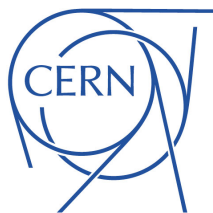


Touch-Optimised Mobile Interface for Invenio Digital Library



Yannick Tapparel
Route de la Plaque 4
3960 Corin de la Crête

Master Projects in Computer Science
École Polytechnique Fédérale de Lausanne

Under the supervision of:

Dr. Christine Vanoirbeek, EPFL, MEDIA Research Group
Dr. Tibor Šimko, CERN, Division IT

Lausanne, 2013

Every child deserves a champion —
an adult who will never give up on them,
who understands the power of connection,
and insists that they become the best that
they can possibly be.
— Rita Pierson

To M. Antille, G. Baumbach and my parents...

Abstract

Invenio is free software platform for digital libraries and document repositories on the web. Invenio was originally developed at CERN to power its scientific document server containing about 1 million of articles, books, photos, videos, and more.

Invenio software is nowadays used by about thirty large institutions and library networks worldwide and is being co-developed by institutions such as Cornell University, EPFL, Harvard-Smithsonian Centre for Astrophysics, or SLAC National Accelerator Laboratory.

The goal of this project is to enhance user experience with Invenio on modern mobile platforms. This includes studying suitability of various techniques such as developing native iOS or Android application versus using HTML5 web-based frameworks such as jQuery Mobile or Sencha Touch.

Résumé

Invenio est une suite logicielle libre pour les bibliothèques numériques et les dépôts de documents sur le web. Invenio a été initialement développé au CERN pour équiper leur serveur de documents scientifiques contenant environ 1 million d'articles, livres, photos, vidéos, etc.

Logiciel Invenio est aujourd'hui utilisé par une trentaine de grandes institutions et réseaux de bibliothèques à travers le monde. Il est co-développé par des institutions telles que l'Université de Cornell, EPFL, Harvard-Smithsonian Centre for Astrophysics, ou SLAC National Accelerator Laboratory.

Le but de ce projet est d'améliorer l'utilisation d'Invenio sur les plateformes mobiles modernes. Cela comprend d'examiner la pertinence des différentes techniques telles que le développement native iOS/Android ainsi que les frameworks HTML5 comme jQuery Mobile ou Sencha Touch.

Contents

Abstract (English)	v
Abstract (French)	vii
Introduction	1
1 Introduction	1
1.1 Motivation	1
1.1.1 Mobile Internet Growth	1
1.1.2 Fragmentation	2
1.1.3 Invenio	3
1.2 Project's Goals	3
2 Apps Development Approaches	5
2.1 Approaches	5
2.2 Native Apps	6
2.3 Web Apps	7
2.4 Hybrid Apps	8
2.4.1 Apache Cordova	9
2.4.2 Adobe PhoneGap	10
2.4.3 Other Hybrid Mobile App Frameworks	10
2.5 Comparison	11
2.6 Choice Of Approach	12
2.7 Touch-Optimized Web Frameworks	14
3 Requirements for Mobile Application Development	15
3.1 Mobile User Interface	15
3.1.1 Common Android UI	16
3.1.2 Common iOS UI	17
3.2 User Habits	18
3.2.1 Overall Appearance	18
3.2.2 Interactive Elements	20
3.3 Native Feel	20
3.3.1 Improve Hybrid App Feel	21

Contents

3.4	App Store Review Guidelines	23
3.5	Responsive Web Design	24
3.5.1	Key Elements	24
4	Design And Implementation	25
4.1	Elements	25
4.1.1	Mobile Phone	25
4.1.2	App	25
4.1.3	Server Side	26
4.2	Components	26
4.3	User Interface And Functionality	27
4.3.1	Search View	28
4.3.2	Toolbar	28
4.3.3	Menus	29
4.3.4	Search Result View	29
4.3.5	Record View	30
4.3.6	Saved Records View	31
4.3.7	Other Views	31
4.4	Tablet UI	31
4.5	Web App stucture	33
4.5.1	Files Organization	33
4.5.2	Execution	34
4.6	Off-line Records	35
4.7	Configurations Files	36
4.8	Optimizations	37
5	Results And Conclusion	39
5.1	Invenio Mobile	39
5.2	Hybrid App	39
5.3	Futures Outlook	40
6	Bibliography	41
6.1	Books	41
6.2	Technical Articles	42
6.3	Conference	42
6.4	Websites	43

1 Introduction

1.1 Motivation

In a World on the move and where the need for information seems important, the mobile solution becomes essential. Whether at work, in the mall or in the train, mobile devices are ubiquitous. With 6.8 billion mobile subscriptions, the mobile market doesn't stop growing. That is equivalent to 96 percent of the world population (7.1 billion). [ITC(2013)] .

The success of the smartphone was very fast. We are just six years after the sale of the first Apple smartphones and people already want to consult information everywhere, even away on a trip. It can be something very important, especially in the professional field. This market don't stop evolving. Information was never so accessible though this technology is very recent.

1.1.1 Mobile Internet Growth

The global mobile data traffic grows incessantly. It grew 70 percent in 2012. According to Cisco, it's not ready to cease. Global mobile data traffic reached 885 petabytes per month at the end of 2012, up from 520 petabytes per month at the end of 2011.[Cisco(2013)]

In the developed world, 77% of the population are Internet users in 2013. Internet users are people using the Internet in the last 12 months. The smartphone penetration continued to grow. At the end of 2012, two-thirds of Americans who acquired a new mobile phone chose a smartphone instead of a feature phone.[Nielsen(2012)] [Wroblewski(2011)]

In the developing world, only 33% of the population are Internet users in 2013. But the situation is evolving and several projects are in the progress of improving the situation. We can mention 2 projects:

- The Google "Loon" project is a network of balloons traveling on the edge of space, designed to connect people in rural and remote areas.[Google(2013b)]

Chapter 1. Introduction

- The 50x15 Initiative, launched by AMD, which seeks to give 50 percent of the world's population access to the internet by 2015.[Ruiz(2007)]

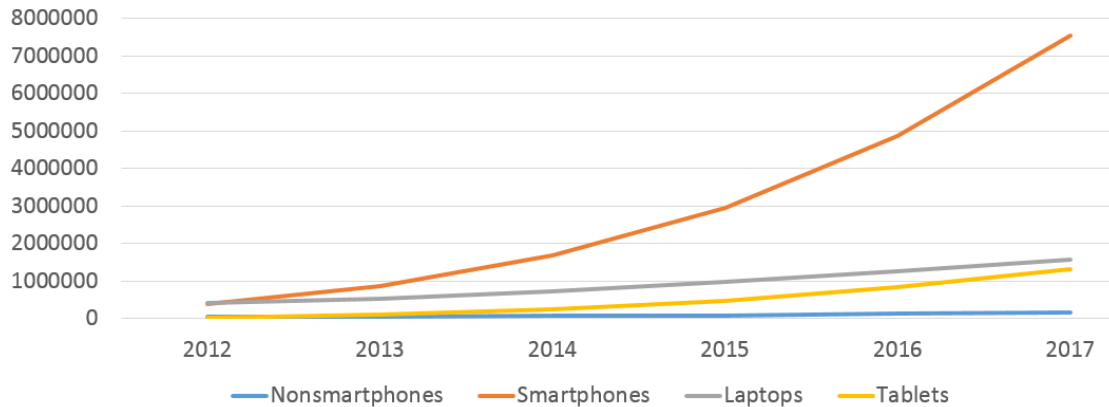


Figure 1.1: Global Mobile Data Traffic, 2012–2017, by device (TB per Month). [Cisco(2013)]

1.1.2 Fragmentation

With 79% of smartphone sales, Android realized 23.0% growth compared to the same 3 months last year. iOS was second during the period with 14.2% of smartphone sales, down 23.4% versus last year. Windows continues to make gains, up to 3.3% of smartphone sales. [Gartner(2013b)]

Table 1.1: Worldwide Smartphone Sales to End Users by OS. [Gartner(2013b)]

OS	2nd quarter 2012	2nd quarter 2013
Android	64.2	79.0
iOS	18.8	14.2
RIM	5.2	2.7
Windows	2.6	3.3
Symbian	5.9	0.2
Other	3.3	0.6

One of the severe problems of the smartphones is the fragmentation of this market. It presents a lot of problems for developers. It is very difficult to make an app¹ which is compatible with all devices. To have different OS leads to have different versions that makes supporting devices a nightmare

If we want to do an app only for Android, because it's a start, in reality this marker is fragmented in different version of this OS. iOS have also a fragmentation, but not as much as Android. Additionally, Android on some device has been modified to work closely with its hardware.[Jones(2013)]

¹mobile application

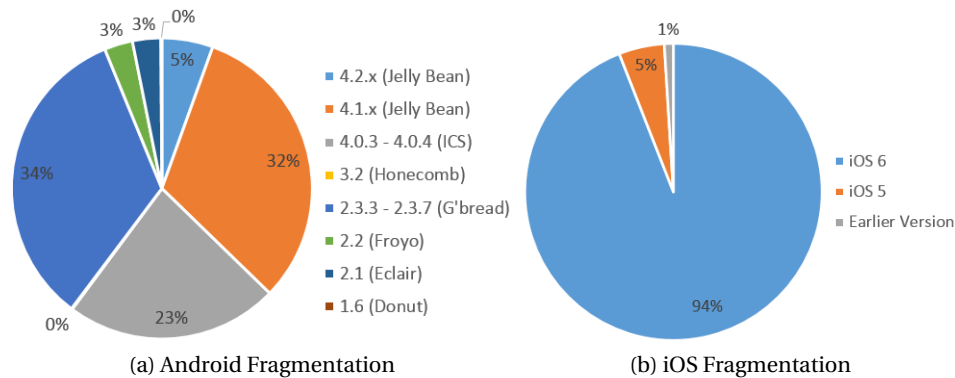


Figure 1.2: Android and iOS Fragmentation Visualized. Data :[Signal(2013)]

A lot of projects could provide new possibilities and be more accessible with this new growing market. It's the case of Invenio.

1.1.3 Invenio

Invenio is a free software suite enabling you to run your own digital library or document repository on the web. The technology offered by the software covers all aspects of digital library management from document ingestion through classification, indexing, and curation to dissemination.[Invenio(2013)]

Invenio has been originally developed at CERN to run the CERN document server, managing over 1,000,000 bibliographic records in high-energy physics since 2002, covering articles, books, journals, photos, videos, and more. Invenio is being co-developed by an international collaboration comprising institutes such as CERN, DESY, EPFL, FNAL, SLAC and is being used by about thirty scientific institutions worldwide.[Invenio(2013)]

1.2 Project's Goals

The goal of this project is to enhance user experience with Invenio on modern mobile platforms. This includes studying suitability of various techniques such as developing native iOS or Android application versus using HTML5 web-based frameworks such as jQuery Mobile or Sencha Touch. The project will include enriching APIs of various Invenio modules in order to suit chosen solution. The server-side code will be written in Python programming language and Flask web development framework, using SQLAlchemy toolkit for persistence.

Different options could be investigated for the project: one possibility would be to implement a full-featured mobile version of an Invenio site, while another direction would be to build a "companion" app that would complement/extend the features of Invenio, without fully replacing it.

