

Smartphone Anwendungen mit Apex

Dick Dral
Smart4Apex
Wageningen, Die Niederlande

Schlüsselworte

Apex, Smartphones, Spracherkennung

Einleitung

Smartphones sind heutzutage kleine Rechner, mit gleichen Leistungen wie Klein PC's vor einigen Jahren. Sie haben Farbschirme mit HD Auflösung. Jedoch gibt es deutliche Beschränkungen wenn man Smartphones benutzt. Der Bildschirm ist klein, die Verbindung zum Netz ist oft langsam und die Tastatur ist nicht einfach im Gebrauch.

Dieser Artikel beschreibt wie man umgehen kann mit diesen Beschränkungen.

Kleiner Bildschirm

Smartphones haben kleine Bildschirme. Die letzten Jahre sind sie gewachsen in Größe, aber 6 Zoll ist wahrscheinlich die obere Grenze. Größer passt nicht in der Tasche und man mag keinen Anruf machen mit einer Mini-Tablette zu sein Ohr.

Die kleinen Bildschirme sind nicht geeignet um normale Websites zu zeigen. Darum werden speziellere mobile Websites erstellt, die benutzt werden können auf einem Smartphone.

Apex unterstützt die Erstellung mobiler Anwendungen. Für diesen Zweck wird JQuery Mobile (JQM) benutzt. Obwohl JQM gemacht ist für Smartphones, wird der Raum des Bildschirms nicht effizient benutzt. In Forms wird das Label über das Item platziert, und wird der Form länger. Es gibt nicht mal Platz für fünf Elemente. Mit anderen CSS Definitionen kann man das Label neben dem Eingabeelement stellen und gibt es Raum für mindestens sechs Elemente mit größeren Buchstaben.

