Mobile Web Applications using HTML5

L. Cotfas

14 Dec. 2011

Reasons for **mobile** web development

- **Many different platforms:** Android, IPhone, Symbian, Windows Phone/ Mobile, MeeGo… (only a few of them)

Reasons for **mobile** web development

Targeting more platforms with native apps

- Higher development and maintenance **costs**
- More **time** needed for development & test
- Few people have the necessary skills to develop using several platforms

Native applications

- **Advantages**
  - Can use all the features of the device
  - Better performance for “resource intensive” client operations
  - Can be sold in popular application stores like: Android, iPhone (also in Operator stores like Orange)

- **Disadvantages**
  - Targeting more platforms requires a lot of time and money
  - Slower development cycle
  - Difficult to find developers that know all the platforms.

Simple **Web** Applications

- **Advantages**
  - Can reach any mobile phone with a web browser
  - Can be easily and frequently updated

- **Disadvantages**
  - Simple user interface
  - Internet connection is constantly required
  - Can not access the features of the device.

Complex **Web** Applications

- **Characteristics**
  - Built using **HTML5, AJAX, CSS3**

- **Advantages**
  - Can reach any mobile phone with a **HTML5** web browser
  - Can access in a **standard** way (**W3C**) an increasing number of phone features: **GPS** (Geolocation API), **Accelerometer & Gyroscope** (DeviceOrientation API)
  - Can be converted to **~native** applications that can be sold through an application store
  - Can be easily and frequently updated
  - Advanced UI (comparable wit native applications)
  - Internet connection is not always required.

Complex **Web** Applications

- **Disadvantages**
  - Can not access all the features of the device **(but a growing number of standard APIs)**
  - Fragmentation - **WURFL** offers a DB with available features for each phone
  - Worse performance for “resource intensive” client operations
  - Storage space is limited to ~10Mb

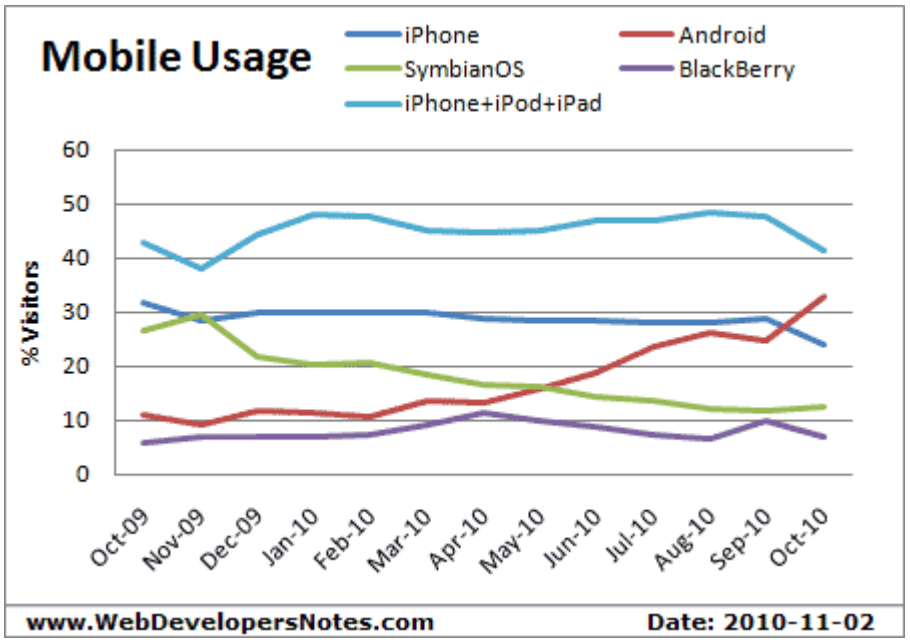
*Note

- **Microsoft** ~dropped **Silverlight** as their platform of choice for **web** because they consider **HTML5** as the only solution for the huge device/ browser /OS fragmentation

HTML5

Available HTML5 **mobile** browsers

- The HTML5 standard is not **yet** finished. The best **partial** mobile implementations are considered to be:
  - iPhone
  - Android
  - WebOS
  - Other will follow **soon**



Viewport meta tag

- In order to prevent the mobile browser from zooming out when loading the page we can add the **viewport** meta tag to the **<head>** element. A 980px width is assumed otherwise.

- We can also specify if the user is able to zoom.