At the beginning of the project a number of use cases have been imagined:

- **Search.** Searching is the most important functionality offered to users. Make sure it is relatively touch-optimised. (Already good thanks to responsive layout.) Beware of facets, beware of detailed record pages etc. Adapt responsive design as needed. Uncluttered search box. Easy-to-get and easy-to-use add-to-search. Stripped-down formats for the results (title/author, DOI, citation count). Priority: high.
- **Submit** material via mobile devices is another important functionality of Invenio web app. Priority: moderate, not really essential yet.
- **Journal.** Invenio offers overlay journal capabilities e.g. see Atlantis Times demo journal at <http://invenio-demo-next.cern.ch/journal/>. Make sure this is nicely readable on mobile devices. Priority: low due to possible move of to <http://bulletin.cern.ch/Drupal>.
- **Annotations, Baskets, Tags.** User baskets contains items of interest, possibly annotated. Desirable to access favourite papers in offline mode. Desirable to allow offline tagging and annotations. Sync on becoming online again. Push/pull considerations. Alternative: revive tagging UI and use tags for "Read Later", "Sync to Mobile", etc.
- **Offline reading.** Users may mark certain pages for future offline access, say author page on INSPIRE. Populate local offline store, study possible use of Box, Dropbox, Drive and friends in this context. Possible sync of pages on becoming online again.
- **Multimedia.** Responsive design slideshow with navigation controls. Once multimedia slideshow is ready, do the same for Plots tab.
- **Switch between Invenio sites.** Some users have split functionality e.g. internal notes living behind password on <http://cds.cern.ch/> and public papers living on <http://inspirehep.net/>. Make sure users can easily switch between services. Study cross-service capabilities, annotations etc.
- **BibCirculation and push notifications.** Send push notifications when the status of a book has changed, for e.g. when a loan is about to expire (or has expired), when a reserved book becomes available, etc.
- **Retrieve PDF of printed document.** Add the possibility to take a picture of a printed paper, and receive the corresponding electronic version (in your basket, or as link in email, etc.). The implementation could try using OCR on the photo of the paper, and then use fulltext-search to retrieve the corresponding record, or rely on the some barcode/QR-code on the printed document.

2 Apps Development Approaches

This chapter presents three different approaches to develop a mobile application and compares their advantages and disadvantages. The chapter explains and justifies the technologies choose in the project

2.1 Approaches

There is not only one way to create an application for a mobile. The organisations have to decide which technologies to follow when the project starts because this choice is an important decision that will influence the results entails parameters, such as cost, project time-frame, performances, functionalities and platform support to name a few. [IBM(2012)] This chapter distinguishes three mobile application development options:

- **Native apps** are designed for a specific operating system and device. They are developed with their respective software development kit in their programming language (e.g., Xcode and Objective-C with iOS, Eclipse and Java with Android).
- **Web apps** or **HTML5 apps** are applications that is handled with a web browser as a client over a network such as the Internet or an Intranet. [Daniel Nations()]
- **Hybrid apps** are web apps using a web-to-native abstraction layer providing the ability to use native capability. The hybrid apps are hosted in the smartphone like a native apps.

2.2 Native Apps

Native apps consist of binary executable files developed for a specific kind of device. To get the apps, the users have to download it by visiting the respectively app store. On some devices, alternative methods can be provided by the mobile manufacturer.

The native apps have a full access to the provided Low-level and High-level APIs. The low-level APIs are close to the hardware and they can interact with the device, network, audio or graphics render,... such as GPS, file system, camera, compass, and so on. High-level APIs are OS built-in components and services such as contact, calendar, phone calls, text messages,...

To work, the developers need a platform-specific SDK¹ for each mobile OS. The SDK provides tools to write the source code, to debug the project or to join additional resources. Each update of the application needs a new submission and a user update. Let's sum those up in the following table:

Table 2.1: Specific software development kit

OS	Languages	Tools	App stores
iOS	Objective-C, C, C++	Xcode	Apple App store
Android	Java (some C, C++)	Android SDK	Google Play
Blackberry	Java	BB Java Eclipse Plug-in	Blackberry App World
Windows Phone	C#, VB.net and more	Visual Studio	Windows Phone Store

Each mobile OS gives you its own user interface elements, such as buttons, menus, input fields, tab, and more. As a result, apps that are designed to work for more than one OS require the designer to be familiar with the different GUI.

Using the GPU², the native elements are loaded quickly and their animations are fluid. Native apps have faster graphics performance because they use the graphic accelerator.

¹Software Development Kit

²Graphics Processing Unit

2.3 Web Apps

The web apps constitute the mobile applications that are based on a browser-supported programming language and are displayed in a web browser. Web browser are ubiquitous in smartphones, which makes web apps particularly accessible for mobile devices. To popularize, it is a website for smartphone.

The web apps don't need to be submitted on an app store or to be downloaded by a user. That requires almost no disk space on the smartphones and that requires no upgrade procedures except to publish the new version on they web server.

People who already have experience with web development can easily implement web apps. Technologies are HTML5, CSS, JavaScript in client-side and PHP, Java, Python or all server-side languages relevant for web development.

iOS and Android devices allow users to add the web apps links to the home screen of the OS. In the case of iOS, there is no difference in the look of a web app and a native app.

Compared to previous versions of HTML, HTML5 brings new features summary: Semantics, Offline & Storage, Connectivity, Device Access, Multimedia, 3D & effect, Performance & Integration, CSS3 & Styling. All details are on the W3C website³. For the mobile devices, the essential element of HTML5 is we can use touch events, geolocation API(GPS, cellular and WIFI), canvas drawing and local storage.

In practice, compared to native application, HTML5 keeps significant limitations. The web apps developers have access to a limited number of these APIs. For exemple, HTML5 doesn't permit to use the accelerometer or to send a message, to read a contact or to capture song. For the storage, HTML5 provides an initial local storage of 5MB per domain and an unlimited session storage (limited only by system resources).[Google(2012)]. The browser needs user's agreement for some packages (up to 10MB, or 25MB, or more...). Moreover, all phone's web browsers don't have implemented the same features.

³<http://www.w3.org>



Figure 2.1: HTML5 Logo [W3C(2012)]

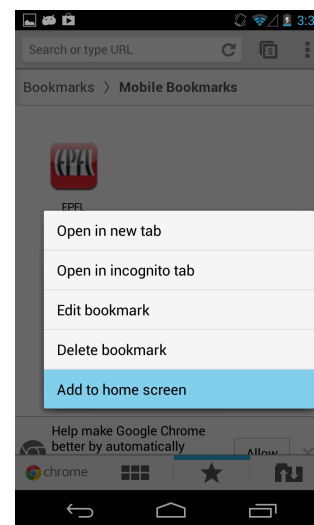


Figure 2.2: Link to the home screen

2.4.1 Apache Cordova

Apache Cordova is a free and open source mobile development framework that allow developers to use web technologies for cross-platform development avoiding each mobile native language [Codrova(2012)]. With a touch-optimized web framework, they provide tools to develop hybrid apps.

If a developer wants to use Cordova only for one part of the apps, Cordova allows to mix freely native application component by implementing a Cordova Web View in the project.



Figure 2.3: Apache Cordova Logo[Codrova(2012)]

Platforms

The Cordova JavaScript libraries are consistent across mobile platforms. As you can see in the following table of features. Cordova produces mobile or desktop applications.

Table 2.2: Set of development tools and device APIs available for each mobile platform.

	Android	Blackberry	iOS	Windows Phone 8	Windows 8
Accelerometer	Yes	Yes	Yes	Yes	Yes
Camera	Yes	Yes	Yes	Yes	Yes
Compass	Yes	✗	Yes	Yes	Yes
Contacts	Yes	Yes	Yes	Yes	Yes
File	Yes	Yes	Yes	Yes	Yes
Geolocation	Yes	Yes	Yes	Yes	Yes
Media	Yes	✗	Yes	Yes	Yes
Notificatoin	Yes	Yes	Yes	Yes	Yes
Splahscreen	Yes	✗	Yes	✗	✗
Storage	Yes	Yes	Yes	Yes	Yes

Cordova also supports some less popular mobile operating system such as webOS, Symbian, Tizen Bada,...

Contributors

Cordova is developed by the open-source community. The project sources are available on the Apache git repository : <http://projects.apache.org/projects/cordova.html>

Apache Cordova graduated in October 2012 as a top level project within the Apache Software Foundation (ASF). Through the ASF, future Cordova development will ensure open stewardship of the project. It will always remain free and open source under the Apache

License, Version 2.0. [Codrova(2012)]

2.4.2 Adobe PhoneGap

PhoneGap is a hybrid framework created by Nitobi, purchased by Adobe Systems in 2011 [Systems(2011)]. PhoneGap was shared with Apache under the name Apache Cordova. There are many articles on the Internet talking about PhoneGap without reference to Cordova. PhoneGap is just a distribution of Apache Cordova. Libraries are the same and what you can do with one can be done with the second. This product can also be found under the name of Apache Callback⁷. In practice, Adobe PhoneGap, Apache Callback, Apache Cordova are three names for almost the same product.



Figure 2.4: PhoneGap Logo
[PhoneGap(2012a)]

Adobe purposes a cloud service named Adobe PhoneGap Build. You upload your web files, the service compiles and returns the iOS, Android, Windows Phone, Blackberry 5/6/7 and webOS mobile applications.[Systems(2013)]

The PhoneGap code was contributed to the ASF⁸ under the name Apache Cordova. Through the ASF, future PhoneGap development will ensure open stewardship of the project. It will always remain free and open source under the Apache License, Version 2.0.[PhoneGap(2012b)]

2.4.3 Other Hybrid Mobile App Frameworks

There are other hybrid mobile app frameworks. To name a few:

- **Appcelerator Titanium** is a web-based platform that supports iPhone, Android and BlackBerry. JavaScript sources are compiled to a native apps.
More information : <http://www.appcelerator.com>
- **IBM Worklight** is a commercial development platform based on Cordova.
More information : <http://www-03.ibm.com/software/products/us/en/worklight/>
- **Oracle ADF Framework** is a framework that supports iPhone, Android . Java, Enterprise JavaBeans, JSP are compiled to a native apps.
More information : <http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html>
- **Mono Touch** and **Mono for Android** are web-based platforms for iOS, android and

⁷The project original name

⁸Apache Software Foundation

Windows devices. C# and .Net libraries compiled to a native app.

More information : <http://xamarin.com/monotouch>

- **Adobe Flash + Adobe Air** is a framework that executes Flash application to a phone devices.

More information : <http://get.adobe.com/fr/air/>

These solutions have probably good result, but will not be considered for this project. The reason is that all have partially or totally proprietary licences or are not free. One goal of this project is to work only with open source and free solutions.

2.5 Comparison

These three different approaches provide very different solutions that have consequent differences to the result that we obtain. The following schema summarizes the features usable for each solution:

Table 2.3: Mobile Application Development Options.