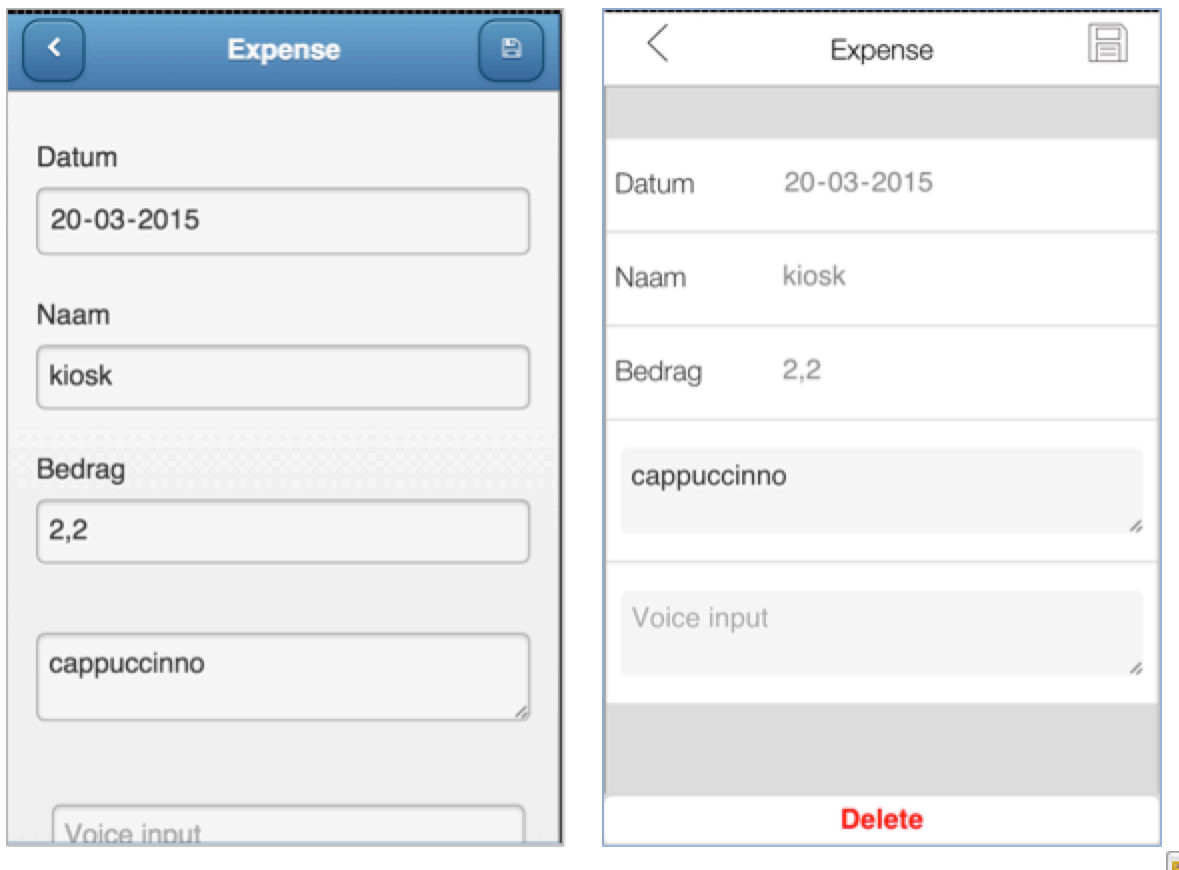


Abb. 1. Standard JQM versus angepasst

Der nächste HTML wird durch JQM für ein Formularelement erzeugt.

```
<div ... class="ui-field-contain">
  <label ...>Tekst</label>
  <input ... />
  
</div>
```

Es ist möglich, um das Layout mit CSS zu beeinflussen.

```
.ui-field-contain label {
  position: absolute;
  left: 5px;
  line-height: 34px;
}
```

```
.ui-field-contain input {  
    position: absolute;  
    left: 30%;  
    width: 68%;  
    line-height: 34px;  
}  
.ui-field-contain img {  
    position: absolute;  
    right: 10px;  
    top: 10px;  
}
```

Das Element **position: absolute** sorgt dafür, dass das Element mit der Klasse **ui-field-contain** eine absolute Position hat relativ zu der ersten Eltern mit **position** definiert. Die Attributen **left**, **right** und **top** definieren diese absolute Position. Indem Sie die Etiketten kurzen, gibt es mehr Platz für die Formularelemente. In diesem Fall besetzen die Etiketten 30% der Breite, so dass 70% übrig bleibt für die Formularelemente. Durch Festlegung der Breite der Eingabeelement auf 68% gibt es keine Überlappung mit dem Symbol an der rechten Seite.

Langsamer Verbindung

Bei Verwendung des Smartphones ist die Geschwindigkeit der Datenverbindung oft begrenzt. Vor allem bei der Wiedergabe eine neue Seite gibt es eine Menge an Netzwerkverkehr, sowohl in Größe und Anzahl der Anfragen. In der Praxis bedeutet dies, dass oft eine Aktualisierung der Seite zwischen 3 und 5 Sekunden dauern kann.

Es ist wichtig die Seitenwechsel so viel wie möglich zu vermeiden um eine brauchbare Anwendung zu bekommen.

Die erste und einfachste Weg ist in den Entwurf die Anzahl der benötigten Seitenwechseln zu verringern. Eine einfache Möglichkeit, die Anzahl der Seitenaktualisierungen zu reduzieren, ist ein Popup-Menü, das in jeder Seite aufgenommen ist anstelle einer separaten Menüseite verwenden.

Die Startseite von vielen Desktop-Anwendungen ist ein Report mit Schaltflächen oder Links zu Formen. Oft ist eine spezielle Form die meistgenutzte Seite der Anwendung. In diesem Fall könnte es eine gute Idee sein, um die Anwendung mit diesem Form statt dem Report zu beginnen. Im allgemeinen erfordert der Gestaltung eine Smartphone-Applikation mehr Aufmerksamkeit auf das Thema der Optimierung der Nutzung der Anwendung.

