



UvA-DARE (Digital Academic Repository)

The mobile library catalogue

Koster, L.; Heesakkers, D.

Published in:

Catalogue 2.0: the future of the library catalogue

[Link to publication](#)

Citation for published version (APA):

Koster, L., & Heesakkers, D. (2013). The mobile library catalogue. In S. Chambers (Ed.), *Catalogue 2.0: the future of the library catalogue* (pp. 65-91). London: Facet.

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <http://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

This is a preprint of a chapter accepted for publication by Facet Publishing. This extract has been taken from the author's original manuscript and has not been edited. The definitive version of this piece may be found in Sally Chambers (editor), *Catalogue 2.0: the future of the library catalogue*, Facet Publishing 2013, which can be purchased from www.facetpublishing.co.uk. The author agrees not to update the preprint or replace it with the published version of the chapter.

Chapter 5: The Mobile Library Catalogue

Lukas Koster and Driek Heesakkers

Introduction

What is a mobile catalogue? A mobile catalogue is a view on a library's collection, with corresponding services, targeted at customers using mobile devices. This definition summarizes the issues involved and the questions that need to be answered.

What are mobile devices? What are mobile applications? Who are mobile users? What type of library are we talking about? What is the collection? Is the mobile view different from the standard view? Which services are targeted at mobile customers? In this chapter we will find out the answers to these questions.

Beginning with defining exactly what a mobile device is, and what it is not, we will then move on to explore the different kinds of mobile applications it is possible to implement including looking at the advantages and disadvantages of each. An overview of a range of mobile platforms is provided followed by a brief explanation of the mobile phone network technology. As mobile library services need to be developed for the end users, the user needs of the target audience from a range of different types of library will be explored. Building on this understanding of user needs, the different types of mobile library services are then explored, before looking specifically at what functionalities of a library catalogue can be provided via a mobile device.

With this background knowledge in place, we will then turn our attention to putting it into practice. Using the University of Library of Amsterdam's 'UBA Mobile' implementation as a Case Study, the practical steps that need to be undertaken to 'get a mobile catalogue' will be followed. This part includes practical tips and lessons learned to assist in making the task of implementing a mobile catalogue easier. Next comes a selection of implementation examples spanning various types of library and different software platforms. The chapter concludes with the ten-point checklist outlining the steps to set up a mobile catalogue.

Mobile devices

The first question to be answered is: which mobile devices are we referring to? The main feature that distinguishes mobile computing from previous technologies is the fact that it enables people to have internet access literally any time, anywhere. This limits the type of mobile devices to the ones small enough to carry around in a pocket or handbag such as mobile smart phones and small internet devices like the iPod Touch.

Besides the "any time anywhere" quality, the other distinguishing feature of mobile devices is location awareness. The physical location of a device can be determined by GPS (Global Positioning System), the telecom provider's network or other means, and subsequently used by programs running on the device.

Both features play an important role in determining the mobile services to offer as an organisation. They will be dealt with in the section on mobile library services.

Tablets usually are location aware. Netbooks do not normally connect to mobile networks or GPS, unless explicitly configured for that or connected to a smartphone.

Mobile devices are small, in order to fit in pockets and handbags. Therefore, they also have limited display screens and input channels, either touch screen, keyboard or both. Mobile services should accommodate these conditions. All essential information should be displayed within the boundaries of the mobile screen, without requiring the user to navigate left-right or up-down, and still be legible. It is important to carefully select what information to show and what to omit. Of course, this should be the practice for all applications and websites, but on larger devices (including tablets and netbooks, which is a reason why they are not focussed on in this chapter) there simply is more space to work with.

Navigating, selecting, clicking and entering text on mobile devices is mainly performed by touching specific spots on the touch screen with one's fingers. Even devices with physical keyboards often use finger touch for navigation. Clickable areas such as buttons and links should therefore be substantially bigger than mouse pointer targets, even more so because of the extra small dimensions of touch screens. As a rule, where twenty pixels diameter are sufficient to easily click a button with a mouse, on a mobile screen this should be fifty to be comfortably touched by a finger.

Simply offering existing applications and websites on mobile devices is not sufficient. Though the device will try to squeeze existing content to fit into its limited screen and input space, the results vary from tolerable to unusable. Users see this as a last resort, and when a mobile-optimised alternative presents itself, they will quickly leave the old site or program behind. Mobile applications should be explicitly designed with these

physical limitations in mind. This also applies to mobile content and services in relation to size limitations, as well as location awareness and any time anywhere access, as we will see.

In this chapter, we have chosen not to focus on netbooks and tablets, such as iPads. These devices are closer to regular computers than smartphones. They are too big and heavy to carry around literally 'any time anywhere', and their screens are large enough to accommodate most websites. Of course, they can be used for a number of specific mobile services. In these cases they will be considered.

Mobile applications

What is a mobile application? This seemingly simple question is not as easy as it would appear.

In the current generation of smartphones, the term 'app' is mostly used for so-called 'native applications' (from here on, 'native apps'). These are programs that run on the smartphone, much like programs on traditional computers. Confusingly however, the term 'app' is also often used for so-called web-applications ('web apps'), which are run from a webserver like any website, but can have much of the look and feel of native apps and can even be stored locally for offline use.

Native applications are generally more polished, but cost significantly more to create than web apps. Which to choose - native, web, or both - depends on an organisation's ambition and budget.

Web apps

Five years ago, web application development was marred by browser incompatibilities and slow uptake of new functionality. The new web development standards HTML5ⁱ and CSS3ⁱⁱ, actively supported by the Firefox browser, which became a real and important alternative to Internet Explorer, turned that around. Mobile browsers lagged behind and only supported a subset of these standards until the advent of Webkitⁱⁱⁱ. This open source software forms the core of the browsers on all iOS^{iv} and Android^v devices, as well as the Blackberry browser from version 6 upwards. This common core makes it possible to build feature-rich web applications in one version for all major platforms^{vi}.

The advantages of web applications are primarily for developers:

- One version for all (modern) platforms can be developed
- Skills required are very similar to those of regular web development, which is likely available in-house (and cheaper to contract)
- Updates are made on the server, no need for the user to upgrade
- Independent of platform vendor and app store

The disadvantages of web apps lie mostly in the user realm:

- Some functionality of the device is not or less instantly available (map and GPS integration, gestures^{vii}, fast scrolling)
- The lack of a single store makes discovery and installation less simple
- Most widgets are not device-specific, making the integration into the mobile platform less seamless

Native apps

The big advantage of native applications is the possibility to create an enhanced user experience, tailor-made to suit the device or platform. The downside is that for each platform, a separate app needs to be created.

The platforms

The Apple iOS platform is simple to oversee. There are two physical forms (iPhone up to 3GS, iPod touch and the iPhone 4). Apps can be tailored to screen size, on-screen keyboard and gestures. Apps can only be installed from Apple's App Store (except for the 5 to 10% of iPhones that are 'jailbroken'^{viii}).

