, ein Vorschau-Bild zu integrieren.

## 4. Lösungsansätze in der Praxis

Im Falle des Nokia N900 / Maemo werden die Zeilen 34 bis 42 angezeigt. Im Maemo-Browser wird ein Flash-Player verwendet, der mit den Tags `object` und `embed` eingebunden wird. Zu beachten ist hierbei, dass der Maemo-Browser nicht den `object`-Tag sondern den `embed`-Tag interpretiert. Da der `embed`-Tag jedoch nicht zum HTML-Standard gehört, ist es performanter den `object`-Tag mitzuliefern.

Wird diese Seite mit dem Opera Mobile auf dem Sony Ericsson XPERIA X1 aufgerufen, werden die Zeile 45 bis 47 ausgeführt. An dieser Stelle wird das WMV-Video mittels `embed`-Tag eingebunden.

### 4.3. Der Login

Als Beispiel zur Lösung des Eingabeproblems wird eine Login-Maske erstellt. Diese basiert auf Autovervollständigung mit Hilfe der Geolocation-API. Zur Verwendung kommen HTML, CSS, JavaScript und PHP. Zusätzlich wird die jQuery-Library verwendet (jQuery: The Write Less, Do More, JavaScript Library 2010).

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1,
6       maximum-scale=1" />
7     <title>Login</title>
8     <script type="text/javascript" src="jquery.js"></script>
9     <script type="text/javascript" src="script.js"></script>
10    <link rel="stylesheet" href="style.css" type="text/css" />
11  </head>
12  <body onLoad="init()">
13    <div>
14      <div class="left">Username: </div>
15      <div class="right">
16        <input name="user" id="user" type="text" placeholder="Username" onKeyUp="vorschlag()"/>
17        <div id="output" style="display:none;"></div>
18      </div>
19    </div>
20
21    <div class="clear"></div>
22
23    <div>
24      <div class="left">Passwort: </div>
25      <div class="right">
26        <input name="pass" id="pass" type="password" placeholder="Passwort"/>
27      </div>
28    </div>
29
30    <div class="clear"></div>
31
32    <div id="login" onClick="login()">
33      Login
34    </div>
35    <div id="geo" style="display:none;">
36      <input type="checkbox" class="check" id="auto" name="auto" checked /> Uservorschläge via
37      Location
38    </div>
39    <input type="checkbox" class="check" id="maske" name="maske" onChange="demaskieren()" />
40    Passwort im Klartext
41  </body>
42 </html>
```

Listing 3: index.html

Die Datei „index.html“ enthält das HTML-Gerüst: zwei Input-Felder (Username und Passwort), ein Login-Button sowie zwei Checkboxes (siehe Abb. 29). Das Feld „Username“ ist vom Typ `text` und das Feld „Passwort“ standardmäßig vom Typ `password`. Die Checkbox „Uservorschläge via Location“ ist aktiviert.

Username:

Passwort:

Uservorschläge via Location

Passwort im Klartext

Abbildung 29: Login-Maske

#### 4. Lösungsansätze in der Praxis

```
1 var geo=navigator.geolocation;
2 var lat, long, text;
3 var xmlhttp = new XMLHttpRequest();
4
5 // Initialisieren, wenn Geolocation da, dann Checkbox
6 function init(){
7     if(!geo){
8         document.getElementById("geo").style.display="none";
9     }else{
10        document.getElementById("geo").style.display="";
11    }
12 }
13 // Bei Checkbox-Klick, dann Typ des Input-Feld Passwort ändern
14 function demaskieren(){
15     if(document.getElementById("maske").checked){
16         document.getElementById("pass").type="text";
17     }else{
18         document.getElementById("pass").type="password";
19     }
20 }
21 // Text wird in Username-Feld eingegeben --> Hole Geo-Position
22 function vorschlag(){
23     if(document.getElementById("auto").checked){
24         text = document.getElementById("user").value;
25         geo.getCurrentPosition(successCallback, errorCallback, {enableHighAccuracy: true});
26     }
27 }
28 // Geo-Position - Erfolg: Hole XML-Datei mit entsprechenden Usern
29 function successCallback(position){
30     lat = position.coords.latitude;
31     long = position.coords.longitude;
32     xmlhttp.open('GET', 'user.php?text='+text+'&lat='+lat+'&long='+long, true);
33     xmlhttp.onreadystatechange = function(){
34         if (xmlhttp.readyState == 4){
35             darstellen(xmlhttp.responseXML);
36         }
37     };