	Native	HTML5	Hybrid
App Features			
Graphics	Native APIs	HTML, Canvas, SVG	HTML, Canvas, SVG
Performance	Fast	Slow	Slow
Native look and feel	Native	Emulated	Emulated
Distribution	Appstore	Web	Appstore
Device Access			
Notifications	Yes	No	Yes
Contacts, calendar	Yes	No	Yes
Geolocation	Yes	Yes	Yes
Offline storage	Secure file storage	Shared SQL	Secure file system and shared SQL
Connectivity	Online and offline	Mostly online	Online and offline
Development skills	ObjectiveC, Java,..	HTML5, CSS, JS	HTML5, CSS, JS

Chapter 2. Apps Development Approaches

To summarize, the main differences are:

Native:

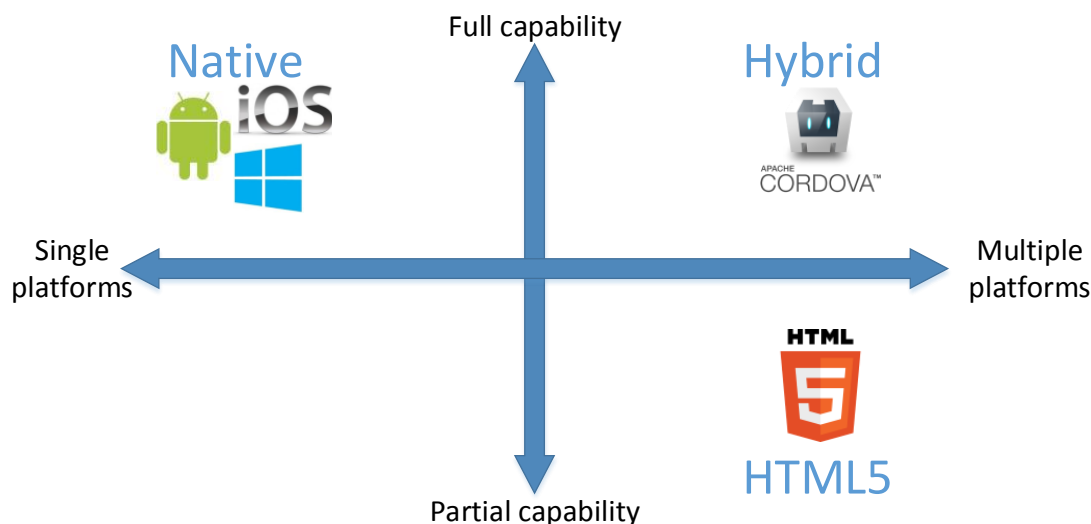
- Advanced UI interactions
- Fastest performance
- App store distribution

Hybrid

- Web developer skills
- Access to native platform
- App store distribution

HTML5

- Web developer skills
- Instant updates
- Unrestricted distribuion



2.6 Choice Of Approach

The choice of the technology is an important and difficult ruling. Each solution has its advantages and disadvantages. This question arises in many companies and the results are contradicted.

On one side, Facebook and LinkedIn decide to stop working with HTML5/hybrid solutions and to implement native apps. In this regard, Mark Zucker (co-founder of Facebook) said:

"When I'm introspective about the last few years I think the biggest mistake that we made, as a company, is betting too much on HTML5 as opposed to native... because it just wasn't there. And it's not that HTML5 is bad. I'm actually, on long-term, really excited about it."

–Mark Zuckerberg, Disrupt SF, September 2012

And on the other side we have other big companies such as IBM, Gartner and other pundits, who predict the market will stay fragmented. They said cross-platform tools should stay a good solution. More specifically Gartner, Inc. said that hybrid apps, which offer a balance between HTML5-based web apps and native apps, will be used in more than 50 percent of mobile apps by 2015: [Gartner(2013a)]

2.6. Choice Of Approach

“While hybrid apps will be the majority of enterprise mobile apps, web technologies like HTML5 will make up the most commonly used languages for building mobile applications by 2015.”

–David Mitchell Smith, vice president and Gartner Fellow, April 2013.

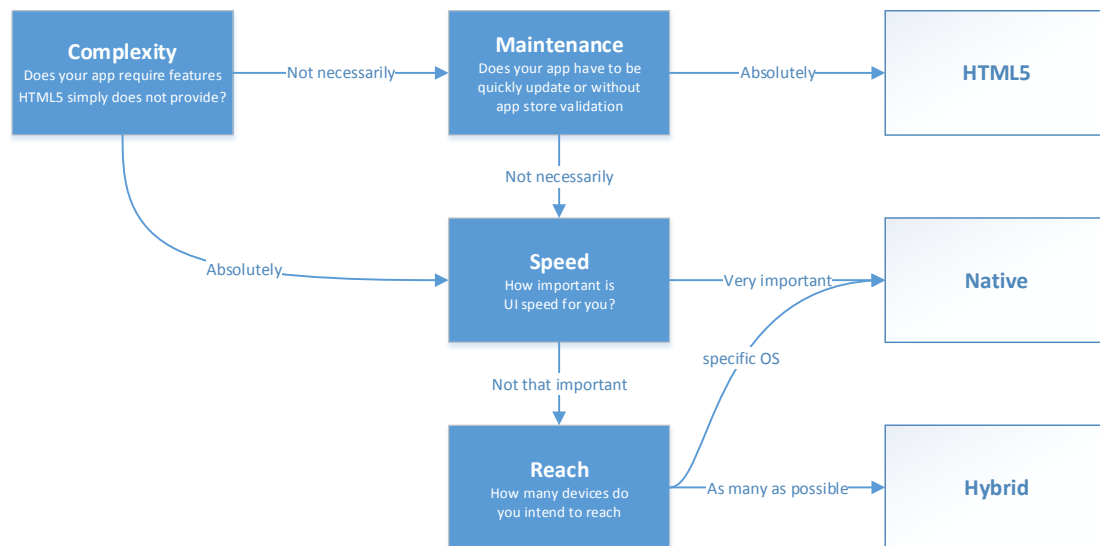
However, different requirements for consumer (B2C) and enterprise (B2E) scenarios will result in very different uses of native, web and hybrid approaches.[Gartner(2013a)]

Table 2.4: Consumer/Enterprise Split in 2015. [Gartner(2013a)]

Category	Native	Hybrid	Web
Consumer	40	40	20
Enterprise	10	60	30

In recent years with its web services like Google Documents, Google maps or Gmail, Google shows it's possible to have web application widely used with over 425 million active users for Gmail.

My opinion, is that there is not one solution that stands out in all situations. The following charts can help you get an idea of which technology to choose but doesn't work in each case. Some less easily measurable factors such as knowledge, experiences, costs, etc are also important.



In the case of Invenio, I chose to focus my project to a hybrid apps. HTML5 doesn't provide all future needed features. The actual web site is almost responsive and already works on

mobile screen. There is not the necessity to have another responsive web site.

The speed is not a very important criterion in this project. The user will employ this app to display texts, pictures, videos and pdf. That not require large graphical resources, while it is important to be functional on multiple platforms. However, the Invenio team has a great experience with HTML and other web technologies.

2.7 Touch-Optimized Web Frameworks

There are a lot of web frameworks for mobile application that we can classified into two categories. First, we have frameworks which generate a design compatible with many mobile OS such as Sencha Touch and jQuery Mobile. Secondly, we have frameworks like Ratchet and Fries which reproduce a design specific for a given OS.

jQuery Mobile and Sencha Touch are touch-optimized HTML5 UI frameworks designed to make websites and apps that are compatible with all major smartphones, tablets and desktop devices. jQuery Mobile is built on top of jQuery core by the jQuery Foundation with the MIT License. Sencha Touch is a product of Sencha based on GPLv3 or commercial Licenses depending on the use.

Ratchet is a HTML5 UI toolkit initially used for prototyping iOS apps. Ratchet is under the MIT license. Fries was inspired by Ratchet but for Android apps.

There are dozens of HTML5 UI Frameworks, but my choice was jQuery mobile. jQuery Mobile has more or less the same performances as Sencha Touch and a large amount of documentations, plugins, etc. We should keep an eye on Ratchet and Fries. These two solutions are very young but can be promising. The first release of Ratchet have less than 3 months when this thesis began and Fries didn't exist. It is a domain that evolves quickly.

3 Requirements for Mobile Application Development

In this chapter is discussed requirements and constraints a mobile application developer has to be careful. First, we talk about how the users employ the mobile application and what we have to be attentive about the user interface. In a second time, the chapter explains how to improve the hybrid app feel and how Apple and Google review all apps to ensure they are reliable and what are the provided rules. The last part presents the Responsive Web Design, or what are the rules if we want to implement a mobile app for different resolution and pixel density devices.

3.1 Mobile User Interface

To obtain an application pleasant to use, its development must be thought to conform its ecosystem. This section describes the global differences between the two main mobile OS. To justify the explanations, examples are based on some existing applications already adapt to both OS. As explained in the next section, the differences between the OS should not be overlooked.

It can seem simple to design a mobile app, but in fact is not so trivial to adapt a Android app to an iOS app and vice versa. It's not possible to just cut and paste you interface. Some elements are share between Android and iOS, but generally they perform different functions. [Rowinski(2013)]

3.1.1 Common Android UI

1) Status Bar

The status bar contains time, battery charge, network connexion and signal strength, volume, update information, or notifications.

2) Main Action Bar

This Bar contains the main elements for the navigation. The action bar has several areas .

The first element is the app icon. It can change if you wish. It provides information about the app.

After that, there is the view control that allows to switch between different views or accounts.

In the third position we found the action button that provides the most important actions in the current section.

And finally we have the action overflow pinned to the right side. It's a contextual menu with the specific less frequently used actions.

All details can be found on <http://developer.android.com/design/patterns/actionbar.html>

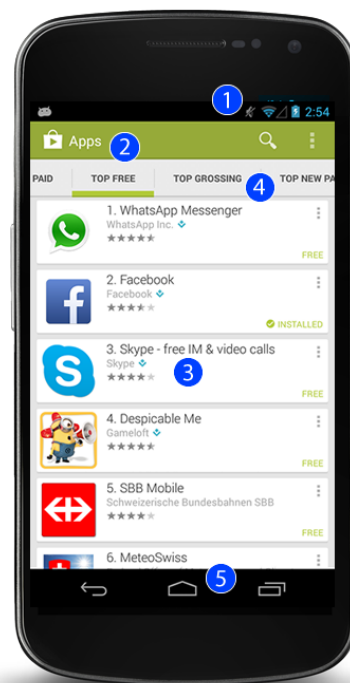


Figure 3.1: Android Play Store

3) Content Area

It's the area where the content is displayed

4) Tab Bar

The bar provides links to the different subsections of this app.

5) Navigation Bar

Only on Android, the navigation bar provides the device navigation controls: Back, Home and Recents. There is no equivalent on iOS. On tablet navigation and status bars are combined into a single "combined bar".

3.1.2 Common iOS UI

Here is just a small summary of the iOS UI for those not familiar with this OS. More details can be found on <http://developer.apple.com/>

1) Status Bar

The status bar contains information about the device such as time, battery charge, network connexion and signal strength, alarm information.

2) Toolbar

The Toolbar contains 2 kinds of elements: the navigation controllers which enable navigation through an information hierarchy and items specific to the context of the current view. iOS devices don't have native navigation specific bar. Therefore, navigation action are made with this Toolbar.

3) Content Area

Like in Android, it's the area where the content is displayed

4) Tab Bar

A tab bar appears at the bottom edge of a screen or view and gives people the ability to switch between different sub-tasks, views, or modes.



Figure 3.2: iOS App Store

3.2 User Habits

As you saw in the previous chapters, iOS, Android and other OS have significant differences in the GPU ¹. This involve differences in the user behaviours.

For example, one difference between iOS, Android and Windows Phone is the Back/Quit button. Android Phone provides a button at the bottom edge of a screen, displayed by the OS. Smartphones with Windows Phone supply a touchable area at the bottom of the screen. For iOS there is no back button.

For that, you must design a back solution on iOS. The iOS user is in the habit of having a back button on top left corner in the mobile application. For android and WP8, displayed a back button on the top left is useless and may seem odd. But when you develop, you have to be sure that the OS back button is supported ². For example, if you load contents without to load a new page, you should change parameters in the URL hash tag to obtain a new entry in the history and to have the possibility to use the back button correctly.