The Google Android platform is less straightforward. Many physical forms exist, with varying screen sizes and resolution, as well as different input mechanisms. On top of that, there are many versions of Android in use, as often devices are intentionally locked from upgrading (in april 2012, Google reported nearly 94% of android devices was running a version from 2010 or earlier)^{ix}. Vendors also add their own graphical user interface on top (Sense^x, Motoblur^{xi}, etc.), creating a user interface that is not consistent across the Android platform. By default, apps can be acquired from the Android Market or downloaded by browser. However, some device vendors or carriers block this, and only allow customers to use their own 'market'. Altogether, this makes it hard to create an Android app that is polished and easily available across the platform.

The RIM platform for Blackberry^{xii} has traditionally had a large share of the smartphone market. Until recently, Blackberry users typically used their device much less than Android or iPhone users for browsing the web and apps, according to data usage

statistics by platform^{xiii}. Blackberries are mainly used for text communication, i.e. email, text messaging and 'ping'^{xiv}. When RIM introduced a new, Webkit-based browser, this pattern seems to have been broken, but this has not stopped the decline in usage of the platform^{xv}.

Windows Phone (the successor to the Windows Mobile platform)^{xvi} is the new mobile operating system developed by Microsoft. The first release of Windows Phone 7 has at the time of writing failed to have significant impact on the market. The strength of Microsoft, and its partnership with Nokia make it unlikely to disappear in the short term, but whether it will become a significant platform remains to be seen. For ageing or niche platforms such as Symbian^{xvii}, Bada and WebOS^{xix}, that is even more doubtful.

Even for a large library, it will be too expensive to create apps for all platforms. Our advice is to focus on iOS and Android, unless it is known that other platforms are widely used in your local community.

Advantages of native apps over web applications

- Usability: Native apps are easy to integrate with device-specific user interface guidelines, can use advanced animations to create a smooth user experience, and can use the full screen when needed, (although in theory a browser can display a web app in full screen mode, the location bar, buttons and other browser clutter, are tricky to hide completely).

- Functionality: Native apps have better access to low-level device functionality, such as map and GPS integration, gestures, fast scrolling, smooth animations etc.
- Discovery and installation: The various app stores have become the first place where users turn to look for apps. Easy installation is provided with an icon on the home screen. It is possible to have an icon pointing to a web app on the home screen, but it is less straightforward and many users do not know about this.

Disadvantages of native apps

- Expensive: Specific programming skills are needed, whether in-house or contracted
- Fragmentation: A separate app is needed for each platform, or even device. For each, only part of the development can be re-used. In addition, apps need to be suited to the platform; users dislike apps that are obviously 'ported' and use non-standard buttons, gestures and other graphic elements. Cross-platform frameworks that allow easy 'porting' of programs from one platform to the other can be used, but it is difficult to give an app created with such a tool a native feel. Costs can be reduced by limiting support to a selection of platforms. However, this will annoy users of unsupported devices, especially if no web app is provided as an alternative.
- Deployment via 'app store' depends on the vendor (for example, the Apple App store has gained notoriety due to the opaqueness of the app approval process^{xx})
- No control over updates, this is up to the user. In practice, older versions of the app will remain in use, and improvements do not reach all users.

Network

A key difference between mobile devices and regular computers is their connection to the internet using the mobile phone network. There are a number of systems in use, some offering faster services than others, but never as fast and reliable as the broadband that users have become accustomed to at work and in their homes.

When available, mobile devices can use Wi-Fi networks to make a fast connection.

When designing an application, this should however not be taken for granted, as not only is Wi-Fi not always available (or free), it also quickly drains the batteries. Therefore many users leave it turned off.

Network performance depends on bandwidth and latency. Bandwidth^{xxi} is the capacity of data that can travel the network, latency^{xxii} the reaction time between sending a message and receiving the answer. For a mobile catalogue, large bandwidth is not very important, for browsing and searching requires but a fraction of the bandwidth needed to stream video. High latency however is, as it translates directly to responsiveness. It will increase the time waiting for search results to be returned to the mobile application and can create a near-instant experience for the user.

A library has no control over the network. It has some control over the catalogue application on the client and the library management system on the server.

Improvements in performance may be achieved by adjusting the way that both systems use the network. On the client side, this can be accomplished by displaying an intermediate page with the first results. The remaining results can be then displayed as they arrive. The AJAX^{xxiii} (Asynchronous JavaScript and XML) web development

technology can be used for this. Another approach would be to group multiple requests to the server into a single query. On the server side, it may be possible to optimise the network configuration to give priority to requests made to the library management system. This work should be undertaken by experienced system administrators. If such fine-tuning of the network settings is possible in your organisation, it will benefit all users of the catalogue, not only the ones using mobile devices.

Types of library and mobile users

Who are the users that libraries want to reach with their mobile services and catalogues? There is no single answer to this question, because there are many different types of libraries with different types of collections and customers. Libraries can be classified using a number of different criteria. The five criteria listed below are directly related to mobile services and user experience:

1. Target audience (local community, staff, students, specific professions, researchers, global community)
2. Number of branches/locations (single location/multiple branches)
3. Collection type (local/remote, physical/digital)
4. Stack type (open/closed)
5. Policy (lending/reference)

Libraries range from small single location public libraries with an open stack physical collection in a deprived area, to multiple branch closed stack university libraries with a large number of online subscriptions, and everything in between.

In the online digital world, access to library web services, including mobile services, is available to everyone. However, access to library services via mobile devices will tend to occur within the actual target audience of a specific library. This is even more the case than for standard web services targeted at non-mobile devices/use. The reason for this is because mobile services are aimed at fulfilling user needs directly related to a particular library. Within a library's target audience, only the owners of mobile devices will use the mobile services. Within this group of mobile device owners, only those who actually have a need for these services will use them.

Mobile services

An academic library offers a range of online services to its users. Two recent projects; the Beluga project^{xxiv}, a next-generation catalogue for libraries in the Hamburg area and the research on mobile technologies from the North Carolina State University (NCSU) libraries^{xxv}, recommend the following:

- aim at saving your user's time

- do not simply copy the existing services to a new interface, think of the different user needs in this specific (mobile) context

- users have high expectations; they appreciate autocorrect, 'did you mean?' suggestions, thesauri, faceted browsing, etc. On mobile devices this becomes even more important, as tiny or on screen keyboards mean more spelling mistakes.

- reuse existing infrastructure

- for digital resources, access is just as important to the user as discovery, if not more.

Mobile library services can be divided in three groups.

1. Practical information

- address, contact, directions, maps
- opening hours
- information on the current availability of workstations in the learning centre

Practical information is well suited for mobile presentation. It is limited enough to present on a small screen; it requires relatively little input from the user, cumbersome compared to a regular workstation; and the location awareness can be used to tailor the information to the user ('what is the nearest open branch?').