38     xmlhttp.send(null);
39 }
40 // Geo-Position - Fehler
41 function errorCallback(error){
42     alert("Fehler passiert: "+error.message);
43 }
44 // Darstellen des XML
45 function darstellen(dom){
46     var output="<div id='vorschlag'>";
47     $("line", dom).each(function(i){
48         var user = $("user", this).text();
49         output += "<div class='user' onClick='getAuswahl(\""+user+"\")>"+user+"</div>";
50     });
51     output+="</div>";
52     $("#output").html(output);
53     document.getElementById("output").style.display="";
54 }
55 // Bei Klick auf Username-Vorschlag
56 function getAuswahl(user){
57     document.getElementById("user").value = user;
58     document.getElementById("output").style.display="none";
59 }
60 // Bei Klick auf Login
61 function login(){
62     alert("Sie werden eingeloggt!");
63 }
```

Listing 4: script.js

Gibt der User einen Buchstaben in das Feld „Username“ ein, so wird die JavaScript-Funktion `vorschlag` (Listing 4, Zeile 22 bis 27) aufgerufen. Wenn die Checkbox „Uservorschläge via Location“ aktiviert ist, wird der Text aus dem Feld „Username“ ausgelesen und die aktuellen Geo-Koordinaten (Latitude `lat` und Longitude `long`) abgerufen. Mittels `XMLHttpRequest`-Objekt und einem Get-Request werden der eingegebene Text, Latitude und Longitude an die Datei „user.php“ (Listing 5) gesendet.

An dieser Stelle werden die gesendeten GET-Parameter (Zeile 3 bis 5) ausgelesen. Das Array `$userArray` ersetzt in diesem Prototyp die Datenbank, welche die userbezogenen Daten speichert. Es erfolgt eine Überprüfung mit der Funktion `preg_match`, ob der eingegebene Text im Usernamen vorkommt. Ist dies der Fall, wird ermittelt, ob die zum gefundenen Usernamen gehörige Geo-Position mit der aktuellen Position des Users übereinstimmt. Im positiven Fall wird der Username in eine XML-Datei geschrieben und ausgegeben.

#### 4. Lösungsansätze in der Praxis

Die XML-Datei wird in der Funktion `darstellen` (Listing 4, Zeile 45 bis 54) für die HTML-Anzeige aufbereitet und dargestellt (siehe Abb. 30).

Username:

Passwort:

Login

Uservorschläge via Location

Passwort im Klartext

Abbildung 30: Vorschläge nach Eingabe von "K"

Wird der angezeigte Username ausgewählt, wird er durch die Funktion `getAuswahl` in das Feld „Username“ übernommen (Listing 4, Zeile 56 bis 59).

Gibt der User das Passwort ein, werden die Zeichen maskiert. Wird die Checkbox „Passwort im Klartext“ aktiviert, so werden die Zeichen im Klartext angezeigt (siehe Abb. 31 und 32). Realisiert wird dies über die JavaScript-Funktion `demaskieren` (Listing 4, Zeile 14 bis 20), die den Typ des Input-Feldes ändert.

Username:

Passwort:

Login

Uservorschläge via Location

Passwort im Klartext

Abbildung 31 : Auswahl "kerstin" / Passwordeingabe

Username:

Passwort:

Login

Uservorschläge via Location

Passwort im Klartext

Abbildung 32: Checkboxauswahl Passwort im Klartext

```
1 <?php
2 // Get-Parameter auslesen
3 $text = $_GET['text'];
4 $lat = $_GET['lat'];
5 $long = $_GET['long'];
6
7 $userArray = array(array(kerstin,48.2, 15.6), array(normalo,48.2,15.6), array(wersonst,
8 47.2,15.6), array(kerstin2,48.3, 15.6),array(kerstin3,48.2, 15.7), array(kerstin4,48.2,
9 15.6), array(kerstin5,48.2, 15.5));
10
11 $count = 0;
12 $xml2 = "";
13
14 // eingegebenen Text mit User & Location im Array vergleichen
15 for($i=0;$i<=4;$i++){
16     if(preg_match('/'.$text.'/i',$userArray[$i][0])){
17         if(round($lat, 1) == round($userArray[$i][1], 1) && round($long, 1) == round(
18 $userArray[$i][2], 1)){
19             $xml2 .= "<line><user>".$userArray[$i][0]."</user></line>";
20             $count++;
21         }
22     }
23 }
24 $xml1 = "<all count='".$count."'>".$xml2."</all>";
25 header ("Content-type:text/xml");
26 echo $xml1;
27 ?>
```

Listing 5: user.php

#### 4. Lösungsansätze in der Praxis

---

Beim Laden der HTML-Seite wird überprüft, ob der Browser die Geolocation-API unterstützt. Ist dies nicht der Fall, wird die Checkbox „Usvorschläge via Location“ nicht angezeigt und somit auch keine Abfrage der Geo-Position durchgeführt.