- The following tag sets the viewport width to the width of the device and disables user zoom:

```
<meta name="viewport" content="user-
  scalable=no, width=device-width">
```

Viewport meta tag

Without

With

UI SDKs

- Several UI SDKs allow the development of applications that replicate the features of native UI
- **iWebKit** (link)
- **jQTouch** (link)
- **jQuery Mobile** (link)
- **Sencha Touch** (link)
- You can write your **own** CSS & JavaScript **UI**

UI SDKs (iWebKit)

```
<!DOCTYPE HTML>
<html >
<head>
<link rel="Stylesheet" type="text/css" href="css/iwebkit.css">
<meta name="viewport" content="user-scalable=no, width=device-width">
<meta content="yes" name="apple-mobile-web-app-capable" />
</head>
<body>
<div id="topbar">
  <div id="title"> Context</div>
</div>
<div id="content">
  <fieldset>
    <ul class="pageitem">
      <li class="checkbox"><span class="name">Distance to POI</span>
        <input type="checkbox" />
      </li>
      <!-- ….. -->
      <li class="checkbox"><span class="name">Temperature</span>
        <input type="checkbox" />
      </li>
    </ul>
  </fieldset>
</div>
</body>
</html>
```

Client side **storage**

- In the past:
  - Only cookies
- HTML5
  - SessionStorage – data is stored with the window object and is discarded upon closing
  - LocalStorage – data is saved after the windows is closed and is available to all pages from the same source
  - WebSQL – offers a JavaScript API to store persistent data in a local SQLite database.

Local/Session Storage **Example**

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.4.min.js" type="text/javascript"></script>
<title>Session & Local Storage</title>
<script type="text/javascript">
$(document).ready(function(){
		//Try to set the value from session storage
		var valueStoredUsingSessionStorage = sessionStorage.getItem('valueStoredUsingSessionStorage');
		$('#lbSessionStorage').text(valueStoredUsingSessionStorage);

		//Try to set the value from local storage
		var valueStoredUsingLocalStorage = localStorage.getItem('valueStoredUsingLocalStorage');
		$('#lbLocalStorage').text(valueStoredUsingLocalStorage);
});
```

Local/Session Storage **Example**

```
function StoreSessionValue()
{
		var valueStoredUsingSessionStorage = $('#txSessionStorage').val();
		sessionStorage.setItem('valueStoredUsingSessionStorage', valueStoredUsingSessionStorage);
		$('#lbSessionStorage').text(valueStoredUsingSessionStorage);
}
function StoreLocalValue()
{
		var valueStoredUsingLocalStorage = $('#txLocalStorage').val();
		localStorage.setItem('valueStoredUsingLocalStorage', valueStoredUsingLocalStorage);
		$('#lbLocalStorage').text(valueStoredUsingLocalStorage);
}
</script>
</head>

<body>
<h1>Session storage</h1>
<input id="txSessionStorage" type="text">
<input type="submit" value="Store value" onClick="StoreSessionValue()"><br><br>
Currently stored value: <span id="lbSessionStorage"></span>
<h1>Local storage</h1>
<input id="txLocalStorage" type="text">
<input type="submit" value="Store value" onClick="StoreLocalValue()"><br><br>
Currently stored value: <span id="lbLocalStorage"></span>
</body>
</html>
```

WebSQL **Example**

```
< script type = "text/javascript" >
var db;
$(document).ready(function() {
    if (!window.openDatabase) {
        alert('Databases are not supported in this browser.');
        return;
    }
    var shortName = 'WebSqlDB';
    var version = '1.0';
    var displayName = 'WebSqlDB';
    var maxSize = 65535;
    db = openDatabase(shortName, version, displayName, maxSize);
    db.transaction(function(transaction) {
        transaction.executeSql(
                'CREATE TABLE IF NOT EXISTS User ' + '
                (UserId INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,' + `
                FirstName TEXT NOT NULL, LastName TEXT NOT NULL);',
                [],
                 function(transaction, result) {;},
                errorHandler);
        ListDBValues();
    });
});
```

WebSQL **Example**

```
function AddValueToDB() {
    if (!window.openDatabase) {
        alert('Databases are not supported in this browser.');
        return;
    }
    db.transaction(function(transaction) {
        transaction.executeSql(
                'INSERT INTO User(FirstName, LastName) VALUES (?,?)',
                [$('#txFirstName').val(),  $('#txLastName').val()],
                function() {alert('Record added');},
                errorHandler);
    });
    return false;
}
```

WebSQL **Example**

```
function ListDBValues() {
    if (!window.openDatabase) {
        alert('Databases are not supported in this browser.');
        return;
    }

    $('#lbUsers').html('');
    db.transaction(function(transaction) {
        transaction.executeSql('SELECT * FROM User;', [], function(transaction, result) {
            if (result != null && result.rows != null) {
                for (var i = 0; i < result.rows.length; i++) {
                    var row = result.rows.item(i);
                    $('#lbUsers').append('<br>' + row.UserId + '. ' + row.FirstName + ' ' + row.LastName);
                }
            }
        }, errorHandler)
        });
}