2. Discovery

- broad searches
- narrow searches
- finding known items

When searching widely, a patron typically skims many results from generic subject searches. This is much harder on a small screen. More specific searches and known item searches however, can still be presented in a useful way.

3. Delivery

- Adding the items to bibliographic databases (citeulike^{xxvi}, endnote^{xxvii}, etc.)

- For physical objects: requesting the item for collection
- Skimming (reading the abstract, skimming the full text)
- For digital objects: reading; reading and annotating; reading and writing

The user can add found items to a reading list, a bibliographic database or another form of social network.

Other uses depend on the object. If it is only available on paper, the user can request it. If an abstract is available, the user can read that. If it is available online - and it can be accessed - the user can read it directly, either to skim it or in-depth. Finally, the user can read it while making notes.

With the limitations of the small screen, it seems unlikely that many people will do much reading on their mobile device beyond skimming. This will be different for tablets. The larger the screen, the more skimming may turn into reading.

Mobile catalogue functions

A library catalogue is the gateway to a library's collection. Through the library catalogue users can find items in collections and get access to them. Traditionally, a library has been a physical location for keeping a physical collection of books and other printed or handwritten items. Users had to go to the physical library location and consult the local physical catalogue (in the form of lists or later cards) to find descriptions of items that might be of interest to them, possibly with the assistance of librarians. In the case of a closed stack item, the user needed the assistance of library staff to actually access the item. If the item in question was available for lending, the customer was allowed to

borrow the item.

In the computer age, the digital catalogue made its appearance. Initially, the electronic library catalogue was only available on standalone workstations. Customers still had to come to the library to find and get items. Only after the introduction of the online web catalogue, or OPAC^{xxviii}, customers were able to search and find items remotely.

Catalogues became digital a long time before collections did.

As library collections are becoming digital, either because items are "born digital" or because they have been digitised, library users will increasingly be able to not only find items remotely, but also to get access to them online, without having to visit a physical library location. This depends on the type of library and, most importantly, on the nature of its collection. Public libraries will for some time still have largely local physical collections, whereas research libraries tend to have subscriptions to a large number of online digital resources.

The meaning of the concept "library collection" has changed from "all items a library owns" to "everything a library has access to". This includes both physical and digital items the library owns itself, and physical and digital items owned by other organisations (libraries, publishers, etc.).

For library customers it may well be that finding items (the catalogue function) will become separated from getting access to items (the delivery function). They may find interesting items in Google or other libraries' catalogues, and may even get direct

access to freely available content online. However, in order to get access to physical items or restricted access digital material, they need to be affiliated to an organisation that can provide that access, for instance a public or university library.

A mobile catalogue will essentially play the same role as the "standard" online library catalogue does. As catalogues are becoming more and more part of 'discovery tools' which aim to integrate discovery and access, so will their mobile counterparts. The "any time anywhere" and "location awareness" features of mobile devices will affect both discovery and delivery. The question is, in what circumstances people will use their mobile devices to search a library catalogue, and what they will subsequently do with the results.

Someone will probably use their mobile device to perform a search in a library catalogue only because, for whatever reason, they need to find a library item then and there and have their smartphone at hand.

More interesting is what happens next. What do people do with results from a mobile catalogue search? This is directly related to the user's objective, the nature of the collection and the items found: physical or digital and for borrowing or reference. Library collections generally consist of textual material targeted at reading (books, journals, articles, sheet music, etc.), audio (mainly music), video (films), images (drawings, photographs, paintings, maps, etc.) and games. All of this material can be physical or digital, or both.

Let us look at digital collections and items first. If someone intends to use an item directly on his or her smartphone, they will want to download it immediately. This may be

an e-book, a journal article, a film or some music. It will also depend on whether they have permission to access the item and download it. If it is text, people may choose to read it on their tiny smartphone screen, if they do not intend to process it in any other way, like making notes, writing summaries, analysing data, etc. If people need books or articles for studying or processing in any way, they are likely to just bookmark the results and download or print these later on a laptop, pc, e-reader or other device. This appears to be confirmed by usage statistics. For instance, in their report on *Mobile Device User Research* for the California Digital Library Mobile Strategy^{xxix}, Rachael Hu and Alison Meier quote research from the Library of Texas A&M University which shows that only 1% of searches in EBSCO Mobile led to a full text download, as opposed to 77% of regular EBSCO searches.

Physical items not only include printed books, journals, images, but also music (in the form of CD's, tapes or vinyl), films (in the form of DVD's, videotapes) and games. Mobile catalogue users can request items to be sent to the nearest branch location. If it concerns items from a reference collection, they can be bookmarked or requested for on-site use. In the case of open stacks customers could be guided to the location of the item they are interested in by using the location awareness of the mobile device in combination with a map. An even more integrated approach would be to use 'Augmented Reality', in which a live image from a mobile device's camera is overlaid with information relevant to the view^{xxx}. For example, further information about the library's collection, similar to the type of information displayed on signs in a physical library, could be displayed to the user directly on their mobile phone screen.

A useful mobile function for physical items will be enabling users to renew items via their mobile device. After receiving an alert by email or text message, library users can prevent fines this way if they are physically not in the position to return their items. Even paying the fines can be a mobile function.

Paradoxically, the latest digital library developments appear to be closely linked to the physical world.

How to get a mobile catalogue

New library systems, especially the new generation of integrated discovery tools like Summon^{xxx1}, Primo^{xxx2}, VuFind^{xxx3}, etc.^{xxx4} are more likely to offer an out-of-the-box mobile interface than the older library management systems. These are likely to be sometimes rather clunky web apps, and often some customisation is needed to create a smooth mobile experience.

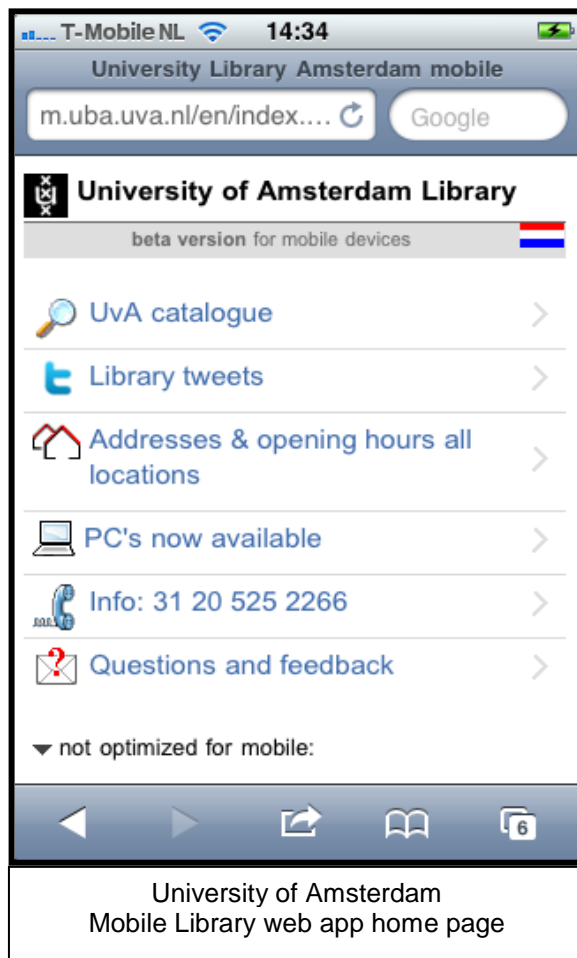
If a usable mobile front end is not included in the system used by your library, then there are a number of options to get onewith a "do it yourself" approach.