Bei den Testbrowsern ist dies der Fall beim Opera Mobile. Die anderen drei Browser (Mobile Safari, Android Browser und Maemo Browser) unterstützen die Geolocation-API.

```
1  input{
2      width: 100%;
3      font-size:medium;
4  }
5  body{
6      width:95%;
7      font-size:large;
8      font-family:Arial, Helvetica, sans-serif;
9  }
10 .user{
11     border:1px solid black;
12     padding: 0.2em;
13     width:100%;
14     background-color:#CC9;
15 }
16 .left{
17     float:left;
18     width:45%;
19     padding-top:0.3em;
20     text-align:right;
21     padding-right:0.5em;
22 }
23 .right{
24     float:left;
25     width:45%;
26     padding-right: 0.5em
27 }
28 .clear{
29     clear:both;
30 }
31 .check{
32     width:6%;
33 }
34 #login{
35     display: block;
36     background-color:#CCC;
37     width: 88%;
38     padding: 1em;
39     margin-top:1em;
40     margin-bottom:1em;
41     border: 0.1em solid black;
42 }
43 #output{
44     position: absolute;
45     z-index:999;
46     width: 45%;
47     font-size:xx-large;
48 }
```

Listing 6: style.css

In Listing 6 ist zu sehen, dass im verwendeten Stylesheet keine fixen Angaben für Position, Größe, Schriftgröße oder Abstand genutzt wurden.

## 5. Die zukünftige Bedeutung des mobilen Internets

### 5.1. Ergebnisse der Arbeit

Dieser Arbeit liegen folgende Forschungsfragen zu Grunde:

*Welche Probleme treten bei der Verwendung des Internets auf Mobiltelefonen auf?*

*Welche Strategien und Techniken existieren, um diesen Problemfeldern optimal zu begegnen?*

Aus der Bearbeitung der Fragen entstanden Best Practices für vier Problemfelder. Aus diesen lassen sich folgende Empfehlungen ableiten:

#### Für das Eingabeproblem

- Usereingaben reduzieren und vereinfachen
- User-Authentifizierung weitgehend automatisieren
- Größe der klickbaren Elemente benutzerfreundlich gestalten

#### Für das Darstellungsproblem

- Viewport auf eigentliche Screengröße festsetzen
- Position, Größe und Abstand von Elementen sowie Schriftgrößen nicht absolut definieren
- Ein-Spalten-Layout verwenden
- auf starken Farbkontrast achten (vor allem in Bezug auf unterschiedliche Lichtverhältnisse)

#### Für das Vollständigkeitsproblem

- 80/20 Regel: 80% der Inhalte einer Desktop-Website sind für den mobilen User nicht relevant
- einfache Seitenarchitektur
- Balance zwischen großer Anzahl an Links und Notwendigkeit durch Vielzahl von Links navigieren zu müssen
- Relevanz der Information im Seitenlayout berücksichtigen
- Link zur Desktop-Version anbieten

#### Für das Limitationsproblem

- nicht alle Geräte unterstützen Flash – zukünftige Alternative: HTML5
- nicht alle Geräte lassen Dateiapload zu – Alternative: Upload via Email
- Datenübertragungen reduzieren

Es ist zu beachten, dass mit jeder neuen Browser-Version diese Empfehlungen auf den Prüfstand gestellt werden müssen. Des Weiteren wurde lediglich auf vier Smartphones getestet, welche die gängigsten Browser nutzen. Andere Testgeräte hätten möglicherweise ein anderes Verhalten gezeigt.

