PubTracker
A mobile Web Application

Continuous Assessment in Distributed and Mobile Computing

Felix Middendorf

X00069339

29.04.2009

This paper walks through the design decisions in the building of PubTracker – a mobile web application which helps friends to meet spontaneously when they go out.

Contents

1 Introduction

PubTracker is a mobile web application which facilitates spontaneous meetings of friends in the evening by allowing them to exchange their current location, i.e. the pub they are currently in.

2 Requirements

2.1 Targeted Users

At the current stage PubTracker is a self-hosted solution built for a closed group of friends or acquaintances, e.g. a class in college. This implies that everyone allows the everyone else to see his or her status updates.
Future versions of PubTracker may be open to the general public and include a permission management system that allows users to manage a list of friends which are allowed to see status updates.

2.2 Features

- Log in using a user name and a password.
- Users can update their own status by selecting their current position from a number of stored locations.
- A list of pubs is available to support the decision where to go.
- A list of all users who are currently in a pub is available, too.
- A table of all active users including their current location is available.
- Status updates expire after a reasonable period of time.

2.3 Design Goals

The following design goals are to be achieved with descending importance:

- High usability despite the limitations of mobile clients.
- Honouring the cost implications of mobile networking.
- Can be used with a variety of different (low-end) mobile clients.
- Adheres to the relevant web standards as defined by the World Wide Web Consortium (W3C).
- Underlying data can easily be manipulated by an administrator.
- Only open-source software should be required to operate the application.

3 Application Design

3.1 Standards and Best-Practices

The site is built based on a collection of best-practices which were collected and edited by Luca Passani [3]. It adheres to the XHTML Mobile Profile (XHTML-MP) standard as defined by the Open Mobile Alliance (OMA) and recommendations by the W3C [1, 2].
All pages are valid XHTML-MP and have been checked with the Markup Validation Service[1] of the W3C. The Casacading Style Sheets (CSS) used throughout the page was validated with the W3C's CSS Validation Service[2].
PubTracker was tested with simulators of OpenWave[3] and Opera Mini[4].

3.2 Client Recognition

Mobile clients are recognised based on the "X-Wap-Profile" and the "Accept" headers of their Hypertext Transfer Protocol (HTTP) requests. If PubTracker identifies a mobile device, the mime type "application/vnd.wap.xhtml+xml" is served. Although PubTracker is targeted at mobile clients, users can visit the site using more sophisticated devices, e.g. a netbook or a personal computer, too. In this case they are served a Extensible HTML (XHTML) 1.0 Strict version in "application/xhtml+xml".

3.3 Layout

In order to reduce the number of HTTP requests, the traffic and costs, PubTracker does not make use of images at all. Instead, colors, characters and symbols are used. For instance, each menu link is accompanied by an opening guillemet (≪) which looks a bit like an arrow. Users generally associate this symbol with the backward button, e.g. of a stereo device.
CSS are used to improve the readability, e.g. high contrast "zebra tables".
The application should be more or less self-explanatory. The copy used in the application is reduced, clear and concise in order to accommodate the constraints imposed by the little screen. Thus, little scrolling is necessary to use the page.
In general, the most important information and links are displayed right at the top of the page (with the exception of the branding headline).

3.4 Usability

Most links have access keys that are intuitively clear to the user. A link back to the menu is included at the end of each page. The access key of this key is the same everywhere: no matter where, pressing the zero key opens the menu.
Users do not have to enter large amounts of text using the keypads of their mobile phones. Wherever possible text input is replaced by selectors or links. Thus, users only have to enter text when they login. However, due to the technique which is used to

[1] `http://validator.w3.org/`
[2] `http://jigsaw.w3.org/css-validator/`
[3] `http://developer.openwave.com/dvl/tools_and_sdk/phone_simulator/`
[4] `http://www.opera.com/mini/demo/`

identify user session (more about this in 3.6), the user can "bookmark" his login. Thus, he theoretically only has to enter it once.

3.5 CSS

CSS are used to improve the usability of the site and brand it. However, only little is used to achieve the aforementioned effect. The minified version is only about 450 bytes in size.
PubTracker's use of CSS does not fully comply with Passani's best practices [3]. This is by intention for the following reasons. Passani advocates to use only inline CSS in order to avoid loading and parsing another file.
However, a large percentage of the PubTracker stylesheet is used by every single page. Thus, by extracting all CSS into a single file it can be easily cached by the browser. PubTracker minifies the stylesheets before sending them to the client and includes an "Expires" header that is far in the future. Thanks to this the impact of the failure of buggy yet popular devices to cache hitherto retrieved CSS, which Passani warns of, is not considerably high. There is certainly a trade-off here.