3.2.1 Overall Appearance

This kind of problem is not only for hybrid apps, it also happens on native apps. You must take these differences into consideration when you choose the GUI. They are different points of view to solve the issue.

One solution consists in having a platform-independent GUI. The application has the same GUI on all mobile OS. It's the case in most of full screen games. They don't have to implement standard elements such as, menus, buttons,...

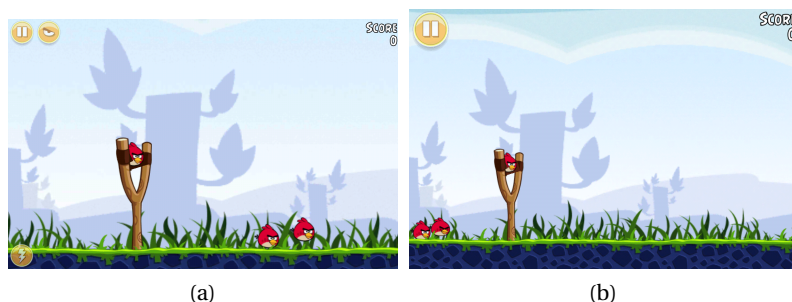


Figure 3.3: Angry Birds game on Android(a) and iOS(b)

In the idea to have the same GUI on different devices and if the app needs to use standard elements. You can design you own GUI or use existing frameworks. In this case, you must be especially careful with user habits.

¹Graphical User Interface

²based on history for Apache Cordova

On the one hand you can mix GUI elements from different OS and try to find compromises. It's the case of the UBS Mobile Banking app based on PhoneGap.[AdNovum(2012)]

On the other hand you can try to design something new or different from usual. It's the case of the Facebook app that was one of the first mobile apps to use an unusual menu. The main content slides out by button tap or drag right.

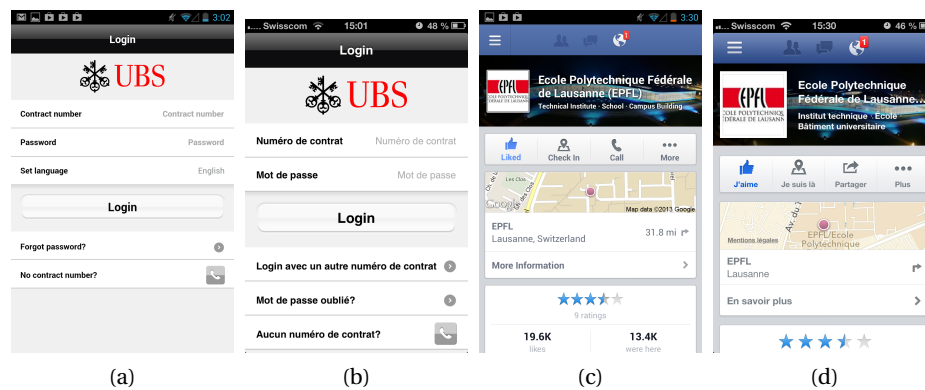


Figure 3.4: UBS Mobile Banking Android(a), iOS(b) and Facebook app Android(c), iOS(d)

If having exactly the same GUI on different device is not your priority. It is recommended to personalize the apps for the given OS. The user will find a known environment with personalized elements. A majority of apps are in this case, such as Dropbox, Gmail, Shazaam,...

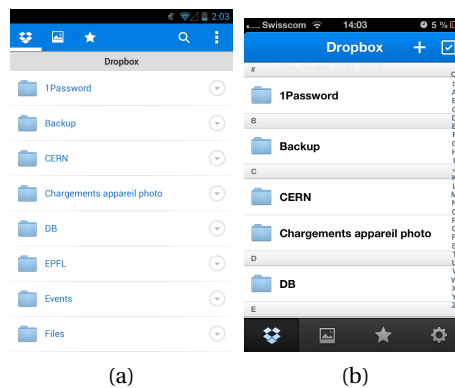


Figure 3.5: Dropbox for Android(a) and iOS(b)

3.2.2 Interactive Elements

Is important to have a GUI adapted for the user, but it's also important to have the usual animations for the interactive elements too. The elements like menus don't work the same way.

For example, Android's menus are displayed on the front of the app and hide a part of the content. On iOS it's almost the opposite, the content slides on the right and reveals the menu.

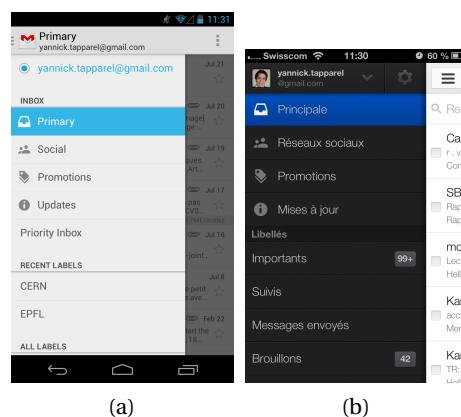


Figure 3.6: Gmail menu for Android(a) and iOS(b)

3.3 Native Feel

For an application, the waiting time makes the user experience unpleasant. Ideally, the user should not spend time to wait.

In practice, there are three important times based on the human perceptual abilities. [Jakob(1993)] These times are the same for thirty years. [Miller(1968)] [Card and Mackinlay(1991)]:

0.1 second : The user feels that the system is reacting instantaneously if the delay is 0.1 second or less.

1.0 second : With a time less than 1.0 second the user has the feel that he is navigating freely without to unduly wait for the computer. For the user the 0.2-1.0 delay is normal when the computer is working.

10 second : The delay higher than 10 second are in general not acceptable.

3.3.1 Improve Hybrid App Feel

It is very important to respect the delays cited above. In the case of Hybrid apps, it is not trivial to react instantaneously and to load content in less than one second. For that, here is some tips:

Use CSS animations

CSS animations are hardware-accelerate by offloading them to the GPU³. Normally, hardware acceleration is an important milestone to handle details 3D modeling but in our case we want our HTML element appear smooth and animate smoothly via GPU.[Hales(2011)]

Webkit provides hardware acceleration for some CSS animations. When it is possible, think to use CSS animation instead of JavaScript animation. With this solution, animations could run smoothly.

Not all CSS properties trigger Hardware Acceleration.[Ubl(2011)]. The mainly aspects of your document can be accelerated by the GPU are:

- General layout
- CSS3 transition and 3D transforms
- WebGL 3D and Canvas Drawing

If you want to know which elements are Hardware-accelerated use this command to run the iPhone simulator.

```
1 $CA_COLOR_OPAQUE=1 /Applications/Xcode.app
  /Contents/Developer/Platforms/iPhoneSimulator
  .platform/Developer/Applications/iPhone\
  Simulator.app/Contents/MacOS/iPhone\ Simulator
```

Hardware-Accelerated elements become red and not are green.

Don't use click event

Usually for a button, a div, or any other elements, we trigger event by click event. Like this:

```
1 <button onclick="myEvent()"></button>
```

or with jQuery:

```
1 <div id="myDiv">Click Here!</div>
2 <script>
3   $("#myDiv").on("click", myEvent())
4 </script>
```

³Graphics Processing Unit

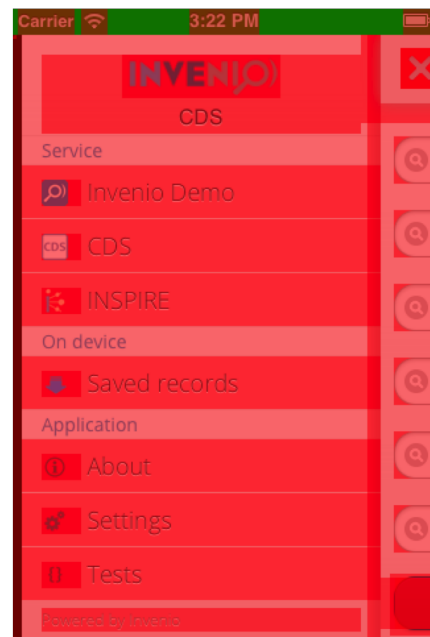


Figure 3.7: Hardware-Accelerated elements are red

Chapter 3. Requirements for Mobile Application Development

But in these both approaches have a significant weakness. The mobile browser will wait almost 300ms . The reason is that the browser is waiting to detect if you are actually performing a double tap.[Fioravanti(2011)]

As we saw, if the latency is more than 100ms the user will notice and be disturbed. We cannot afford to wait 300ms. With this delay the application feel more laggy. Moreover, we generally don't use double click/touch event on mobile device. There are several solutions to solve this problem:

- **touchStart** and **touchEnd** can be used to simulate a touch event. With this solution you have to be sure that the element where you start to touch and where your finish is the same.
- **vclick** is a virtualized click event handler provide by jQuery Mobile. This solution can be easily used if you already use JQuery Mobile framework in your project. They recommend using vclick with caution anytime the action being triggered has the possibility of changing the content underneath the point that was touched on screen. [jQuery Mobile API(2013)]
- **FastClick plugin** is a simple and popular library for eliminating the 300ms delay under the MIT licence developed by FT Labs, part of the Financial Times.

Use Phonegap features

When a feature exists on both JavaScript and Phonegap/Cordova prefer the Cordova method. Your feature is interpreted by the web viewer in the JavaScript case. Contrariwise, it is partially executed in native code with the Cordova solution. Moreover, with Cordova the feature is often most complete and customizable. For example, with confirm box, the title is definable with Cordova, but that is not the case with JavaScript.

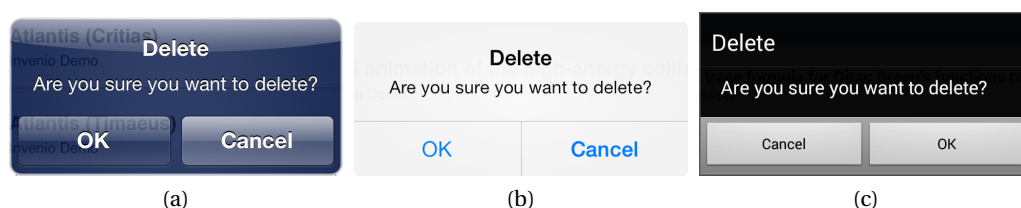


Figure 3.8: Confirm box on iOS6(a) iOS7(b) Android(c)

Consider smartphone displays difference

To improve the native feel, don't forget that you work with a smartphone screen and they have two main particularities which must be considered:

- **There are much variety of pixel density.**
While iOS support two type of resolution retina screen(2x) and regular screen(1x), Android support XHDPI(2x), HDPI(1.5x), MDPI(1x), LDPI(0.75x). You have to adjust you

graphic elements to be well supported by the screen. More explanation are on the "Responsive Web Design" chapter.

- **They are touchable.**

With the finger, the user is less precise than with a mouse. Interactive elements should be enough big to be easy target with a finger. Google recommends using button with 48dp ($\approx 9\text{mm}$). It's a compromise between big size and overall information density.[Google(2012)]

3.4 App Store Review Guidelines

"If you're coming from the web, you need to make sure that you give people an iOS app experience, not a web experience. Remember, people can visit your website on their iOS devices using Safari on iOS."

—"App Design Strategies", Apple Inc.[Apple(2013b)]

During the first years, hybrid apps have not been accepted on the app store. Two years after Apple opened this first app store, the company allowed the developers to provide hybrid applications. As for native applications, apple control all applications before publishing and can reject some applications. Apple does not refuse an application because the UI is built using HTML but can reject it if the application does not respect some rules.