### 5.2. Das mobile Internet in der Zukunft

Wie bereits zu Beginn dieser Arbeit erwähnt, misst der Analyst Gartner dem mobilen Web eine große Bedeutung für 2010 und 2011 bei (vgl. Gartner Outlines 10 Mobile Technologies to Watch in 2010 and 2011 2010). Auch die Tatsache, dass weltweit mehr internetfähige mobile Endgeräte im Einsatz sind als Desktop-Rechner (vgl. Bieh 2008, 159f) stützt diese Prognose.

Sollte das mobile Internet separat betrachtet werden oder sollten die mobilen Endgeräte bei der Erstellung EINER Webapplikation für Desktop und mobil mit Beachtung finden? Niklas Bürger ruft auf „mobile zeitgeist“ auf „[...] „Mobile“ nicht mehr separat zu behandeln, sondern fest integriert zu sehen – es gibt letztlich nur EIN Internet.“ (Kommt das Ende des “Mobile Internet”? | mobile zeitgeist 2010) Voraussetzung dafür ist allerdings, dass die verwendeten Browser sich in Sachen Funktionalität vor allem bezüglich HTML-Interpretation, HTML5 und Zugriff auf das Dateisystem annähern.



## 5. Die zukünftige Bedeutung des mobilen Internets

Wie verhält es sich mit den nativen Anwendungen? Das Problem des Dateiuploads ist bei nativen Applikationen nicht vorhanden. Bei dieser Art der Anwendung hat der Entwickler die Rechte auf Dateien zuzugreifen. Auch für das Eingabeproblem bieten sich bei nativen Applikationen Alternativen, wie bspw. die Spracheingabe oder die Eingabe per Kamera.

Hier stellt sich also die Frage, warum überhaupt Webanwendungen entwickeln und nicht nur native? Die wichtigste Antwort auf dieser Frage ist sicherlich der Kostenfaktor. Die Kosten für jede Plattform eine native Applikation zu entwickeln sind sicherlich höher als die Kosten für eine Webanwendung, die möglichst viele Browser unterstützt.

Als Alternative bieten sich an dieser Stelle hybride Applikationen an. Sie sind eine Kombination aus webbasierter und nativer Applikation. Hybride Applikationen werden durch eine Browser-View realisiert und können auf Funktionen des Gerätes wie bspw. das Dateisystem zurückgreifen.

„Der mobile Zugang zum Web macht noch keine Revolution aus, es sind die Funktionen, die über die reine Webnutzung hinausgehen, die eine Revolution ausmachen könnten.“

(Alby 2008, 47)

## Verzeichnisse

### Quellen

Adobe - Flash Lite. 2010. Available at: <http://www.adobe.com/products/flashlite/> [Zugegriffen Mai 14, 2010].

Adobe Labs - Adobe Flash Player 10.1. 2010. Available at: <http://labs.adobe.com/technologies/flashplayer10/> [Zugegriffen Mai 14, 2010].

Alby, Tom. 2008. Das mobile Web. 1. Aufl. München: Hanser Carl.

aon Breitband-Duo. 2009. Available at: [http://www.telekom.at/portal/site/telekom/menuitem.dacebe59cof5ab5c5cf20310d3a041ca/?vgnnextoid=6bfa159f5b2a0210VgnVCM2000004f1a1facRCRD&vgnnextchannel=4c1da80220d2c110VgnVCM100000230a14ac\\_\\_\\_\\_&taxlang=de](http://www.telekom.at/portal/site/telekom/menuitem.dacebe59cof5ab5c5cf20310d3a041ca/?vgnnextoid=6bfa159f5b2a0210VgnVCM2000004f1a1facRCRD&vgnnextchannel=4c1da80220d2c110VgnVCM100000230a14ac____&taxlang=de) [Zugegriffen Mai 15, 2010].

Apple. 2010. Available at: <http://www.apple.com/at/> [Zugegriffen Mai 8, 2010].

Apple - iPhone - Mobiltelefon, iPod und Internetgerät. 2010. Apple. Available at: <http://www.apple.com/at/iphone/> [Zugegriffen Mai 8, 2010].

Apple - iPhone - Technische Daten. 2010. Apple. Available at: <http://www.apple.com/at/iphone/specs.html> [Zugegriffen Mai 8, 2010].

Apple iPhone 3GS - Technische Daten - Datenblatt. 2010. 4phones.de. Available at: <http://www.4phones.de/technische-daten/apple/iphone-3gs/1322.html> [Zugegriffen Mai 8, 2010].