- Libraries can build their own native apps using an API to their library system or an export of their catalogue data.
- Libraries can partner with commercial library app framework vendors like Boopsie or Blackboard.
- Libraries can build their own web app using an API to their library system or an export of their catalogue data. Optionally an existing web app framework like iWebkit can be used.

- If their library system allows adding and customising OPAC front ends, libraries can create extra OPAC front ends optimised for mobile use.

Case study: UBA Mobile

Having explored the mobile applications, platforms and the networks, it is time to put this background knowledge into practice. To do this, we will use the University Library of Amsterdam's 'UBA Mobile' implementation project as a case study.



In the spring of 2010, the library of the University of Amsterdam (UvA)^{xxxv} set up a mobile service in a project called 'UBA Mobiel'. This was accomplished with minimal budget and within weeks, using an 'agile' software development approach. It is an interesting case study for libraries on a tight budget.

Several of the library's staff are enthusiastic, early adopters. Early on, they noticed that the library's web-presence was rather lacking on mobile devices. It was hard to use, required

much scrolling and zooming, and it looked amateurish. In the spring of 2010, the Technical University in Delft, The Netherlands (TU Delft)^{xxxvii} was making headlines in both traditional media and high-profile blogs with their iPhone app. Although the app covered the whole university, the library figured prominently. Meanwhile, library staff at

the University of Amsterdam, noticed that a huge increase in mobile devices use by library users, especially students, as found in a simple, non-scientific test, namely counting smartphones in the cafeteria. Although in the long term the library system vendor would likely provide a solution, with the users ready and waiting now we couldn't afford to wait for that to happen. Something had to be done.

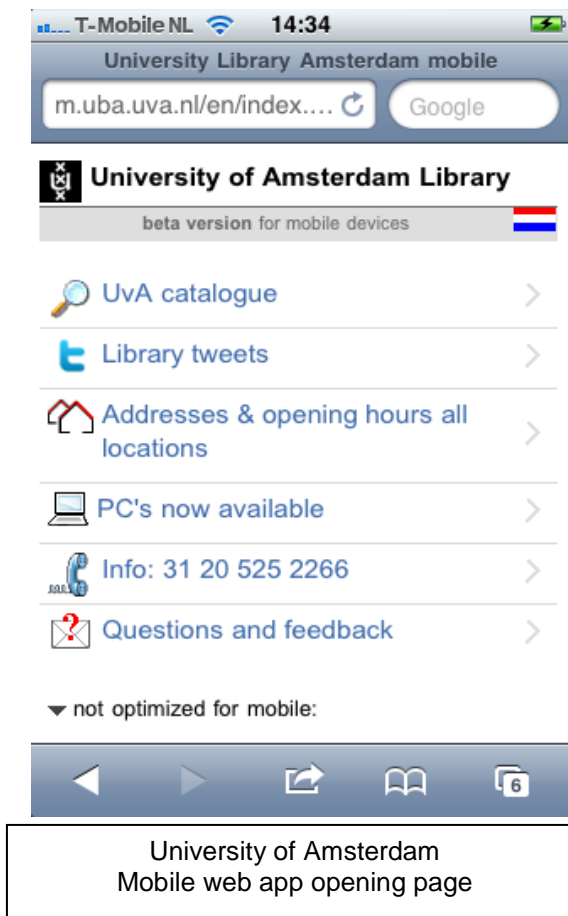
Resources were limited. The library had just finished migration to a new library management system and introduced RFID self-service and was still clearing the backlog. A project was allowed with a limited budget of €1.500 and a maximum of 250 hours of staff time. The project structure had to be minimal and the University of Amsterdam library took inspiration from the 'agile' programming style.

A small team was assembled, consisting of staff from the library's user services and IT departments. For the three months of the project, library staff could spend at most four hours a week on the project. To make the most of this limited time, short meetings were held every fortnight to assess and re-assess the priorities. In these meetings, ideas were identified, then prioritized by multiplying two factors: importance and ability to build quickly, both scored on a scale from 3 (most important / easiest to build) to 1 (least important / hardest to build). The outcome was a list of tasks, starting with the highest priority items (the 'nines', very important and easy to build), working towards, the lower priority items (less important and harder to build).

Early in the project, the decision was made to focus on creating web-apps only. Native apps would cost far too much money and time for a 'quick-win' project.

The spirit of 'eternal beta' was embraced. The prototypes that were built, was put out to the public to test. Their feedback was used to reassess the importance of functionality in the priorities list. To complement the spontaneous emails, we conducted several short surveys to obtain more detailed feedback.

Simple web app



In a short period of time, the team came up with three products. The first is a simple web app, presenting the library at a dedicated URL: m.uba.uva.nl.

This app gives quick access to the information users find the most important when using a mobile device: opening times, location information, phone numbers and news. The main website has a link to the mobile version, which has a low-profile in the regular view, but features more prominently when seen on a mobile browser. The mobile web app includes a link to the 'full' website. Automatic

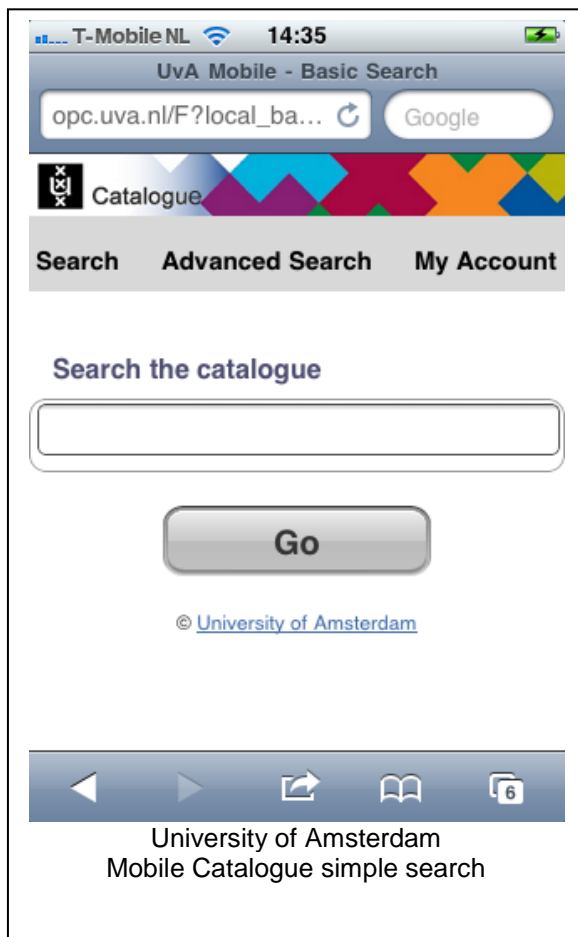
redirection, based on device detection was considered, but rejected as not user-friendly enough. Detection errors cannot be ruled out, especially with new devices, and in such case the user would not have a choice and be forced to use the 'wrong' version.