Auf der technischen Seite gibt es eine andere Möglichkeit, die Anzahl der Seitenaktualisierungen zu reduzieren. Dies geschieht durch alle Komponenten der Anwendung auf einer Seite zu versammeln, eine so genannte Single Page Application. Für eine Apex-Anwendung bedeutet dies, dass alle Regionen auf einem Apex-Seite definiert werden.

JavaScript wird verwendet um die Regionen zu eigen oder zu verstecken. Bei einer logischen Seitenwechsel sollte häufig Daten vom Server abgerufen werden. Dies kann mit Hilfe Dynamic Actions für Form und Partial

Page Renedering für Reports durchgeführt werden. Speichern von Daten wird auch mit Hilfe von Dynamic Actions ausgeführt. Zeigen von (Fehler-)Meldungen können mit JavaScript gestaltet werden.

Es sei klar, dass die Schaffung einer Single Page Application mit Apex viel mehr Aufwand erfordert als die Schaffung eines normalen Apex-Anwendung. Vor allem die JavaScript und Dynamic Actions brauchen viel mehr Programmierung. Doch der Geschwindigkeitsgewinn kann enorm sein, was den zusätzlichen Aufwand rechtfertigen kann.

Dateneinfuhr mit den Tastatur

Die Tastatur von einem Smartphone auf dem Bildschirm ist nicht geeignet für die Eingabe von großen Textmengen. Die Eingabe ist langsam und auch die Korrektur von Tippfehlern nimmt eine Menge Zeit. Persönlich mache ich jede 10 bis 20 Zeichen ein Tippfehler. Für eine effiziente Nutzung der Anwendung sollte die Notwendigkeit zur Verwendung der Tastatur, so viel wie möglich reduziert werden.

Es gibt mehrere Möglichkeiten, dies zu tun:

1. weglassen der Eingabeelementen
2. Vorfüllung
3. einfache Möglichkeit, um das Element zu löschen
4. Smart Items (Auto-Complete)
5. Eingabe mit Touch
6. Eingabe durch Sprache

Die einfachste Möglichkeit ist das Weglassen der Eingabeelemente. In Desktop-Anwendungen sind die Seiten oft mit Elemente, von denen viele nur gelegentlich verwendet werden. Überlegen Sie genau, welche Elemente der Benutzer eingeben soll. Welche Daten sind wirklich notwendig? Können die Daten abgeleitet werden? Sind die Daten wesentlich für die Anwendung?

Vorfüllen der Elemente kann bedeuten dass viel weniger Daten mit den Tastatur eingegeben werden sollen. Ein bekanntes Beispiel ist die Vorfüllung der Wert der heutigen Datum für das Datum in einem Zeiterfassungsanwendung. Auch das Projekt oder eine Aktivität kann mit der zuletzt eingegebenen Wert gefüllt werden. Bei der Verwendung von vorgefüllt Produkte ist es wichtig, um das Element einfach löschen zu können um eine anderen Wert einzugeben.

In meiner eigenen Zeit Registrierungs Anwendung kann ich die Zeilen von einem Tag auf ein anderes Datum kopieren. Manchmal ist eine Änderung erforderlich ist, aber oft können die Daten eingereicht werden, wie sie sind.

Zurück in der Arena: das autocomplete item

Das autocomplete item wird nicht für die mobile Entwicklung unterstützt. Es tritt in der Liste der Artikel wenn man **Show unsupported** wählt. Der Grund, warum es nicht unterstützt wird, ist unklar für mich, denn mit diesen Element kann eine Menge von Tastatureingaben vermieden werden.

In viele Anwendungen werden die gleichen Texte immer wieder benutzt, auch wenn sie Klartext-Spalten sind. Durch die Definition eines solchen Item als Auto-Complete Element und mit dem Inhalt der Spalte als List of Values, braucht der Benutzer nur ein paar Zeichen zu typen, bevor man eine kurze Auswahlliste bekommt. Wenn keiner dieser Texte gilt kann der Benutzer den Text vollständig einführen.

Das Styling des autocomplete Items ist nicht an die mobile Anwendung angepasst. Der Hintergrund der Liste ist transparent und die einzelne Zeilen sind zu klein um durch berühren zu wählen.