3.6 Security

The software can be operated both over HTTP or HTTP Secure (HTTPS) depending on the configuration of the web server. However, HTTPS might be overkill as the data involved can be considered non-sensitive.
Instead of cookies, which cause a lot problems on many devices due to poor support, the different users are identified via a token that is part of the Uniform Resource Locator (URL). When the user logs in, a unique token is randomly generated and stored in his record in the database. The users client is now given the token and embeds it in the URL. Thus, the system is able to identify the user's session by looking up the received token in the database. The token generation is undertaken with every single login. Thus, a new login destroys older sessions. This guarantees that an account is only used by one client at a time.
All passwords are stored as salted hashes. The hash algorithm is the rather weak md5, though [4]. However, the used algorithm can easily be altered.

3.7 Database and Administration

By default PubTracker uses SQLite, an embedded, single file Relational Database Management System (RDBMS) that offers an Structured Query Language (SQL) interface, to store all its data. SQLite is supported by a number of tools. Thus the data can comfortably manipulated.

4 Networking Considerations

PubTracker is very likely to be a low-traffic site during the day. If used in Ireland, the activity is likely to be between 7p.m. and 4a.m. Most people would use such an application on a Friday or Saturday night.

Up to five times an evening seems to be a reasonable estimate of the amount of interactions per person.

Such a typical user interaction includes:

- Displaying the menu (login can be omitted thanks to bookmark)
- Displaying the list of people
- Displaying a pub

According to measurements using FireBug, this has a footprint of less than 3kb. The status update that would typically follow later adds another 2kb. Of course, these numbers highly depend on the number of people actively using the site – the more people post their updates, the more XHTML-MP is generated and has to be transported.

Due to the fact that PubTracker does not use any other elements than XHTML-MP pages and only a single CSS that is likely to be cached, only one, at most two HTTP requests per page are made.

In order to reduce the amount of traffic, the CSS is minified. Moreover, the XHTML-MP code is rather minimalistic and stripped down to the bare minimum necessary. Many pages are considerably smaller than 1kb in size; most smaller than 2kb depending on the activity of the users.
The traffic footprint could possibly be further reduced by compressing the data before sending it, e.g. with "mod_deflate" if the Apache Web Server is used. However, this is more of a web server configuration issue and PubTracker does not need to be altered for this.

These considerations show that at the current stage a very basic Virtual Private Server (VPS) should be sufficient to host such a site. There is no need to worry about requests per second and peak load times until the service is extended and opened for public (see 2.1).

5 Installation and Set-Up

5.1 Dependencies

On Debian GNU/Linux and related distributions, e.g. Ubuntu, the following commands install all dependencies. Root rights are necessary.

```
# update the list of available packages first
apt-get update
apt-get install php5 php5-sqlite3 sqlite3
```

5.2 Database

PubTracker uses SQLite. Although not tested, other RDBMS might work as well if a PHP Data Objects (PDO) driver exist. PDO's connection string can be configured in "config.php". A SQLite database is set up as follows.

```
cat schema.sql | sqlite3 data.db
```

Pubtracker brings along some demo user accounts and pubs in a seperate file.

```
cat demodata.sql | sqlite3 data.db
```

Afterwards the chmod command needs to be used in order to grant the user who runs php write access to these files.

5.3 Webserver

The “DocumentRoot”, as it is called in the Apache Web Server, should point to the “www” subdirectory. The “open_basedir” option of php should be set to PubTracker’s base directory (very likely to be named “pubtracker”).

References

[1] Open Mobile Alliance. Xhtml mobile profile, 2001. http://www.openmobilealliance.org/tech/affiliates/wap/wap-277-xhtmlmp-20011029-a.pdf.

[2] World Wide Web Consortium. *XHTML 1.1 – Module-based XHTML (Working Draft)*. World Wide Web Consortium, 2 edition, 2007. http://www.w3.org/TR/xhtml11/.

[3] Luca Passani. Global authoring practices for the mobile web, 2009. http://www.passani.it/gap/.

[4] Thomas Ptacek. Enough with the rainbow tables: What you need to know about secure password schemes, 2007. http://www.matasano.com/log/958/.

A Abbreviations

CSS Casacading Style Sheets

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

HTTPS HTTP Secure

OMA Open Mobile Alliance

PDO PHP Data Objects

RDBMS Relational Database Management System

SQL Structured Query Language

URL Uniform Resource Locator

W3C World Wide Web Consortium

XHTML Extensible HTML

XHTML-MP XHTML Mobile Profile

VPS Virtual Private Server

B Declaration of Authorship

The work herein contained is the exclusive work of Felix Middendorf.

29th April 2009