Staff that were used to creating regular HTML pages learned very quickly how to write mobile HTML^{xxxix} as these are not very different. We did find out that it was important to

test the app on real devices, rather than using emulators. Emulating was fine for testing the layout, but the tactile touch navigation cannot be properly understood on a mouse-driven interface. A variety of mobile devices were therefore bought for this purpose.

To encourage users into giving feedback, the bottom of the web app featured Facebook-style 'thumbs-up' and 'thumbs down' icons. Both linked to a new email message, with either 'like' or 'dislike' in the subject line. This strategy worked, as the team received dozens of replies, most of them positive and, more importantly, with suggestions for improvements and next steps.

Mobile version of the catalogue



The second product delivered was a mobile version of the catalogue. This would have been impossible to accomplish if it would have to be built from scratch. Luckily, an active open source community exists for the library management system used, Ex Libris' Aleph^{xl}. The University of Jönköping in Sweden^{xli} had already created a mobile 'shell' for the Aleph OPAC and made it available^{xlii}. This uses a simple 'trick'. Aleph uses a template system with a different set of files for each language. The mobile version was added as a new template set, for the

language 'Mobile'. This template set was then adapted further.

In line with the advice to "do less, but better" in the constraints of a mobile internet environment, choices had to be made, for both layout and functionality. The regular OPAC featured a banner and a large menu, both were trimmed. Functionality was brought back to the basics: browsing and searching, requesting and renewing.

Creating a usable advanced search page was especially difficult, as the drop down menu functionality takes up too much screen space. To complicate things further, web forms display very differently on the iPhone and Android platforms, which made it very hard to create a layout that worked well across these platforms. The advanced search functionality for the mobile catalogue therefore had to be severely limited compared to the regular catalogue. In the same vein, the default simple search was trimmed down to one field and a large 'go' button.

Although there still is room for improvement in the mobile version of the catalogue that resulted at the end of the project, it is still considerably more usable than using the regular web version of the catalogue on a mobile device. Feedback from users showed that although they noticed the shortcomings, they were, on the whole, enthusiastic that the functionality was now available on mobile devices, especially requesting items from the closed stacks.

Recommendations

In addition to the 'simple web app' and the mobile catalogue, the project also delivered a third outcome; recommendations for follow-up. This report was based on the items on the priority list that scored four, or lower, based on the user feedback and the experience of the staff.

The first recommendation was that in future vendor negotiations, for example when considering a new Content Management System (CMS), a mobile interface should be included as a must-have requirement. This mobile interface should work on the various mobile platforms and this should be tested thoroughly on actual devices.

Secondly, it was recommended to continue refining the mobile web catalogue, and strive to make it function as much as possible as a native application. The regular catalogue should also get QRcodes^{xliii}, to make it possible for users to scan an item's QRcode with their mobile device and have the location and availability information ready when locating the item in the open stacks.

Finally, the maintenance and further development of both the mobile catalogue and the web app should become part of the core tasks of the appropriate department.

Also, a number of possible 'quick wins' were listed, that did not make it in the short duration of the project. Some of these depended on data from third parties. As soon as the relevant data became available, their priority status would be increased to 'high'. The most prominent example was real-time availability numbers for computers in the study centres, which makes it possible to create a 'find the nearest free computer' function. Other possibilities, such as a mobile version of metasearch and full-text finder,

depended on the suitability and availability of more complex open source software, that needed time to investigate.

It was recommended not to put more effort into location-aware services, such as the automatic detection of the nearest library branch when requesting material, into the OPAC at this time. Technically, creating such a service as a web app is achievable, the 'NCSU WolfWalk^{xliv}' sets a great example. The challenge lies however in integrating the app into the existing library system.

Reusing information from the library's Content Management System (CMS), also turned out to be complicated, due to closed nature of the current CMS. The practical information, such as locations, opening times, holiday schedules etc. unfortunately therefore need to be maintained in two places, within the CMS for the main website and in Google Calendar for the mobile app.

Next steps

In August 2010, four months after the launch of the first version of the mobile web app, QR codes were added to the standard OPAC. It was also decided to integrate the opening time and location pages *from* the mobile app *into* the main CMS-driven site instead of vice versa. This provided an alternative way to maintain this data in one place.

In November 2010, after the central University ICT department released real-time workstation availability data, the 'currently available workstations by location' service was added. This turned out to be the most successful mobile library service, as described below under "Usage". It proved so popular that VisualSpace, an independent software company, created a free iPhone app, 'Plaats'^{xlv} ("available space" in Dutch)

that simply scrapes the page and displays it within a simple native app. This app has seen 394 new downloads (purchases) from the iTunes App Store in the first ten weeks it has been available, which is around 40 a week.

Development of mobile versions of the metasearch and full-text finder services was abandoned when the institution decided to acquire Ex Libris' Primo unified discovery tool, that came with an out-of-the-box mobile interface. Unfortunately, significant work was still needed to make it usable. Primo, including the mobile view, was launched in April 2012.

Usage

After the launch of the mobile web app in early April 2010 collection of usage statistics started in May 2010. Analysis of the usage statistics is made using Google Analytics^{xlvi}.

Mobile catalogue

Between May 1, 2010 and April 1, 2012, the mobile catalogue had on average 9 visits per day, with 6 page views per visit. This amounts to 54 page views per day, with 60% of the traffic originating from mobile devices. Of the total number of visits 48% came from returning visitors. For the mobile devices the percentage of returning visitors was 52%. This was less than 1% of the total use of the standard OPAC, which averaged during that same period around 3000 visits per day, with 10 page views per visit, amounting to 30,000 page views per day. The percentage of returning visitors was 66%.

Mobile web app

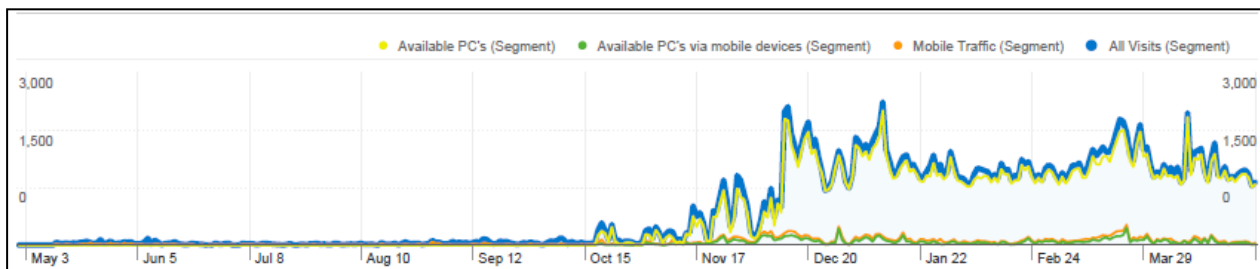
During the first six months, following the launch of the mobile web app, between May 1 and November 1, 2010, the number of unique visits to the mobile web app (excluding the mobile OPAC) averaged 16 per day, with 25% originating from mobile devices. Of the total number of visits 38% came from returning visitors. For the mobile devices the percentage of returning visitors was 60%. The most visited page was the mobile web app home page (80%).