Mit dem folgenden CSS ist das Item in einer JQM Anwendung nutzbar.

```
.ac_results {
  background-color: white;
  border: 1px solid #cccccc;
}

.ac_results ul {
  margin-top: 0;
  padding-left: 5px;
}

.ac_results li {
  line-height: 30px;
  list-style: outside none none;
}
```

Zeiteingabe mit Touch

Seit ich meine erste Smartphone erhielt habe ich die Idee, etwas mit Zeiteingabe und einem analogen Uhr zu machen. Vor zwei Jahren hatte ich die Zeit und das Know-how, um diese Idee zu realisieren. Die Einfuhr ist auf das Zeichnen des Stunden- und Minutenzeiger an einem analogen Uhr auf dem Smartphone-Bildschirm basiert.

Ein dergleichen Element bietet ein Genauigkeit auf 5 Minuten, ausreichend für die meisten. Diese Methode der Zeiteinfuhr ist intuitiver und -nach einige praxis- auch schneller als die Verwendung der Tastatur.

Bild 2. zeigt das Element. Links sieht man eine Form, in der die Zeiteingabeelemente eine Uhr Bild enthalten. Nach einem Klick auf dieses Uhr Bild wird das Bild auf der rechten Seite angezeigt. In diesem Bild kann die Stunden- und Minutenzeiger aus der Mitte nach außen gezeichnet werden. Sie können nach AM / PM zu konvertieren mit den Tasten auf der Oberseite.