Bieh, Manuel. 2008. Mobiles Webdesign : Konzeption, Gestaltung, Entwicklung [Struktur, Design und Programmierung Umsetzung mit (X)HTML, CSS und PHP Standards und Best Practices]. 1. Aufl. Bonn: Galileo Press.

bob datenpaket 1gb. 2010. bob.at. Available at: <http://www.bob.at/bobdatenpaket1gb> [Zugegriffen Mai 8, 2010].

Boehringer, Joachim, Peter Buehler, und Patrick Schlaich. 2003. Kompendium der Mediengestaltung für Digital- und Printmedien. 2. Aufl. Berlin [u.a.]: Springer.

Carbonell, Noëlle, und European Research Consortium for Informatics and Mathematics. 2003. Universal access : theoretical perspectives, practice, and experience : 7th ERCIM International Workshop on User Interfaces for All, Paris, France, October 24-25, 2002 : revised papers. Berlin ; New York: Springer.

Datenschutz bei Facebook: Wie die Privatsphäre erodiert - SPIEGEL ONLINE - Nachrichten - Netzwelt. 2010. Available at: <http://m.spiegel.de/article.do?id=694388> [Zugegriffen Mai 14, 2010].

derStandard.at. 2010. Available at: <http://derstandard.at/> [Zugegriffen Mai 14, 2010].

derStandard.at Mobile. 2010. Available at: <http://mobil.derstandard.at/> [Zugegriffen Mai 14, 2010].

Device Central. 2006. Available at: [http://help.adobe.com/de\\_DE/DeviceCentral/1.1/help.html?content=WSA289C74B-7575-4546-B800-4511B7794D5A.html](http://help.adobe.com/de_DE/DeviceCentral/1.1/help.html?content=WSA289C74B-7575-4546-B800-4511B7794D5A.html) [Zugegriffen Mai 16, 2010].

FAQ - WHATWG Wiki. 2010. Available at: [http://wiki.whatwg.org/wiki/FAQ#When\\_will\\_HTML5\\_be\\_finished.3F](http://wiki.whatwg.org/wiki/FAQ#When_will_HTML5_be_finished.3F) [Zugegriffen Mai 15, 2010].

File API. 2009. Available at: <http://www.w3.org/TR/2009/WD-FileAPI-20091117/> [Zugegriffen Mai 14, 2010].

Fingerstift: HTC meldet Stylus für kapazitive Displays zum Patent an - PC-WELT. 2009. PC-WELT. Available at: [http://www.pcwelt.de/start/mobility\\_handy\\_pda/pda\\_smartphone/news/2101068/htc\\_meldet\\_stylus\\_fuer\\_kapazitive\\_displays\\_zum\\_patent\\_an/](http://www.pcwelt.de/start/mobility_handy_pda/pda_smartphone/news/2101068/htc_meldet_stylus_fuer_kapazitive_displays_zum_patent_an/) [Zugegriffen Mai 8, 2010].

Firtman, Maximiliano. 2010. Programming the mobile web. Farnham: O'Reilly.

Flickr® für Mobiltelefone. 2010. Available at: <http://de.mobile.yahoo.com/flickr> [Zugegriffen Mai 22, 2010].

Fling, Brian. 2009. Mobile design and development. 1. Aufl. Beijing; Sebastopol Calif.: O'Reilly.

Gartner Outlines 10 Mobile Technologies to Watch in 2010 and 2011. 2010. Gartner Technology Business Research Insight. Available at: <http://www.gartner.com/it/page.jsp?id=1328113> [Zugegriffen Mai 8, 2010].

Google Goggles for Android. 2010. Google. Available at: <http://www.google.com/mobile/goggles/#landmark> [Zugegriffen Mai 8, 2010].

Google Mobile App for your phone. 2010. Google. Available at: <http://www.google.com/mobile/google-mobile-app/> [Zugegriffen Mai 8, 2010].

High Capacity Color Barcode Technology - Microsoft Research. 2010. Microsoft Research. Available at: <http://research.microsoft.com/en-us/projects/hccb/about.aspx> [Zugegriffen Mai 8, 2010].

Hoggan, Eve, Stephen A. Brewster, und Jody Johnston. 2008. "Investigating the effectiveness of tactile feedback for mobile touchscreens." In Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI ,08, Florence, Italy, p. 1573. Available at: <http://portal.acm.org/citation.cfm?doid=1357054.1357300>.