The regular library website shows on average around 4100 visits per day on weekdays, with around 45% returning visitors.

Currently available work-stations service

After the introduction of the “currently available work-stations” service the number of visits to the mobile web app increased to 150 per day on average (November 1, 2010 - April 1, 2011), with 33% originating from mobile devices. Of the total number of visits, 52% came from returning visitors. For the mobile devices the percentage of returning visitors was 75%.

The most visited page was the “currently available workstations” page (90%, or around 1200 page views per day).



Page views per day, mobile web app (excluding mobile catalogue).
The significant increase after the introduction of “currently available work stations’ service is almost completely caused by that particular service

In the period November 1, 2010 – April 1, 2012 the number of visits to the mobile app is around 100 per day on average, with 45% from mobile devices, 64% returning visitors, and 83% returning visitors for mobile devices. There is a peak in May (214 visits per day) right before the end of term, and a low in July and August (54 visits per day) during the summer break.

The most visited page was the “currently available workstations” page (95%, or around 930 page views per day)

About 75% of visits to the “currently available workstations” page were direct hits. 17% originated from links from other sites, including the regular Library website and only 7% originated from the mobile web app home page.

According to Google analytics, only 6% of the visits originated from mobile devices. This percentage does not include hits made by the ‘Plaats’ app. This might raise the actual use on mobile devices, and also indicates that discovery through the app store draws a different audience than a regular (mobile) webpage gets. But even if the number of app users is equal to those visiting using mobile device webbrowsers, seven out of eight visits comes from regular computers.

Impact and lessons learned

Both the mobile website and the mobile OPAC appear to attract a steady but limited number of visits per day. Only the “currently available workstations” service attracts a significant number of visitors, and even then only one in eighth comes from a mobile device. This seems to indicate that academic library users prefer regular workstations over mobile devices for their day-to-day work .

Although the number of visits to the mobile catalogue is only a tiny fraction of the usage of the standard OPAC, this does not mean that the service should be discontinued. The high number of returning visitors indicates that those who use it find it handy. The usage statistics of the mobile catalogue do not diverge much from the usage statistics of the other mobile web app services of the library website. However, when a library does not have any mobile services yet, the introduction of a mobile service for practical information should be considered first, before a mobile catalogue.

It is clear that mobile users expect practical services that are useful to them 'here and now.' Libraries should not ignore this. The mobile catalogue offers a number of these 'here and now' services that are really important, for example; search and bookmark for later use, request items for picking up and renewing loans.

The mobile catalogue offers a search window on only a part of the library's content, consisting mainly of the traditional print holdings. As part of the implementation of a new library unified discovery interface the mobile catalogue will be replaced by the mobile interface included in the discovery interface, which will give mobile search access to the catalogue, library repositories, online databases and electronic journals.

Lessons learned

The 'quick win' approach and 'agile' style worked well. With minimal resources (200 hours of staff time, spread over 5 staff, and €1100 out of pocket costs), the Library of the University of Amsterdam managed to set up a mobile presence that seems attractive to the users, as the percentage of repeat visitors is average to high.

Secondary results were a boost in staff morale and publicity for the library in various well-read magazines and blogs. Also, in-house knowledge of mobile devices and their requirements was gained, especially useful in negotiations with vendors.

Other points:

- Don't be afraid to make mistakes!

- Stimulate the early adopters among your staff and use their ideas.

- It is vital that staff have some freedom to experiment with tools of their own choice.

Setting up the mobile website would not have been possible if the webmaster had not had the freedom to use a standard webserver as playground for experimentation, instead of being limited to the University CMS system with its technical and political constraints.

- And finally: you cannot work on mobile presence without owning some devices! If smartphones are too expensive, or cannot be purchased due to contractual issues, Wi-Fi only devices such as Apple's iPod Touch can be an alternative.

Mobile Catalogues: Implementation Examples

The following section provides a selection of mobile catalogues that have been implemented in a range of libraries. The examples cover public versus academic libraries web versus native apps and commercial versus in house development.

Los Angeles Public Library with Boopsie

Boopsie, www.boopsie.com

iOs, <http://itunes.apple.com/app/lapl-to-go/id383513855>

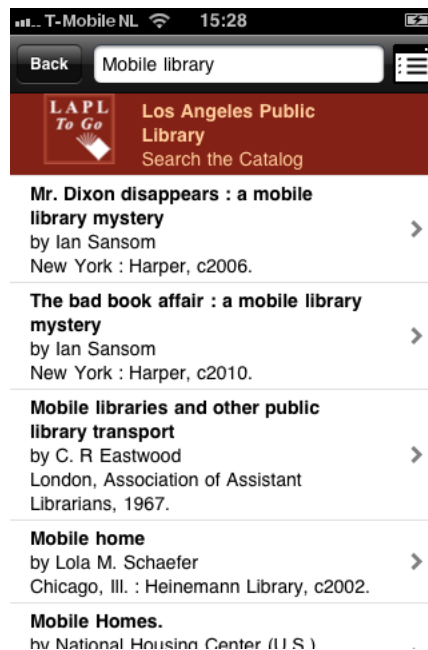
Other platforms, <http://lapl.boopsie.com>

Los Angeles Public Library homepage, www.lapl.org

Boopsie is a commercial vendor that offers a configurable framework for native apps for libraries. Besides practical library services it also integrates any proprietary ILS. The Los Angeles Public Library chose this approach to create their 'LAPL To GO' app.



'LAPL To Go', Los Angeles Public Library's iPhone app, home screen



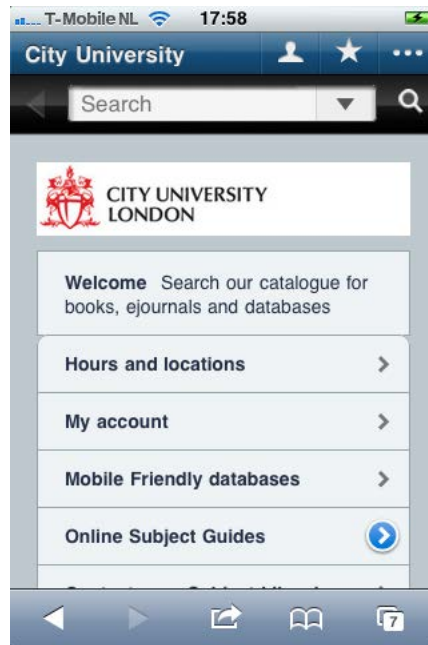
'LAPL To Go' mobile catalogue
results page

City University London Library with Library Anywhere (by LibraryThing)

<http://www.libanywhere.com/m/229>

The free social book readers' network LibraryThing offers a commercial framework for web and native (iPhone, Android, Blackberry) apps for libraries. This offers practical library services and integrates with many proprietary library management systems.