Apple rejects applications that do not[Trice(2012)]:

- have a user experience that feels like an "app"
- feel "at home" in the iOS ecosystem
- offer a differentiation from a mobile web experience

More precisely, Apple's "App Store Review Guidelines" has additional tips relate to HTML-based experiences, including the following tips:

- "2.12: Apps that are not very useful, unique, are simply web sites bundled as Apps, or do not provide any lasting entertainment value may be rejected"
- "10.3: Apps that do not use system provided items, such as buttons and icons, correctly and as described in the Apple iOS Human Interface Guidelines may be rejected"
- "10.6: Apple and our customers place a high value on simple, refined, creative, well thought through interfaces. They take more work but are worth it. Apple sets a high bar. If your user interface is complex or less than very good, it may be rejected"
- "12.3: Apps that are simply web clippings, content aggregators, or a collection of links, may be rejected"

The Google play policies are less restrictive concerning the UI but ask that the app must have a good stability, performance and visual quality.[Google(2013a)]

To summarize if you want to publish an app on an apps store you need to get the app ready and obtain a good visual quality. The feel is very important. Users should not feel that is a website.

3.5 Responsive Web Design

As explained in the previous paragraph it is mandatory to obtain a good feel with the app. The RWD⁴ can give a good approach.

The notion of Responsive Web Design groups several tenets, technologies and design approach that aims to provide an optimal viewing experience and interaction across a wide range of device resolutions, screen densities.

The RWD concept is described in three Key Elements. [Marcotte(2011)] [jQuery Foundation(2013a)]

3.5.1 Key Elements

- **CSS media queries**, used to target styles to specific device characteristics such as screen width break-point or resolution.
- **A fluid grid** specifies elements and widgets in flexible units with the goal of making them flow to fill their containers. In some cases, the fluid grid is not considered necessary to have RWD.
- **Flexible images and media**, are also sized in relative units so they re-size to fit within their containers.

RWD is not only for mobile applications, but can also be used for all technologies which use HTML, such as websites, emails,...

⁴Responsive Web Design

4 Design And Implementation

In this chapter are discussed the most relevant parts of the project's implementation. First, we talk about the global architecture of Invenio Mobile, highlighting its main components. Then, the user interface and the functionality is deeply studied, with a special section for the tablet UI. The last parts present the files organisation and the optimizations specific for Invenio Mobile.

4.1 Elements

As mentioned in the previous chapters, for the Invenio mobile application, I focused the project on a hybrid application. This hybrid app is built with Apache Cordova and jQuery mobile as UI framework. To operate the project needs three main components: a mobile phone, an app and the server-side.

4.1.1 Mobile Phone

With Apache Cordova, which is an abstraction layer, the project should works with a wide range of mobile devices. By pragmatism the project focus on Android and iOS devices who together represent 95.3% of smartphone sales.[Worldpanel(2013)]

It is not very hard to add other OS support such as Windows Phone or Blackberry but the UI Design needs to be adjust in order to be consistent with the given OS guideline.

4.1.2 App

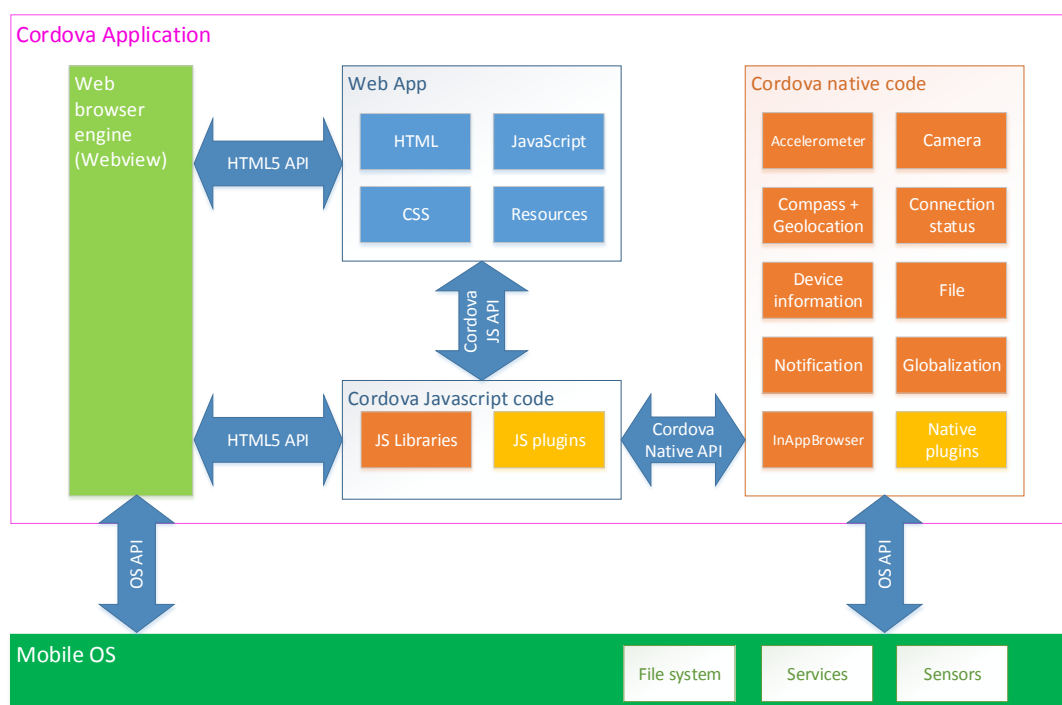
The app should be executable in local without internet connection but not all function are accessible off-line. The application is in large part the same on each device except for some graphical elements which are personalized in function of the OS or screen size.

4.1.3 Server Side

Invenio is a software used by several institutes who personalize the software for their needs. All of these institutes use more or less recent versions of Invenio but not all the same. To be compatible with the wide range of Invenio instance, the server side modifications must be as light as possible and compatible with most of them.

Invenio has the possibility to define some output formats. For example, we can request to get the result of the query in HTML, MARCXML, BibTeX, etc. To communicate with the mobile application the project supplies two new output formats which are computed to be directly displayed on the mobile screens: mobile brief (mobb) and mobile details (mobd). Mobile brief contains only the strict necessary to display the search results. Mobile details contains all information for a given recode.

4.2 Components



The Invenio mobile app is built with Apache Cordova. The components of the app are directly linked to the Apache Cordova structure. The components in the mobile application are :

- The **Cordova native code** is supplied by Apache. It is compiled in native application and provide to Invenio mobile the possibility to use native features. The project is designed

to work with the version 2.9 of this development framework. The Cordova native code depend on the OS

- The **Cordova JavaScript code** is the bridge between the HTML5/JS code and the Apache Cordova native code. In Android, the JS to Cordova native communication is set to the prompt dialog. In iOS, communicate with the native APIs through an iframe. The JS calls are stored in a JS queue that is read and executed by the native component. [Puthraya(2012)]
- the **webview** is generated by Apache Cordova, it's the visible part of the app. iOS and Android use Webkit but the render is not the same in both cases. Some HTML iOS feature are not supported Android and vice versa.
- The **web application** is the main part of the project. It is the essential code in HTML5, CSS and JS of the project.
- **Plugins** can be added to the project to supplement Cordova features. Plugins are composed of JS and native code. Plugins for Phonegap and Cordova are the same. People propose a lot of open source plugins such as bluetooth, extract zip files, NFC, Speech recognizer, barcode scanner,... In the case of this project, I had a file opener plugin which opens popular file (pdf, word, jpg, videos,...) for android. [github(2013)]
- The file system, some services and sensors are managed by the **Mobile OS**. It execute the application and provide information that Apache Cordova needs

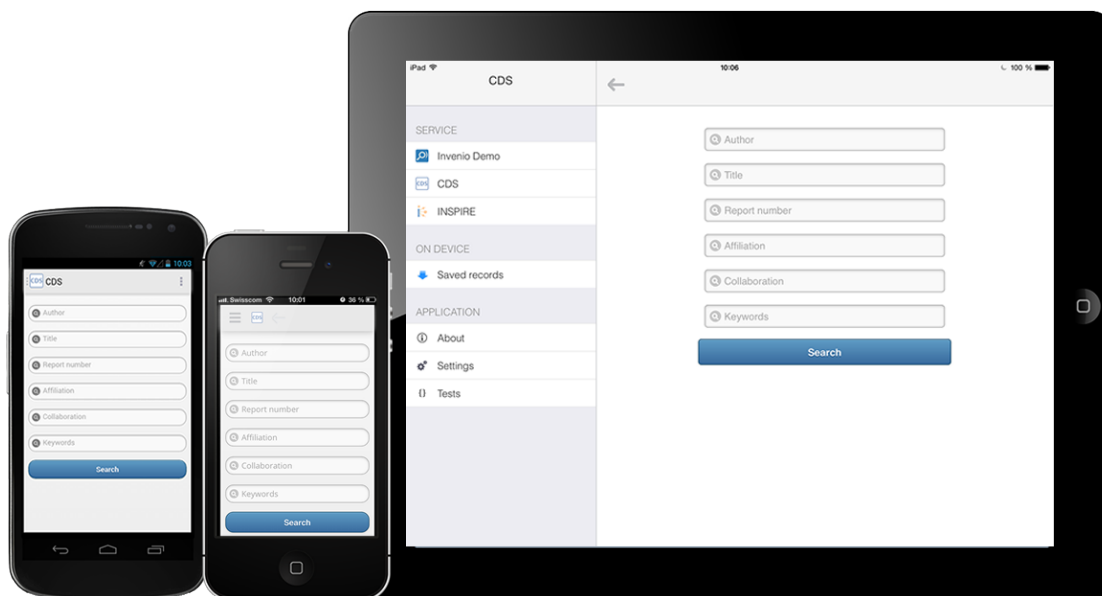
4.3 User Interface And Functionality

In the interest of saving time and not reinventing the wheel, the user interface was designed with jQuery Mobile. After some tests and musings about the user interface, I came to the conclusion that it is not ideal to have exactly the same UI on different sort of devices.

The project keeps some jquery mobile elements and look almost similar in the whole in Android and iOS while maintaining OS specific. However, the project include a version for iPad with a beta of iOS7 that is pretty different with the smartphones Android and iOS version.

For the project we want to keep the advantage of hybrid apps to not need to be programmed for each OS or device and to have visual differences. For that only almost 5% of the source code of this app change between the different versions.

iOS and Android don't have the same set of font. You can't count on standard font such as Arial or Helvetica being present. Add you own font to the project. To help you, Google provides a website with a list of open and free font. <http://www.google.com/fonts/> For the project I use "Open Sans" by Steve Matteson. Apache License, version 2.0. Open Sans was optimized for print, web, and mobile interfaces, and has excellent legibility characteristics in its letterforms.



4.3.1 Search View

The main goal of this app is to search documents. After the splash screen the user arrives on the main view which permits to search record.

Contrary to web search engine like google, most of the time Invenio's users know what kind of information they key. Sometime, they are searching a book from a given author, a record with a specific word in the title, or all information about a topic. To obtain pertinent result, the user is encouraged to key the request in the good request categories.

Whatever the kind of smartphone or tablet used, the user sees directly all possibilities and can search without scrolling. When the keyboard is displayed on the screen and hide a part of the screen, a key gives the possibility to search without to scroll on the bottom and click to the search button.