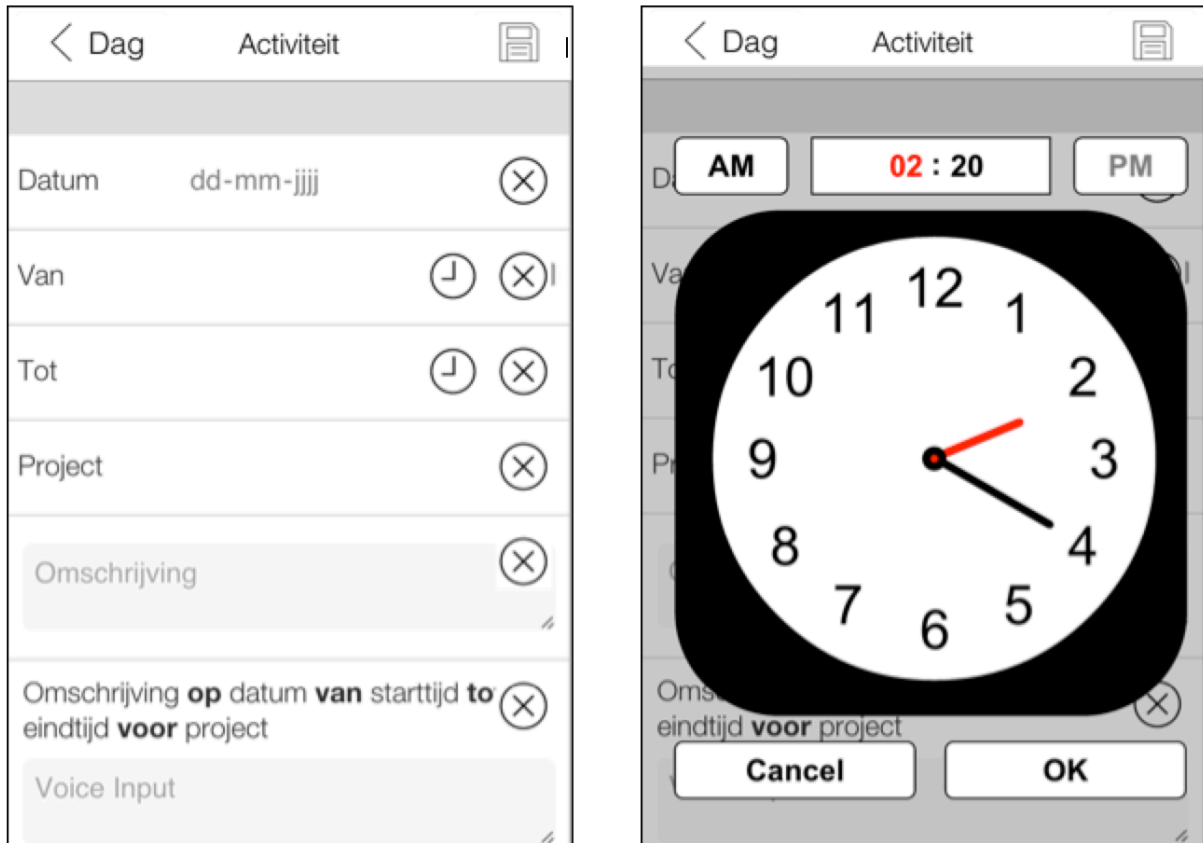


Abb. 2. Zeiteinfuhr mit Touch

Mehr Informationen über die Eingabe gibt es in einem Blog-Eintrag (dickdral.blogspot.nl/2015/02). Dieser Blogeintrag enthält auch einen Link zu einer Beispielanwendung, in dem man selbst diese Form der Eintrag kann erleben.

Dateneingabe mit Sprache

Für mich was eine der wichtigsten Verbesserungen in iOS8 die Verfügbarkeit von niederländischen Spracherkennung. Dies funktioniert überraschend gut. *Weil die Sprache auf einem Server interpretiert wird, ist eine Netzwerkverbindung benötigt für die Spracherkennung.*

Spracherkennung kann von der Tastatur eingeleitet werden. Auf diese Weise gibt es die Möglichkeit von Spracherkennung in jeder Gelegenheit, wo die Tastatur zur Verfügung steht. Also kann es auch für Texteingabe Elemente in Web-Anwendungen verwendet werden, so. *In Android steht auch die Spracherkennung auf ähnliche Weise durch die Tastatur zur Verfügung.*

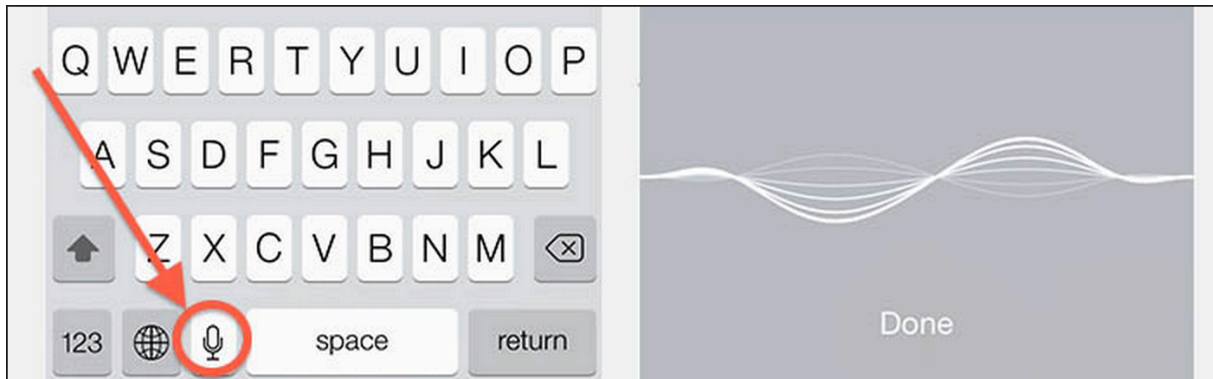


Abb. 3 Tastatur (links) und aktiver Spracherkennung (rechts) in iOS

Es ist also möglich, Daten mit Sprache in eine bestehende Apex-Anwendung einzugeben. Es ist jedoch sehr ineffizient, dies für jedes einzelnes Element zu tun. Jedesmal soll das Mikrofon aktiviert werden, danach wird der Text muss gesprochen werden, am Ende soll Fertig gedrückt werden und dann drückt man die Navigation zum nächsten Element. Dies ist nur sinnvoll, wenn viel Text in einem Element eingegeben werden muss.