function errorHandler(transaction, error) {
    alert('Error: ' + error.message + ' code: ' + error.code);
}
```

WebSQL **Example**

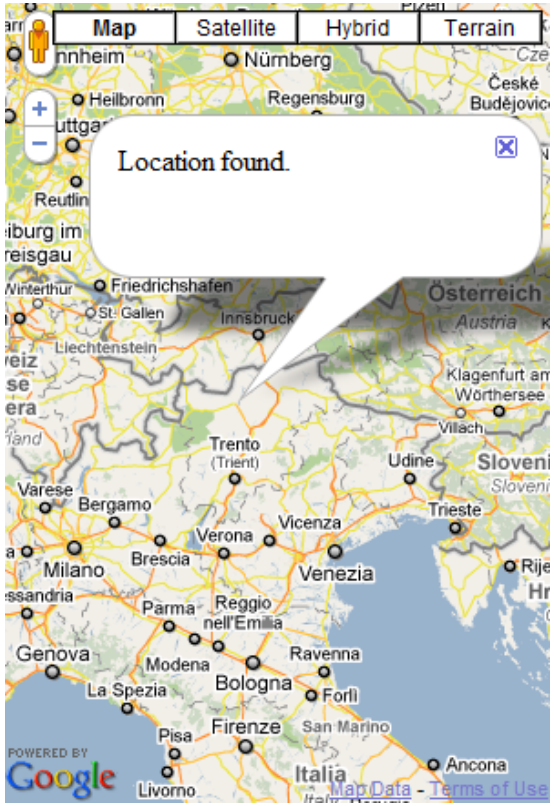
```
</script>
</head>

<body>
<h1>WebSQL</h1>
<input id="txFirstName" type="text" placeholder="FirstName">
<input id="txLastName" type="text" placeholder="Last Name">
<input type="button" value="Add record" onClick="AddValueToDB()">
<input type="button" value="Refresh" onClick="ListDBValues()">
<br>
<br>
<span style="font-weight:bold;">Currently stored values:</span><span id="lbUsers"></span>
</body>
</html>
```

Location **data**

- GPS coordinates can be obtained using the W3C Geolocation API

- Combining Geolocation API with Google Maps SDKs:



Example

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>GeoLocation API</title>
<meta name="viewport" content="user-scalable=no, width=device-width">
<meta content="yes" name="apple-mobile-web-app-capable" />
<script type="text/javascript">

$(document).ready(function(){
navigator.geolocation.getCurrentPosition(
      function (position) {
            alert('Latitude: '+position.coords.latitude + ' Longitude: '+position.coords.longitude);
      }
      );
}
);

</script>
</head>
<body></body>
</html>
```

Offline Web Applications

- The *offline application cache* allows users to run web apps even when they are not connected to the internet

- In order to add the manifest:

  <html **manifest="Web.manifest"**>

- The manifest contains a list of all the files that should be cached locally (downloaded in the background)

Offline Web Applications

- NETWORK: always request from the server
- FALLBACK: fallback mechanism
- “WEB.manifest” content:

CACHE MANIFEST

index.html

NETWOK:

logo.jpg

FALLBACK:

images/ images/offline.jpg

Other **HTML5** features

- 2D Canvas Animation API in HTML5
- Semantic Document Structure
- Native Audio and Video Controls

~**Native** Web Apps

- Several frameworks allow the conversion of web application to hybrid ones that can be installed and can access more device features

- **PhoneGap** (link)

- **RhoMobile** (link)

- **Titanium Mobile** (link)

- Others…

~Native Web Apps (PhoneGap)

	IPHONE	ANDROID	BLACKBERRY	SYMBIAN	PALM
GEO LOCATION	✔	✔	✔	✔	✔
VIBRATION	✔	✔	✔	✔	✔
ACCELEROMETER	✔	✔	OS 4.7	✔	✔
SOUND	✔	✔	✔	✔	✔
CONTACT SUPPORT	✔	✔	✔	✔	N/A

Microsoft **Windows Phone 7** will be supported soon!*

*Source: http://www.phonegap.com/about, http://wiki.phonegap.com/w/page/28291160/roadmap-planning

Further reading

- J. Stark, Building Android Apps with HTML, CSS, and JavaScript. O'Reilly Media, Inc., 2010.
- S. Allen and V. Graupera, "Pro Smartphone Cross-Platform Development: IPhone, Blackberry, Windows Mobile and Android Development and Distribution. Apress, 2010.
- G. Frederick and R. Lal, Beginning Smartphone Web Development: Building Javascript, CSS, HTML and Ajax-Based Applications for iPhone, Android, Palm Pre, Blackberry, Windows Mobile and Nokia S60. Apress, 2010.
- W3C Geolocation API Specification - dev.w3.org/geo/api/spec-source.html

Further reading

- **Device APIs and Policy Working Group** http://www.w3.org/2009/dap/
- **Mobile Web Application Best Practices -** http://www.w3.org/TR/mwabp/
- **JQTouch -** http://www.jqtouch.com/
- **iWebkit -** http://www.iwebkit.net/
- **jQuery Mobile** - http://jquerymobile.com
- **PhoneGap -** http://www.phonegap.com/

Thank you!