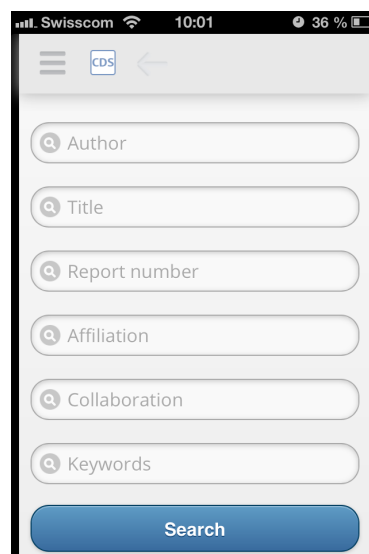


Figure 4.1: Search view

4.3.2 Toolbar

The toolbar (or action bar) depends on the OS. Both don't have the same button and its actions are not the same neither. For example. iOS has a back button not displayed on Android.



Figure 4.2: Android action bar(a) and iOS tool bar(b)

4.3.3 Menus

The left menu is a jQuery Mobile menu, it is also personalized for each OS as mention in the paragraph 3.2.2. The contents, the displays, and the animations are not the same. The app allows the user to work with different services using Invenio (Invenio Demo, CDS, INSPIRE on the following pictures).

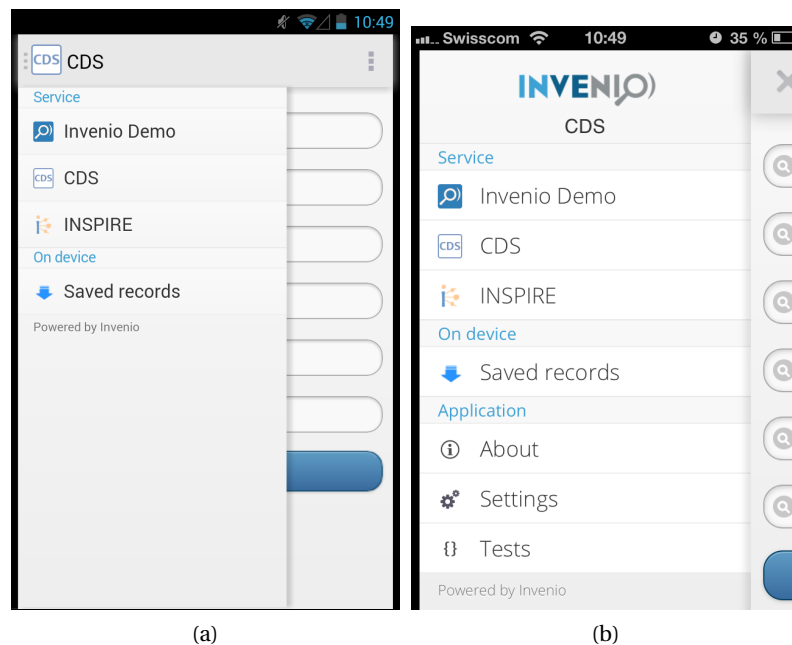


Figure 4.3: Android action bar(a) and iOS tool bar(b)

This menu can be opened by clicking on the selected service icon on Android or the menu icon on iOS in the upper left. The menu can also be opened by sliding the finger from left to right. About, Settings and Tests items are not in this menu on android but on the contextual menu.

4.3.4 Search Result View

As soon a request is made, the query appears in a blue tape at the top of the screen who is displayed beyond the scope of the view. The query can be edited and changed. An auto-completion tools help the user to correctly write his research. The request can be combined with logical elements. For example: "TITLE:science AND NOT AUTHOR:darwin"

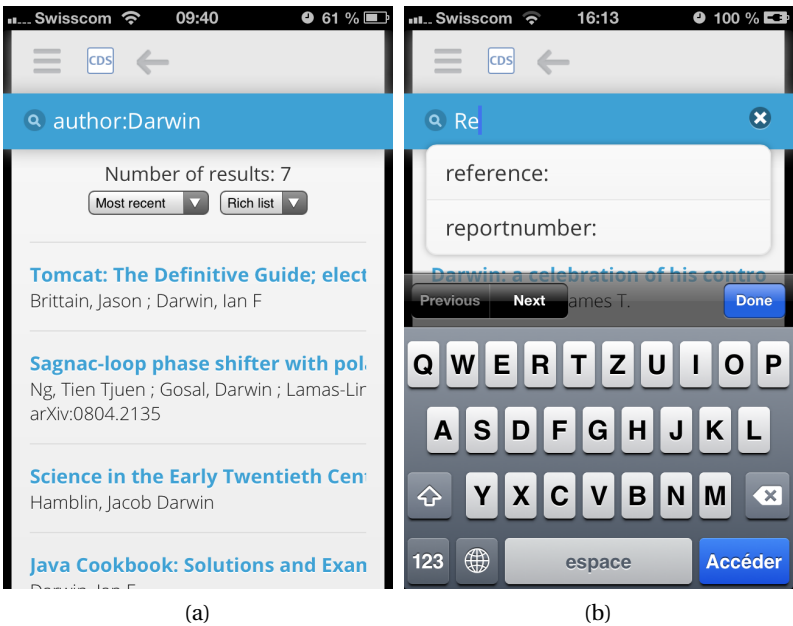


Figure 4.4: iOS search result(a) and search autocomplete(b)

4.3.5 Record View

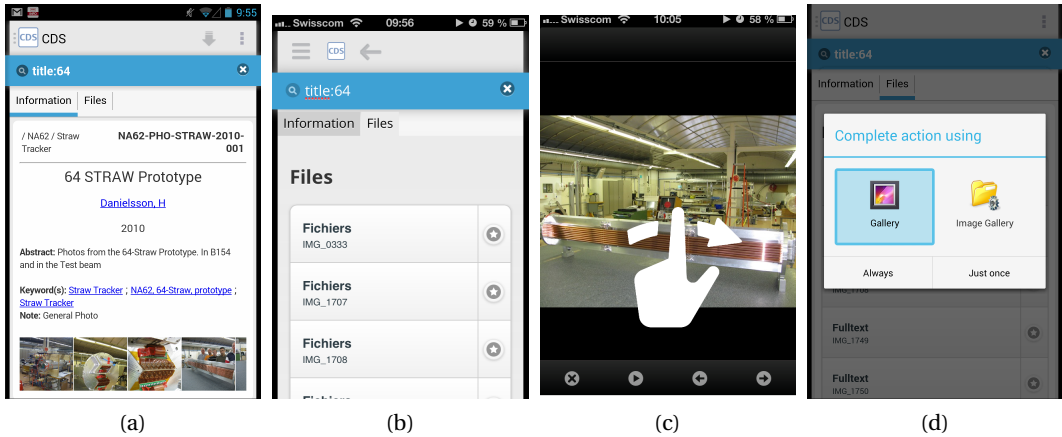


Figure 4.5: Information(a) and Files(b) tabs, photo slideshow(b) and files dialogue box(d)

The records are displayed with tabs. The first tab "Information" is displayed on each record, the other tabs depend on the content. Users can see texts, pictures slideshow and videos directly on this first tab.

All tabs except the "Files" tab can be downloaded to be displayed later by clicking to the arrow on the top. The arrow becomes blue if the tab is already downloaded. On the "Files" tab, files can be opened by clicking on the name or can be downloaded off-line by clicking on the

start near the file name. The project uses a Cordova plugin who permit the user to chose with which app he want to open the file.

4.3.6 Saved Records View

When a pdf, a video, a picture, or any other record's contents are downloaded, the user can consult it in this view. The user can look every record from any domain order by name. The user can delete only one file or directly a whole record with many files. To delete the user has two possibilities: By sliding his finger on the given file/record or by selecting files with the checkbox and clicking on the trash.

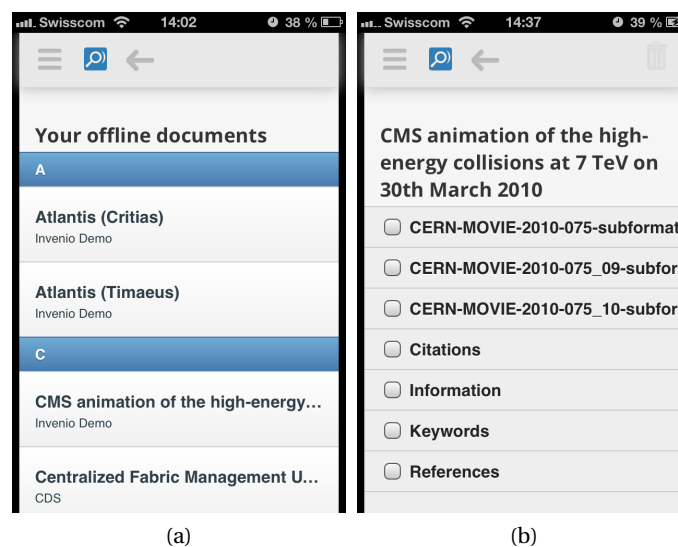


Figure 4.6: Saved records (a) and detailed items(b)

4.3.7 Other Views

The app contain other views such as :

- "Settings" gives the possibility to add and manage new domain, to enable a debug mode, clear cache, and define some options...
- "About". Just displays some information concerning the app.

4.4 Tablet UI

Inspire is a high-energy physics literature database at CERN based on Invenio. The following statistics of the inspire mobile user community highlight the importance of the iPad.

Table 4.1: Mobile user community in one month- INSPIRE, April 2013

OS	Visitors
iPad	6866
Android	4560
iPhone	4294
iPod	145
BlackBerry	121
Windows Phone	47
Bada	44
Symbian	40
Other	41

For this reason, I decide to implement an iPad version of Invenio Mobile. The tablet UI is rather different. The screen is large enough to show all information. Consequently, the swipe menu is not requisite. The screen is divided into two parts.

The design and colors are copied of what it can be found on the first betas of iOS7. The overall looks is inspired by the iOS7 app "Mail". The tabs are the same of what we can see in the app "Calendar". The functionality are the same as the smartphone versions.