HTC Legend - Technische Daten. 2010. HTC. Available at: [http://www.htc.com/de/specification.aspx?p\\_id=313](http://www.htc.com/de/specification.aspx?p_id=313) [Zugegriffen Mai 8, 2010].

HTC Legend - Technische Daten - Datenblatt. 2010. 4phones.de. Available at: <http://www.4phones.de/technische-daten/htc/legend/1466.html> [Zugegriffen Mai 8, 2010].

HTML5. 2010. World Wide Web Consortium (W3C). Available at: <http://dev.w3.org/html5/spec/Overview.html> [Zugegriffen Mai 8, 2010].

iConji. Connecting the world. 2010. Available at: <http://www.iconji.com/about/> [Zugegriffen Mai 16, 2010].

Image Exchange - Free Image Hosting & Photo Sharing. 2010. Available at: <http://www.imgxc.com/mobile.php> [Zugegriffen Mai 15, 2010].

jQuery: The Write Less, Do More, JavaScript Library. 2010. Available at: <http://jquery.com/> [Zugegriffen Mai 23, 2010].

Kaikkonen, Anne. 2008. "Full or tailored mobile web- where and how do people browse on their mobiles?." In Proceedings of the International Conference on Mobile Technology, Applications, and Systems - Mobility ,08, Yilan, Taiwan, p. 1. Available at: <http://portal.acm.org/citation.cfm?doid=1506270.1506307>.

Karlson, Amy, Benjamin Bederson, und Jose Contreras-Vidal. 2006. "Understanding Single-Handed Mobile Device Interaction." Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.8287&rep=rep1&type=pdf>.

Kommt das Ende des "Mobile Internet"? | mobile zeitgeist. 2010. Available at: <http://www.mobile-zeitgeist.com/2010/04/21/kommt-das-ende-des-mobile-internet/> [Zugegriffen Mai 23, 2010].

KONSUMENT.AT - Mobiles Internet. 2010. Available at: <http://www.konsument.at/cs/Satellite?pagename=Konsument%2FMagazinArtikel%2FDetail&cid=318869429429> [Zugegriffen Mai 15, 2010].

Lee, Seungyon, und Shumin Zhai. 2009. "The performance of touch screen soft buttons." In Proceedings of the 27th international conference on Human factors in computing systems - CHI ,09, Boston, MA, USA, p. 309. Available at: <http://portal.acm.org/citation.cfm?doid=1518701.1518750>.

Mobile Web Application Best Practices. 2010. World Wide Web Consortium (W3C). Available at: <http://www.w3.org/TR/2010/CR-mwabp-20100211/> [Zugegriffen Mai 8, 2010].

Mobiles Breitband-Internet: Langsam und teuer - Mobilfunkler - derStandard.at > Web. 2010. Available at: <http://derstandard.at/1269448154644/Test-Mobiles-Breitband-Internet-Langsam-und-teuer> [Zugegriffen Mai 15, 2010].

Nokia Deutschland - Nokia N900 - Technische Daten. 2010. Nokia Deutschland. Available at: <http://www.nokia.de/produkte/mobiltelefone/nokia-n900/technische-daten> [Zugegriffen Mai 8, 2010].

Nokia N900 - Technische Daten - Datenblatt. 2010. 4phones.de. Available at: <http://www.4phones.de/technische-daten/nokia/n900/1381.html> [Zugegriffen Mai 8, 2010].

Nokia Schweiz - Nokia 1110i - Produkte. 2010. Nokia Schweiz. Available at: <http://www.nokia.ch/produkte/alle-modelle/nokia-1110i> [Zugegriffen Mai 8, 2010].

NYT Mobile. 2010. The New York Times. Available at: <http://mobile.nytimes.com/main?edz=us> [Zugegriffen Mai 8, 2010].

Parhi, Pekka, Amy K. Karlson, und Benjamin B. Bederson. 2006. "Target size study for one-handed thumb use on small touchscreen devices." In Proceedings of the 8th conference on Human-computer interaction with mobile devices and services - MobileHCI ,06, Helsinki, Finland, p. 203. Available at: <http://portal.acm.org/citation.cfm?doid=1152215.1152260>.

Passani, Luca. 2010. "WURFL." Available at: <http://wurfl.sourceforge.net/> [Zugegriffen Mai 15, 2010].