There is only one app to find and use all libraries that have implemented the framework, but libraries can link to their specific site, such as in this example from City University London library.

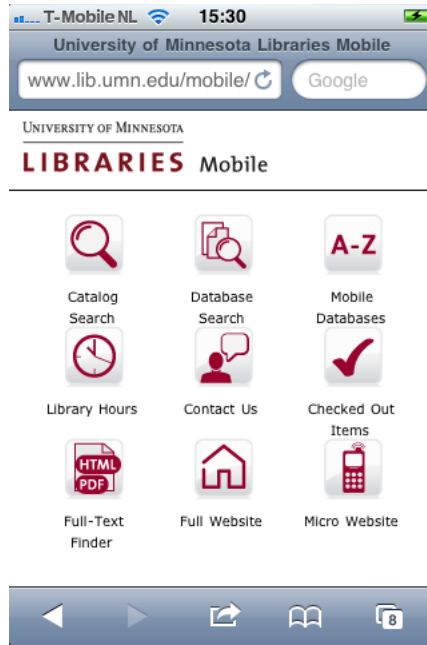


City University London's mobile library interface using libanywhere.com

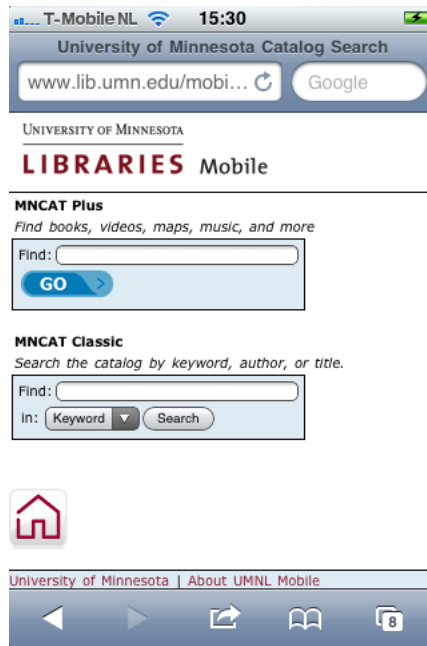
University of Minnesota Library

<http://www.lib.umn.edu/mobile/>

The University of Minnesota Library developed their own mobile web app in-house using PHP scripts. In addition to practical library services, the mobile web app offers access to a number of catalogue and content services by connecting to the library catalogue Aleph, the discovery interface Primo, the federated search engine MetaLib and the OpenURL link resolver SFX via an Application Programming Interface (API).



University of Minnesota libraries mobile web app home screen

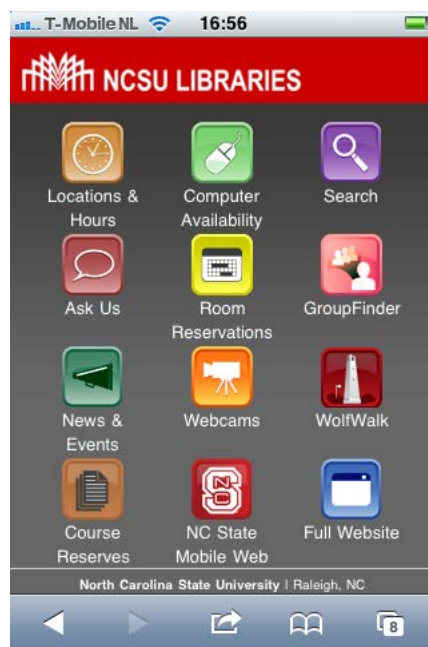


University of Minnesota mobile catalogue search screen

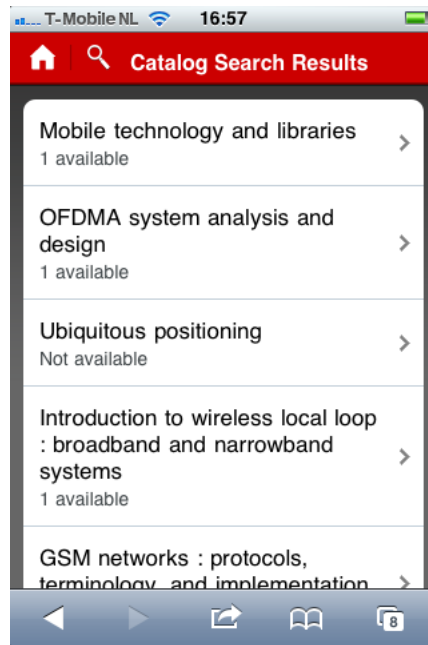
North Carolina State University

<http://m.lib.ncsu.edu/>

'NCSU Libraries Mobile' shows how far a web app can go. The layout remains clear and usable across different devices and is well adjusted to the screen size. In the search and search results screens, only the most important information is shown at first.



NCSU libraries web app home screen



NCSU libraries mobile catalogue search results page

To keep the search results list compact, the NCSU mobile catalogue only shows title and availability.

Conclusion

There are still many uncertainties surrounding the deployment of mobile library services, especially when it comes to the catalogue. It appears to be too early to be able to determine what kind of library services will be desired and appreciated on the mobile platform. It is essential that end-user experiences and feedback are monitored closely and that mobile library applications can be adjusted rapidly, based on that feedback. However it seems that services of a practical nature are appreciated more than mobile catalogues. Mobile users expect services that are useful to them here and now. This is confirmed by recent studies^{xlvii}.

Mobile services providing practical information, like contact information and opening hours, of course are not specific to libraries. However, requesting physical library material, based on the results of a catalogue search, and renewing loans certainly are valid mobile library services. These services will be used on a mobile platform as long as libraries provide access to physical material.

Providing access to all kinds of information can be considered a core service that libraries offer. However, libraries should start thinking "outside the box". In the near future mobile customers will not be satisfied when they only get bibliographic metadata on their smartphones in reply to a search for information.

Offering real "location aware" services on the mobile platform will prove to be essential. For example, the automatic provision of the nearest library location when requesting a book. In this case, the physical location is directly related to practical mobile services.

Further in the future, "location awareness" will be linked more to the need for real information and this will not necessarily be limited to library-related information services. Augmented reality will also play a role, for instance to provide detailed information on a specific building in view. Through an augmented reality application information will become available in very diverse forms, from a number of different sources. One of these sources might be a library catalogue. However, there will also be museums, archives and local government agencies, providing such information. Based on the users' physical location, one of the mobile services available may be, for example, a reference for a book at the nearest library.