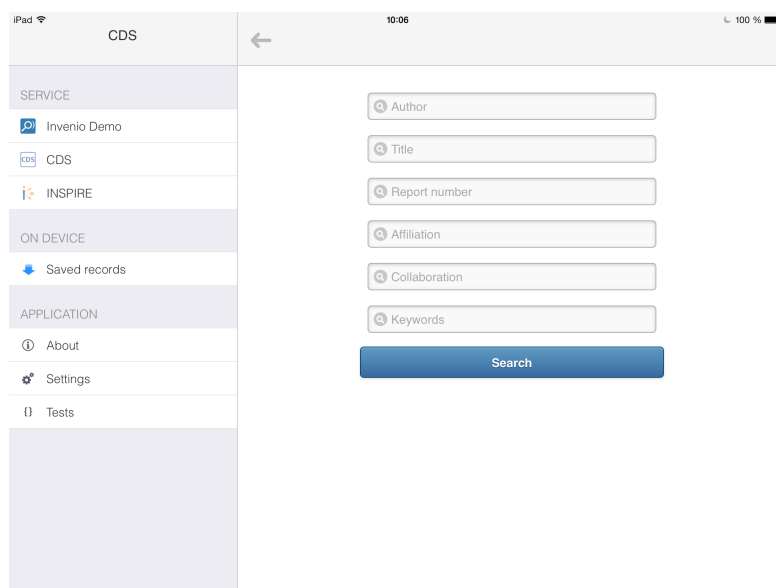


Figure 4.7: iPad home view

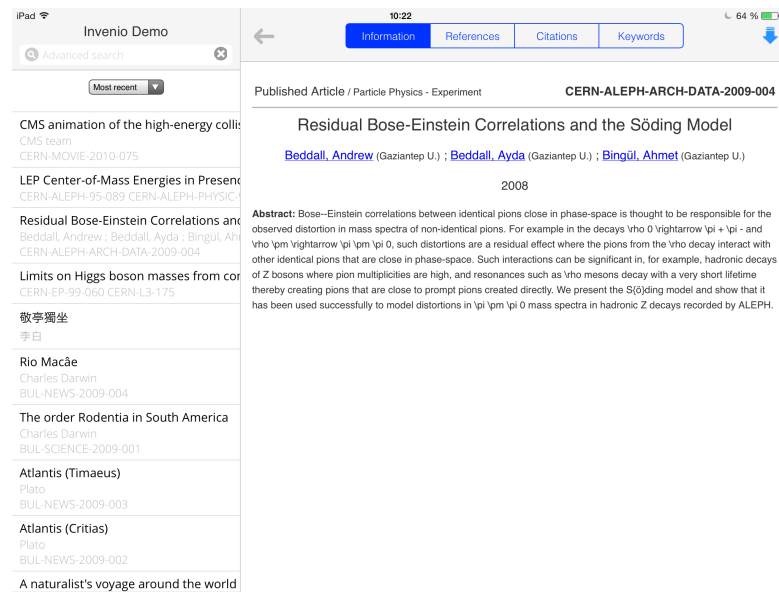


Figure 4.8: iPad record view

4.5 Web App stucture

4.5.1 Files Organization

The app is a web app wrapped in a Cordova native app. The main part of the app consists of web files. The file organization which composes the app is made in the following way:

- **index.html** is the first file loaded. It's just a redirection to the correct index for the good device.
- **index.iphone.html**, **index.android.html** and **index.ipad.html** are three different categories of devices. These files with the linked CSS are the only differences between the different versions of the app. These files contain the UI of the specific devices to correspond to the user habits.
- **pages** is the folder who are the HTML content of the several views. All views have a file in the folder such as : home.html, offlineFile.html, record.html, search.html, settings.html, and so.
- **img** is the folder who are the images of button, icon, ...
- The **css** folder contains
 - **Invenio-mobile.css** who are the main CSS. It's the common CSS with all mobile device and contain the main CSS parameters.
 - **Invenio-mobile.iphone.css**, **Invenio-mobile.android.css** and **Invenio-mobile.ipad.css** which are the css linked with the specific index files.

- A **font** folder with different the Open Sans fonts with different font weight.
- **jquery.mobile-1.3.0.min.css**, the css for jQuery Mobile.
- **photoswipe.css** for the photoswipe JavaScript tool.
- **js** is the folder with the functionalities for specific for Invenio Mobile:
 - **Invenio-mobile.js** is the main JavaScript file for the proper functioning of Invenio Mobile. It contains a set of functions usable in the pages such as functions to load new pages, to read and write files offline, to parse and clean the pages, to communicate with servers,...
 - **Invenio-mobile.config.js** contains global variable with some initial parameters.
 - **Invenio-mobile.debug.js** is set of tools to help the developer and display information for optimization
 - **Invenio-mobile.domain.json** is a JSON file with the list of domains accessible by default
 - **Invenio-mobile.initialization.js** takes care of the first initialization of the app. It loads the initial parameters, enables click and touch event listeners, loads auto-complete, ...
 - **Invenio-mobile.settings.js** provides tools to manage the personal settings of the users.
- **js_external** is the folder with the external JavaScript tools and libraries use in the project:
 - Android and iOS **Cordova** 2.9 JavaScript code
 - **jQuery** 1.9.1 and **jQuery Mobile** 1.3.0
 - **photoswipe** 3.0.5 for jQuery for images galleries and its dependencies.
 - **fileopener** is a free and open source Cordova plugin for Android to open pdf, photos and video in native apps

4.5.2 Execution

The execution of the mobile application can be summarized in the following way. There are four different steps:

1. The "**pageready**" event starts the initialization
 - The Cordova native code run index.html
 - index.html loads Invenio-mobilie.config.js and redirects the initialisation to the correct index file like index.android.html or index.iphone.html.
 - When the specific index file is loaded and Cordova become ready to work, Cordova triggers a "pageready" event.

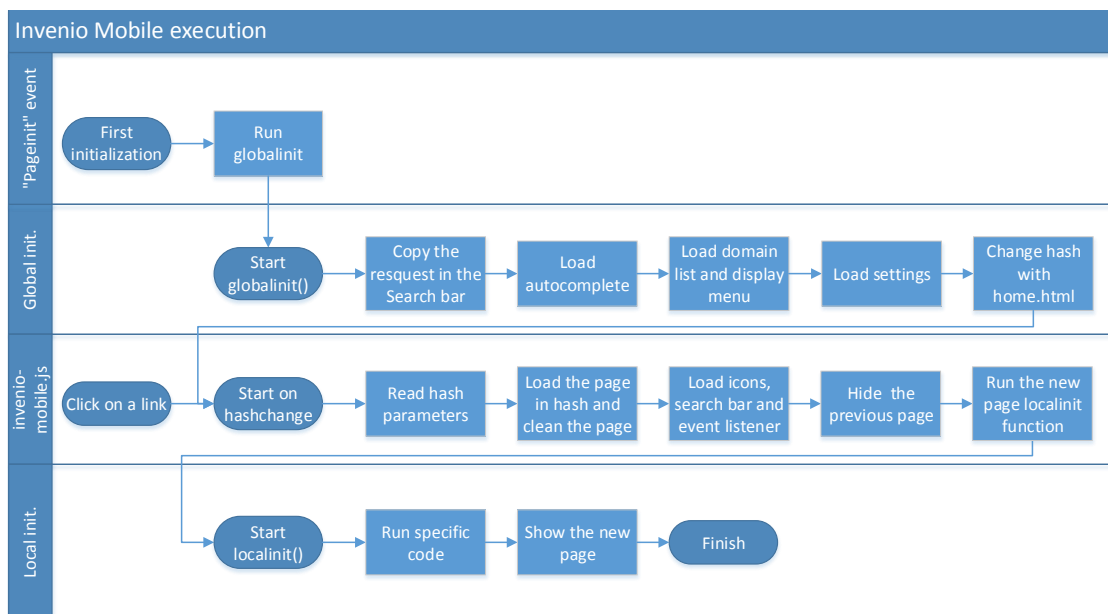


Figure 4.9: Invenio Mobile execution

2. There is a **global initialization** common with all pages. At the end, the url hash is changed with the parameter of the home page.
3. **Invenio-mobile.js** loads the HTML page in parameters.
4. To finish the app executes the **local initialization** specific for this HTML page.

4.6 Off-line Records

If a user downloads records, its data are saved in the smartphone in a "Invenio" folder. For each record saved on the phone, the app does a folder named with the references of the record and save all information in this folder.

- Files such as video, photos, pdf, ... are directly copied in the folder.
- Pages such as information, references, citations are saved in HTML files and are ready to be show.
- In addition, a JSON file is create with the title and domain information.

For example, if a user wants to save the record 113 from Invenio-demo.cern.ch with the information and the references pages and one video, it will obtain the following result in the the folder /Invenio/demo-next113/:

- CERN-MOVIE-2010-subformat-big.jpg

- record_page_Information.html
- record_page_References.html
- record_info.json :

```
1 {
2     "title" : "CMS animation of the high-energy collisions at 7 TeV"
3     "domain" : Invenio Demo"
4 }
```

4.7 Configurations Files

There are two configuration files. The first one is a read-only file with the initial configuration and the second is editable and contains the personal settings of the user.

Configuration files are written directly in JSON. I made the choice of JSON because it is light and well supported by JavaScript without plugins or external libraries.

Here is an example with a part of Invenio-mobile.domain.json that contains the list of domains accessible by default:

```
1 {
2     "domain": [
3         {
4             "id": "demo-next",
5             "domain": "Invenio-demo-next.cern.ch",
6             "favicon": "favicon_Invenio.png",
7             "name" : "Invenio Demo",
8             "enable" : true,
9             "record": {
10                 "display_file": true,
11                 "display_video": true,
12                 "allowed_file_format":["gif","pdf","jpg","jpeg","txt","webm"],
13                 "offline": true,
14                 "tab": [
15                     {
16                         "text": "References",
17                         "url": "/references"
18                     },
19                     {
20                         "text": "Citations",
21                         "url": "/citations"
22                     },
23                     {
24                         "text": "Keywords",
25                         "url": "/keywords"
26                     }
27                 ]
28             }
29         },
30         {...}
31     ]
32 }
```

4.8 Optimizations

There are two kinds of optimization in the project. There are some optimizations to reduce the data flow (to reduce the size and the number of request) and some optimizations to lighten the user interface. In addition to the solutions presented in section 3.3. some specific solutions are explain in this section.

Ajax page loading

The different views of the app are loaded with Ajax. Only the content is changed without refresh the page. The fixed canvas is not refresh and for this part the jQuery Mobile render is done only on the first load.

Ajax request in cache

The requests from the device are process by Ajax. An Ajax cache is enable and requests already done are executed only one time.

Search result is loaded as needed.

The search result is a scrollable list of record. To reduce the amount of transmitted data, only the first 25 matches are loaded. When the user scroll near the bottom, the next 25 elements are loaded and so on.

Download in background and cache.

When a user open a file, this one is stored in cache for the next 24 hours. The download happens in the background. The user can read other record. The file is automatically opened when the file is downloaded. The user can also abort the download if he wants.

Record tab in cache

To facilitate the navigation between tabs in a record and to reduce the data flow, only the unvisited tabs are downloaded. Its HTML render is saved in the session cache. For already visited tabs, the content is directly displayed form the cache.

Reduce the number of jQuery Mobile items

jQuery Mobile is really convenient to help the developer with UI elements, but they are particularly resource intensive. When it was possible, I avoid jQuery Mobile and implement light elements in very long view(e.g. the search result view).

5 Results And Conclusion

5.1 Invenio Mobile

The result of this project is the "Invenio Mobile" app in three versions : Android, iPhone/iPod and iPad. The application contains almost 4000 lines of source code. Only 5% of this source code changes in the different versions with respect to native apps where all the source code needs to change.

Compared to the original goals, this mobile application implements a majority of the desired use cases. The users can search records, save pages and documents for future offline access, display pictures and pdf, play videos, and switch between the Invenio sites.

The user interface is customized for each of the supported operating systems. Screenshots of Invenio Mobile can be seen in chapter 4. The UI is adapted from smartphones with 3.5-inch display to the tablets with 10-inch screen.

5.2 Hybrid App

Besides the actual mobile application, which is an asset in itself, this project has highlighted the feasibility of hybrid apps but also the necessity of devoting time to optimize the source code.

Mobile development is a constantly moving target. Regularly there are new mobile operating systems, such as Ubuntu Touch OS, Firefox OS, BlackBerry 10, iOS7 or Jelly Bean 4.3 which have made the news this year. At the beginning, the new features are only accessible with native APIs but these ones are quickly added by the open-source community for hybrid apps.

With the arrival of the new OS, fragmentation won't go down and solutions must be found. The hybrid approach can provide a solution on condition that the slowness of HTML5 can be reduced. At the present time, we can use the hybrid approach only for a limited number of situation.