Auf der technischen Seite brauchen wir auch mit Sprache eingegebene Daten und Zahlen zu Konvertieren nach einen Format der Apex versteht. Dass heisst verschiedenen andernungen an der Apex Seite sind nötig um die Seite geschickt zu machen für Spracheingabe.

Es ist jedoch wesentlich effizienter, um alle Daten in einem Satz einzugeben. Dafür brauchen wir nur um eine Spracheingabeelement hinzuzufügen von dem der Inhalt nach Eingabe verarbeitet wird.

Die Inhalte der verschiedenen Elementen kann mithilfe Stop-Wörter erkannt werden. Ein Sonderfall ist die Situation mit nur einem Datum in dem Satz. Für dieser Datumwert kann man eine relative Datumandeutung wie **gestern** benützen. Diese andeutung kann dann erkannt werden ohne die Notwendigkeit eines Stoppwort. Auch das erste Element benötigt keine Stoppwort, weil es von der ersten Stoppwort des nächsten Elements begrenzt wird. Dies wird im folgenden Beispiel erläutert.



Abb. 4. Beispiel Spracherkennung

Im obigen Beispiel wird der Satz **Kaffe und Kuchen gestern bei Kaufhaus für € 4,95** eingegeben. Das Wort **gestern** wird als Datumswert erfasst. Die Stoppwort **bei** markiert den Anfang des Namens, während die Stoppwortmarken **für** die Start der Inhalt der **Betrag** bedeutet.

Die Beschreibung wurde als das erste Element des Satzes erkannt, der durch den Datumswert **gestern** begrenzt wird.

So kommt der Prozess zum nächsten Ergebnis:

Element	Inhalt
Umschreibung	Kaffee und Kuchen
Datum	24-09-2015
Name	Kaufhaus
Betrag	4.95

Bemerk dass das Wort **Gestern** wird von einem Datum in das entsprechende Datumsformat ersetzt und dem Euro-Zeichen wird aus den Betrag entfernt.

Diese Werte werden in den entsprechenden Positionen platziert. Alle dass den Benutzer tun muss, ist prüfen, ob die Werte in den Form richtig sind und danach auf **Speichern** zu drücken.

In bestehenden Apex Anwendungen kann auf der beschriebenen Art und Weise mit wenig Aufwand Spracheingabe zur Verfügung gestellt werden.

Abschluss

Die Verwendung von (Web-) Anwendungen auf Smartphones hat seine Grenzen. Es gibt verschiedene Möglichkeiten, mit diesen Beschränkungen umzugehen.

Entwerfen Sie die Anwendung mit diesen Einschränkungen im Hinterkopf. So schaffen Sie die richtige Grundlage für eine nutzbare Anwendung. Die Anwendung sollte an den kleine Bildschirmgröße angepasst werden. CSS kann verwendet werden, um die Anordnung zu optimieren.

Erstellen einer Single Page Application kann eine Antwort auf niedrige Netzwerkgeschwindigkeiten sein. Obwohl es mehr Aufwand kostet, eine solche Anwendung zu erstellen, ist es die Mühe in einigen Fällen wert.

Es gibt eine Reihe von Möglichkeiten, um die Verwendung der Bildschirmtastatur zu vermeiden. Vorfüllung und Autocomplete verringern die Notwendigkeit, die Tastatur zu verwenden. Touch und Spracheingabe können die Tastatur ebenfalls teilweise ersetzen.

Sie können die beschriebenen Techniken verwenden, um die Benutzerfreundlichkeit von Apex Smartphone-Anwendungen zu verbessern.

Kontaktadresse:

Dick Dral

Smart4Apex

Eykmanstraat 26

6706JX Wageningen

Die Niederlande

Telefon: +31 (0) 6 16 36 76 21

E-Mail dick.dral@detora.nl

Internet: www.detora.nl

Blog: dickdral.blogspot.com