In order to make linking of information from different sources possible, the concept of linked data should be embraced by more content providers (for more information, see Chapter 7 'Enabling your catalogue for the semantic web'). The future of mobile internet lies in the combination of location awareness, augmented reality and linked data.

Implementing mobile library services: Checklist

1. Identify your target audience or audience groups
2. Identify the mobile services you want to offer
3. Identify available internal and external useful services that can be integrated
4. Identify level of in-house native app and web app development capacity
5. Check existing frameworks and suppliers that can be used, both for native and web apps
6. Determine time frames for implementation
7. Decide on whether to implement a native or web app
8. Decide on outsourcing or in-house development
9. Don't forget PR
10. Gather and process feedback continuously

References

ⁱ The HTML5 standard, is currently being revised, <http://dev.w3.org/html5/spec/Overview.html>

ⁱⁱ CCS3, www.css3.info

ⁱⁱⁱ Webkit, www.webkit.org

^{iv} iOS, Apple's iPhone Operating System, www.apple.com/ios

^v Android, www.android.com

^{vi} Even though most mobile browsers are now based on webkit, there are still some (relatively minor) differences that need to be considered when developing a mobile webapp. See www.quirksmode.org/webkit.html for a detailed overview.

^{vii} Gestures are the finger movements made on a touch-screen mobile device.

^{viii} According to Wikipedia, jailbreaking 'is a process that allows devices running Apple's iOS (also known as iPhone OS prior to iOS 4.0) operating system (such as the iPad, iPhone, iPod Touch, and recently Apple TV) to gain full access (root access) to unlock all features of the said operating system', http://en.wikipedia.org/wiki/IOS_jailbreaking

^{ix} Statistics taken from the official Google Android Developers Platform Versions page: <http://developer.android.com/resources/dashboard/platform-versions.html> – retrieved on april 14, 2012, collected during a 14-day period ending on April 2, 2012

^x HTC Sense, http://en.wikipedia.org/wiki/HTC_Sense

^{xi} Motoblur, <http://en.wikipedia.org/wiki/Motoblur>

^{xii} RIM (Research in Motion), the technology behind BlackBerry devices, www.rim.com

^{xiii} Consumer Reports: 'iPhones hog much more data than other smart phones' <http://news.consumerreports.org/electronics/2010/02/iphone-data-usage-smart-phones-smartphones-blackberry-mb-network-att-carrier-istress.html>

^{xiv} Pinging is a BlackBerry to BlackBerry messaging system.

^{xv} StatCounter Global Stats report, December 1 2010 noted the sharp increase in Blackberry browser usage: <http://gs.statcounter.com/press/blackberry-overtakes-apple-in-mobile-wars> - however, from december 2010 onwards the Blackberry browser share fell from 18% to 6% in march 2012 in the Statcounter Global Stats, see http://gs.statcounter.com/#mobile_browser-ww-monthly-201001-201203. This is confirmed by the comScore, see http://www.readwriteweb.com/archives/microsofts_mobile_comeback_is_looking_terrible.php

^{xvi} Windows Phone, http://en.wikipedia.org/wiki/Windows_Phone_7

^{xvii} Symbian, <http://en.wikipedia.org/wiki/Symbian>

^{xviii} Bada, [http://en.wikipedia.org/wiki/Bada_\(operating_system\)](http://en.wikipedia.org/wiki/Bada_(operating_system))

^{xix} WebOS, <http://en.wikipedia.org/wiki/WebOS>

^{xx} Approval of iOS apps, http://en.wikipedia.org/wiki/Approval_of_iOS_apps

^{xxi} Bandwidth, [http://en.wikipedia.org/wiki/Bandwidth_\(computing\)](http://en.wikipedia.org/wiki/Bandwidth_(computing))

^{xxii} Latency, [http://en.wikipedia.org/wiki/Latency_\(engineering\)](http://en.wikipedia.org/wiki/Latency_(engineering))

^{xxiii} AJAX programming, [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))

^{xxiv} Beluga, is a home-grown next-generation catalogue for all the libraries in the Hamburg area in Germany, <http://beluga.sub.uni-hamburg.de/>, see also Christensen, A. (2009) Next generation catalogs: What do users think? Conclusions from the beluga project in Hamburg, IFLA 2009, Satellite Meetings, Florence, www.ifla2009satellitelflorence.it/meeting3/program/assets/AnnChristensen.pdf

^{xxv} Tito Sierra, 'Mobile Library projects at North Carolina State University' ,CNI Fall 2009 Membership meeting, Washington, DC, www.lib.ncsu.edu/dli/projects/wolfwalk/docs/cni2009f.ppt

^{xxvi} Citeulike, www.citeulike.org

^{xxvii} Endnote, www.endnote.com

^{xxviii} OPAC, Online Public Access Catalogue

^{xxix} Hu, R. and Meier, A (2010). *Mobile Strategy Report, Mobile Device User Research*. California Digital Library,

https://confluence.ucop.edu/download/attachments/26476757/CDL+Mobile+Device+User+Research_final.pdf?version=1 (p.27)

^{xxx} Augmented reality, http://en.wikipedia.org/wiki/Augmented_reality

^{xxx}_i Summon, www.serialssolutions.com/summon

^{xxx}_{ii} Primo, www.exlibrisgroup.com/category/PrimoOverview

^{xxx}_{iii} VuFind, <http://vufind.org>

^{xxx}_{iv} For more information about new-generation discovery tools , see Chapter 3 'New Generation Discovery'.

^{xxx}_v Library of the University of Amsterdam (UvA), www.uba.uva.nl/en

^{xxx}_{vi} Agile software development, http://en.wikipedia.org/wiki/Agile_software_development

^{xxx}_{vii} Technical University (TU) Delft, <http://home.tudelft.nl/en>

^{xxx}_{viii} For example, www.mobilegarageblog.com/2010/03/going-mobile-tu-delft-netherlands.html

^{xxx}_{ix} HTML 4.0 Guidelines for mobile access, www.w3.org/TR/NOTE-html40-mobile

^{xl} ExLibris' Aleph library management system, www.exlibris.co.il/category/Aleph

^{xli} University of Jönköping, Sweden, <http://hj.se/en.html>

^{xlii} Aleph Mobile OPAC, created by Daniel Forsman, University of Jönköping, Sweden,

www.exlibrisgroup.org/display/AlephCC/Aleph+Mobile+OPAC

^{xliii} QR (Quick Response) codes, http://en.wikipedia.org/wiki/QR_code

^{xliv} North Carolina State University (NCSU) Libraries, WolfWalk project,

<http://www.lib.ncsu.edu/dli/projects/wolfwalk>

^{xlv} 'Plaats' app for iOS: <http://itunes.apple.com/app/plaats/id415467214>

^{xlvi} Google Analytics, www.google.com/analytics

^{xlvii} Boopsie: Non-catalog services more popular with mobile library patrons,

www.prweb.com/releases/2010/11/prweb4787284.htm