The selected framework¹ is very complete, but causes latency which need many optimizations to avoid these troubles. Very young promising frameworks like Lungo, Ratchet or Fries could be a solution.

During my internship, other teams at CERN developed mobile applications and had similar questions concerning the approach. They arrived at the same conclusion to use PhoneGap/-Cordova and jQuery Mobile. It's the case of the CERN Open Days app².

5.3 Futures Outlook

Among several possibilities for improving Invenio Mobile based on the obtained assets, I would like to mention these three points:

- **Invenio API.** At this time, Invenio APIs being developed by the Invenio team. They could provide good tools for the Invenio mobile app which can help to synchronize tags, baskets, or annotations.
- **Notification.** With Apache Cordova it is possible to run applications in the background and to use notifications for iOS and Android. It can be very interesting for the user to have the possibility to receive an alert on the device in some situations. For example, when a loan period has expired (or is about to expire), when user is requested approval in a workflow, when a new document is added to a specific collection, or when a given search query matches a new record.
- **Windows Phone 8.** For this project, I focused on Android and iOS, but some other OS such as Windows Phone 8 are growing. When this OS will have a significant part of the mobile market, the project will need to be adapted. The current project should works on these devices but the UI will have to be rethought for the given OS.

To conclude, Apache Cordova and other hybrid frameworks don't stop evolving. It will be very important to follow these evolutions if we want to enhance user experience with Invenio on modern mobile platforms.

¹jQuery Mobile 1.3.1

²<http://opendays2013.web.cern.ch>

6 Bibliography

6.1 Books

- [Card and Mackinlay(1991)] George G. Card, Stuart K. a nd Robertson and Jock D. Mackinlay. *The information visualizer, an information workspace*. CHI '91. ACM, New York, NY, USA, 1991. ISBN 0-89791-383-3. doi: 10.1145/108844.108874. URL <http://doi.acm.org/10.1145/108844.108874>.
- [Jakob(1993)] Jakob Jakob. *Usability Engineering*. Morgan Kaufmann, 1993. ISBN 0125184069. URL <http://www.nngroup.com/books/usability-engineering/>.
- [Kosmaczewski(2012)] Adrian Kosmaczewski. *Mobile JavaScript application development*. O'Reilly, Beijing China Sebastopol, Calif, 2012. ISBN 978-1449327859.
- [Marcotte(2011)] Ethan Marcotte. *Responsive Web Design (Brief Books for People Who Make Websites, No. 4)*. A Book Apart, 2011. ISBN 098444257X. URL <http://www.abookapart.com/products/responsive-web-design>.
- [Miller(1968)] Robert B. Miller. *Response time in man-computer conversational transactions*. AFIPS '68 (Fall, part I). ACM, New York, NY, USA, 1968. doi: 10.1145/1476589.1476628. URL <http://doi.acm.org/10.1145/1476589.1476628>.
- [Wroblewski(2011)] Luke Wroblewski. *Mobile First*. A Book Apart, 2011. ISBN 1937557022. URL <http://www.abookapart.com/products/mobile-first>.

6.2 Technical Articles

- [Cisco(2013)] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2012–2017. Technical report, Cisco Systems Inc., San Jose, CA, February 2013. URL http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.
- [Gartner(2013a)] Gartner. Gartner recommends a hybrid approach for business-to-employee mobile apps. Technical report, Gartner, Inc., April 2013a. URL <http://www.gartner.com/newsroom/id/2429815>.
- [Gartner(2013b)] Gartner. Gartner says smartphone sales grew 46.5 percent in second quarter of 2013 and exceeded feature phone sales for first time. Technical report, Gartner, Inc., August 2013b. URL <http://www.gartner.com/newsroom/id/2573415>.
- [Google(2012)] Google. Developer's guide - client-side storage (web storage). Technical report, Google, October 2012. URL <https://developers.google.com/web-toolkit/doc/latest/DevGuideHtml5Storage>.
- [IBM(2012)] IBM. Native, web or hybrid mobile-app development. Technical report, IBM Corporation, Route 100, Somers, NY 10589, April 2012. URL <http://public.dhe.ibm.com/common/ssi/ecm/en/wsw14182usen/WSW14182USEN.PDF>.
- [ITC(2013)] ITC. Ict facts and figures - the world in 2013. Technical report, International Telecommunication Union, Place des Nations, 1211 Geneva 20, Switzerland, February 2013. URL <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013.pdf>.
- [Signal(2013)] Open Signal. Android fragmentation visualized. Technical report, Open Signal, July 2013. URL <http://opensignal.com/reports/fragmentation-2013/fragmentation-2013.pdf>.

6.3 Conference

- [Ruiz(2007)] Hector Ruiz. Ted : The thinking behind 50x15. In *The thinking behind 50x15*, 2007. URL http://www.ted.com/talks/hector_ruiz_on_connecting_the_world.html.

6.4 Websites

- [50pixels()] 50pixels. Native, web or hybrid apps? URL <http://www.50pixels.com/blog/native-web-or-hybrid-apps/>.
- [AdNovum(2012)] AdNovum. Ubs mobile banking - iphone and android app for e-banking customers, 2012. URL <http://www.adnovum.ch/en/angebot/referenzen/finance/ubs-mobile-banking.html>.
- [Android(2013)] Android. Ui overview, 2013. URL <http://developer.android.com/design/get-started/ui-overview.html>.
- [Apple(2013a)] Apple. App review guidelines, 2013a. URL <https://developer.apple.com/appstore/guidelines.html>.
- [Apple(2013b)] Apple. App design strategies, 2013b. URL <http://developer.apple.com/library/ios/DOCUMENTATION/UserExperience/Conceptual/MobileHIG/AppDesign/AppDesign.html>.
- [Apple(2013c)] Apple. ios ui element usage guidelines, 2013c. URL <http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/UIElementGuidelines/UIElementGuidelines.html>.
- [CanIUse(2013)] CanIUse. calc() as css unit value, 2013. URL <http://caniuse.com/#feat=calc>.
- [Codrova(2012)] Codrova. Apache codrova, 2012. URL <http://cordova.apache.org/>. Apache License, Version 2.0.
- [Daniel Nations()] former About.com Guide Daniel Nations. What is a web application? URL http://webtrends.about.com/od/webapplications/a/web_application.htm.
- [Fioravanti(2011)] Ryan Fioravanti. Google - creating fast buttons for mobile web applications, 2011. URL https://developers.google.com/mobile/articles/fast_buttons.
- [github(2013)] github. Fileopener plugin for phonegap, 2013. URL <https://github.com/markeefb/FileOpener>.
- [Google(2012)] Google. Metrics and grids, 2012. URL <http://developer.android.com/design/style/metrics-grids.html>. Creative Commons Attribution 2.5.
- [Google(2013a)] Google. Publishing checklist for google play, 2013a. URL <http://developer.android.com/distribute/googleplay/publish/preparing.html>.
- [Google(2013b)] Google. Loon for all, 2013b. URL <https://www.google.com/loon/>.
- [Hales(2011)] Wesley Hales. html5rocks.com - html5 techniques for optimizing mobile performance, 2011. URL <http://www.html5rocks.com/en/mobile/optimization-and-performance/>. Creative Commons Attribution 3.0 License.

Bibliography

- [Invenio(2013)] About Invenio. About invenio, 2013. URL <http://invenio-software.org>.
- [JDN(2012)] Thierry Picard JDN. Web application, application native ou application hybride : choix cornélien !, September 2012. URL <http://www.journaldunet.com/developpeur/expert/52314/web-application--application-native-ou-application-hybride---choix-cornelien.shtml>.
- [Jones(2013)] Margaret Jones. Android fragmentation: More os versions, more problems, 2013. URL <http://searchconsumerization.techtarget.com/feature/Android-fragmentation-More-OS-versions-more-problems>.
- [jQuery Foundation(2013a)] The jQuery Foundation. Going responsive, 2013a. URL <http://view.jquerymobile.com/1.3.1/dist/demos/intro/rwd.html>.
- [jQuery Foundation(2013b)] The jQuery Foundation. jquery mobile: Touch-optimized web framework for smartphones and tablets, 2013b. URL <http://jquerymobile.com>.
- [jQuery Mobile API(2013)] jQuery Mobile API. jquery mobile api - vclick, 2013. URL <http://api.jquerymobile.com/vclick/>.
- [Korf and Oksman(2012)] Mario Korf and Eugene Oksman. Native, html5, or hybrid: Understanding your mobile application development options, 2012. URL http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options.
- [Matteson(2011)] Steve Matteson. Google webfonts - open sans, 2011. URL <http://developer.android.com/distribute/googleplay/publish/preparing.html>. Apache License, version 2.0.
- [Nielsen(2012)] Nielsen. Two thirds of new mobile buyers now opting for smartphones, decembre 2012. URL <http://www.nielsen.com/us/en/newswire/2012/two-thirds-of-new-mobile-buyers-now-opting-for-smartphones.html>.
- [PhoneGap(2012a)] Adobe PhoneGap. Adobe phonegap, 2012a. URL <http://phonegap.com/>. Copyright © 2013 Adobe Systems Inc. All rights reserved.
- [PhoneGap(2012b)] Adobe PhoneGap. License, 2012b. URL <http://phonegap.com/about/license/>.
- [Puthraya(2012)] Thejaswi Puthraya. Dissecting phonegap's architecture, 2012. URL <http://agiliq.com/blog/2012/09/dissecting-phonegaps-architecture/>.
- [Rowinski(2013)] Dan Rowinski. iphone and android app design: Developers cheat sheet [infographic], 2013. URL <http://www.developer-tech.com/blog-hub/2013/may/24/infographic-developers-guide-to-ios-and-android-app-design/>.

- [Seven(2012)] Doug Seven. What is a hybrid mobile app?, 2012. URL <http://www.icenium.com/blog/icenium-team-blog/2012/06/14/what-is-a-hybrid-mobile-app->.
- [Systems(2011)] Adobe Systems. Adobe announces agreement to acquire nitobi, creator of phonegap, 2011. URL <http://www.adobe.com/aboutadobe/pressroom/pressreleases/201110/AdobeAcquiresNitobi.html>.
- [Systems(2013)] Adobe Systems. Adobe® phonegap™ build, 2013. URL <https://build.phonegap.com>.
- [Trice(2012)] Adobe Andrew Trice. Phonegap advice on dealing with apple application rejections, 2012. URL <http://www.adobe.com/devnet/phonegap/articles/apple-application-rejections-and-phonegap-advice.html>. Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License.
- [Ubl(2011)] Malte Ubl. html5rocks.com - improving the performance of your html5 app, 2011. URL <http://www.html5rocks.com/en/tutorials/speed/html5/>. Creative Commons Attribution 3.0 License.
- [U.S. Census Bureau(2013)] USA U.S. Census Bureau. U.s. and world population clock, July 2013. URL <http://www.census.gov/popclock/>.
- [W3C(2012)] W3C. Html5 (hypertext markup language 5) logo, 2012. URL <http://www.w3.org/html/logo/>.
- [Wikipedia(2013)] Wikipedia. Wikipedia, 2013. URL <http://en.wikipedia.org/>. Page: Web application, PhoneGap, HTML5, PHP, Hybrid, PhoneGap, HTML5 in mobile devices.
- [Worldpanel(2013)] Kantar Worldpanel. Android regains lead among u.s. smartphone os sales, February 2013. URL <http://www.kantarworldpanel.com/Global/News/Android-Regains-Lead-Among-US-Smartphone-OS-Salesl>.