Perry, Keith, und Juan Hourcade. 2008. "Evaluating one handed thumb tapping on mobile touchscreen devices." In Proceedings of graphics interface 2008, Windsor, Ontario, Canada: Canadian Information Process Society; Canadian Human-Computer Communications Society. Available at: <http://portal.acm.org/citation.cfm?id=1375714.1375725&coll=ACM&dl=ACM&CFID=89546066&CFTOKEN=89845430> [Zugegriffen Mai 8, 2010].

PHP: imap\_search - Manual. 2010. Available at: <http://www.php.net/manual/de/function.imap-search.php> [Zugegriffen Mai 22, 2010].

README - iphone-photo-picker - Project Hosting on Google Code. 2009. Available at: <http://code.google.com/p/iphone-photo-picker/wiki/README> [Zugegriffen Mai 15, 2010].

Schmiedl, Grischa. 2010. "Probleme des mobilen Internets." Experteninterview.

Schmiedl, Grischa u. a. 2009. "Verwendbarkeit und Verwendung des mobilen Webs." Available at: [http://medieninformatik.fh-stpoelten.ac.at/index.php?option=com\\_content&task=view&id=185&Itemid=10](http://medieninformatik.fh-stpoelten.ac.at/index.php?option=com_content&task=view&id=185&Itemid=10).

Schmiedl, Grischa, Markus Seidl, und Klaus Temper. 2010. "The Mobile Web Study - Usage, Usability, Fallacies, Pitfalls." In 4. Forschungsforum der österreichischen Hochschulen, Pinkafeld, Eisenstadt: Fachhochschulstudiengänge Burgenland GmbH, p. 555.

Schneider, Tobias. 2010. "Home - gordon - GitHub." Available at: <http://wiki.github.com/tobeytailor/gordon/> [Zugegriffen Mai 15, 2010].

SELFHTML: Stylesheets / CSS-Formate definieren / Maßeinheiten, Farbangaben und Wertzuweisung. 2007. SELFHTML. Available at: <http://de.selfhtml.org/css/formate/wertzuweisung.htm#numerische> [Zugegriffen Mai 9, 2010].

Sony Ericsson - Products - Mobile phones - Overview - K800i. 2010. Sony Ericsson. Available at: <http://www.sonyericsson.com/cws/products/mobilephones/overview/k800i?cc=de&lc=de> [Zugegriffen Mai 8, 2010].

Sony Ericsson XPERIA X1 - Technische Daten - Datenblatt. 2010. 4phones.de. Available at: <http://www.4phones.de/technische-daten/sonyericsson/xperia-x1/1011.html> [Zugegriffen Mai 8, 2010].

STATISTIK AUSTRIA - IKT-Einsatz in Haushalten. 2008. Available at: [http://www.statistik.at/web\\_de/dynamic/statistiken/informationsgesellschaft/ikt-einsatz\\_in\\_haushalten/publdetail?id=305&listid=305&detail=499](http://www.statistik.at/web_de/dynamic/statistiken/informationsgesellschaft/ikt-einsatz_in_haushalten/publdetail?id=305&listid=305&detail=499) [Zugegriffen Mai 8, 2010].

The New York Times - Breaking News, World News & Multimedia. 2010. Available at: <http://www.nytimes.com/> [Zugegriffen Mai 8, 2010].

Top 9 Mobile Browsers in Austria from May 09 to Apr 10 | StatCounter Global Stats. 2010. StatCounter Global Stats. Available at: [http://gs.statcounter.com/#mobile\\_browser-AT-monthly-200905-201004](http://gs.statcounter.com/#mobile_browser-AT-monthly-200905-201004) [Zugegriffen Mai 8, 2010].

Tote bei Straßenschlachten in Bangkok - Thailand - derStandard.at › International. 2010. Available at: <http://derstandard.at/1271376633950/Tote--bei-Strassenschlachten-in-Bangkok> [Zugegriffen Mai 14, 2010].

Tote bei Straßenschlachten in Bangkok - Thailand - derStandard.at › International Mobile. 2010. Available at: <http://mobil.derstandard.at/1271376633950/Tote--bei-Strassenschlachten-in-Bangkok> [Zugegriffen Mai 14, 2010].

Vorgehensweise: Einrichten von Clientzertifikaten. 2004. MSDN: Microsoft Developer Network (Deutschland). Available at: <http://msdn.microsoft.com/de-de/library/cc431323.aspx> [Zugegriffen Mai 8, 2010].

Wigdor, Daniel u. a. 2007. "Lucid touch." In Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07, Newport, Rhode Island, USA, p. 269. Available at: <http://portal.acm.org/citation.cfm?doid=1294211.1294259>.

XHTML 1.1 - Module-based XHTML. 2010. World Wide Web Consortium (W3C). Available at: <http://www.w3.org/TR/2001/REC-xhtml11-20010531/> [Zugegriffen Mai 8, 2010].

### 3. Probleme bei der Verwendung des mobilen Internets

#### Abbildungen

Abbildung 1: Mobiltelefon - Nokia 1100 .....	3
Abbildung 2: Feature Phone - Sony Ericsson K800i.....	3
Abbildung 3: Smartphone - iPhone.....	4
Abbildung 4: Non-Phone Gerät - iPad.....	4
Abbildung 5: Internetzugriff mit mobilen Geräten .....	5
Abbildung 6: Top 9 der mobilen Browser von Mai 2009 bis April 2010 .....	7
Abbildung 7: iPhone Tastatur Hochformat .....	9
Abbildung 8: iPhone Tastatur Querformat.....	9
Abbildung 9: Nutzung von ein- oder zweihändiger Eingabe.....	10
Abbildung 10: <code>&lt;input type="password" /&gt;</code> im Mobile Safari .....	11
Abbildung 11: Verhalten von Mobile Safari beim <code>input</code> -Attribut <code>placeholder</code> .....	12
Abbildung 12: QR-Code der URL <code>http://medieninformatik.fhstp.ac.at</code> .....	13
Abbildung 13: 8farbiger Barcode .....	13
Abbildung 14: iConji Applikation.....	14
Abbildung 15: PPI-Problem – 72 Pixel im Verhältnis zu 1 Zoll .....	16
Abbildung 16: New York Times Desktop-Version .....	17
Abbildung 17: New York Times Mobil #1 .....	17
Abbildung 18: New York Times Mobil #2 .....	17
Abbildung 19: derStandard.at Desktop-Version.....	21
Abbildung 20: derStandard.at Desktop-Version Artikelseite.....	22
Abbildung 21: derStandard.at Mobil.....	22
Abbildung 22: derStandard Mobil Artikelseite .....	22
Abbildung 23: Downloadgeschwindigkeiten mobiler Internetverbindungen .....	25
Abbildung 24: Maemo Browser - <code>&lt;input type="file" /&gt;</code> .....	27
Abbildung 25: Mobile Safari - <code>&lt;input type="file" /&gt;</code> .....	27
Abbildung 26: <code>&lt;video&gt;</code> im Mobile Safari und Android Browser #1 .....	28
Abbildung 27: <code>&lt;video&gt;</code> im Mobile Safari und Android Browser #2 .....	28
Abbildung 28: <code>&lt;video&gt;</code> im Opera Mobile und Maemo Browser .....	28
Abbildung 29: Login-Maske .....	32
Abbildung 30: Vorschläge nach Eingabe von "K".....	34
Abbildung 31: Auswahl "kerstin" / Passwordeingabe .....	34
Abbildung 32: Checkboxauswahl Passwort im Klartext.....	34

#### Tabellen

Tabelle 1: Technische Daten der Testgeräte.....	8
Tabelle 2: Kompatibilität der Werte <code>tel:</code> , <code>mailto:</code> , <code>sms[to]:</code> , <code>mms[to]:</code> .....	13
Tabelle 3: Farbtiefen bei mobilen Geräten.....	15
Tabelle 4: Pixeldichte der Testgeräte.....	16
Tabelle 5: Font-Unterstützung bei mobilen Geräten.....	19
Tabelle 6: Darstellbare Titellänge der Browser.....	24
Tabelle 7: JavaScript-Unterstützung .....	26
Tabelle 8: Download-Kompatibilität .....	27

#### Listings

Listing 1: PHP-Skript zum Abrufen von Emails und darstellen des Bildanhanges .....	30
Listing 2: Code zum Einbinden von Videos.....	31
Listing 3: <code>index.html</code> .....	32
Listing 4: <code>script.js</code> .....	33
Listing 5: <code>user.php</code> .....	34
Listing 6: <code>style.css</code> .....	35