Mobile Development Technologies Assessment

Author: Optisoft S.r.l.
Partner: Venice Port Authority
Venice, 11/30/2012

Summary

List of figures

1 Introduction

The current document describes the cutting edge technologies to develop an application on mobile devices, in particular It focuses upon the chances offered by the tablet devices compared to smaller device like smartphone.

The document collects also the outcome of a real experience in developing two mobile applications [APC_MOB_REQ], the approach pursued, the designed architecture, the obstacles encountered and the possibilities that, using these new tools in our hands, we can exploit.

2 Abstract

Mobile applications are one of the most hot topics. Most businesses are looking to leverage the explosive growth of apps to help drive marketing, sales, and customer service.

The rapid spreading of these mobile devices, linked to their characteristics of mobility and usability make them the most effective target for the realization of a new series of successful business-oriented applications. However, the business applications are scarce, as opposed to those dedicated to consumer market. The main reason is that it lacks the basic infrastructure designed to quickly implement the functionality required by the common applications management. This happens mainly for the following reasons:

1) Offline Structure: tablets and smartphones are not designed to have a connectivity always guaranteed, so applications must also run offline, that is not necessarily linked to an external server or application or database. You cannot then use the web traditional architectures, but some times we need a framework fully resident in the device.

2) Distributed Database: management applications use a centralized database with high performance. With the mobile architecture, instead, each terminal often uses its own local database that must be synchronized with the central, opening considerable problems of synchronization and management of distributed databases.

3) New user interface model: management applications are now-developed using framework that implement the user interface with a desktop oriented models. On mobile devices has become a standard to produce a model different from the desktop, so it is necessary to rethink the User Interface.

4) Multiple Target: applications of B2B and B2C cannot be limited to only one type of devices. It's necessary to make the application usable by the most common device: at least Apple and Android.

What's the correct choice for mobile development?

We involve in our analysis smartphones development and tablets development.

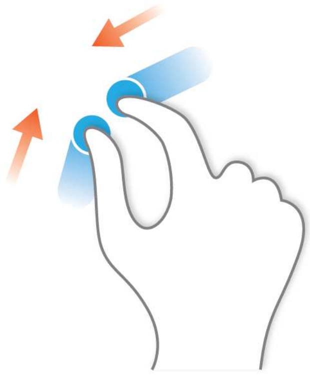
3 Mobile user interaction

The introduction of smartphones and tablets has changed in an unbelievable way how the user interact with software and hardware devices. That aspect has a critical role in the development of mobile software: developers should have in mind from the beginning of the project the user interfaces. For bigger mobile projects, a team of user experience designers work side by side to technical developer in order to implement a quality product.

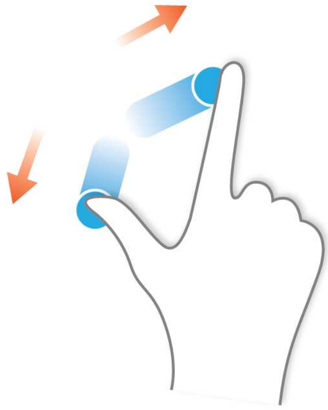
So what are the main differences between computer/laptop and mobile user interaction (let consider first smartphones only)? First of all, generally when a person is in front of a computer he is seated in a predictable environment (house, office), has a large screen, from 14' for laptop and from 17' for desktop computer, and can use a mouse and a keyboard. A person with a smartphone can be everywhere (street, train, seaside), the display is absolutely smaller: Apple iPhone has a display of 3.5', Samsung Galaxy 4.3'; no physical keyboard nor physical mouse.

Let's start analysing what a small display means in term of user interaction. The content of a mobile app page can contain only few information; smartphone are not suitable for high volume of data application, such as books, spreadsheets, movies. Furthermore only one page at a time can be open, multitasking is lower on mobile devices, at least for those tasks that require to interact with user (it is nice to listen to the music while using an app).

The lack of physical keyboard and mouse is filled up with a touchscreen display that allow the user to emulate the click of the mouse and also, with a virtual keyboard, to insert text. As a consequence, the insertion of a keyboard on the screen reduces even more the display available for the user to read the data. In order to make a complete analysis, there are some smartphones that has a physical keyboard and a physical trackball, in particular the Blackberry devices, but represent only a small part of the overall smartphones business and users. Touchscreens have been a niche sector for years, used only in particular application where a physical keyboard is unavailable. With the introduction of the first iPhone, touch displays become a standard de-facto for high end smartphones. A main factor for the introduction of touch technology to the consumer market is the availability of multi-touch displays, that is a touch sensing surface that is able to recognize the presence of two or more points of contact with the surface. This plural-point awareness is often used to implement advanced functionality, not used with traditional mouse, such as pinch to zoom or activating predefined programs.



Pinch to zoom out



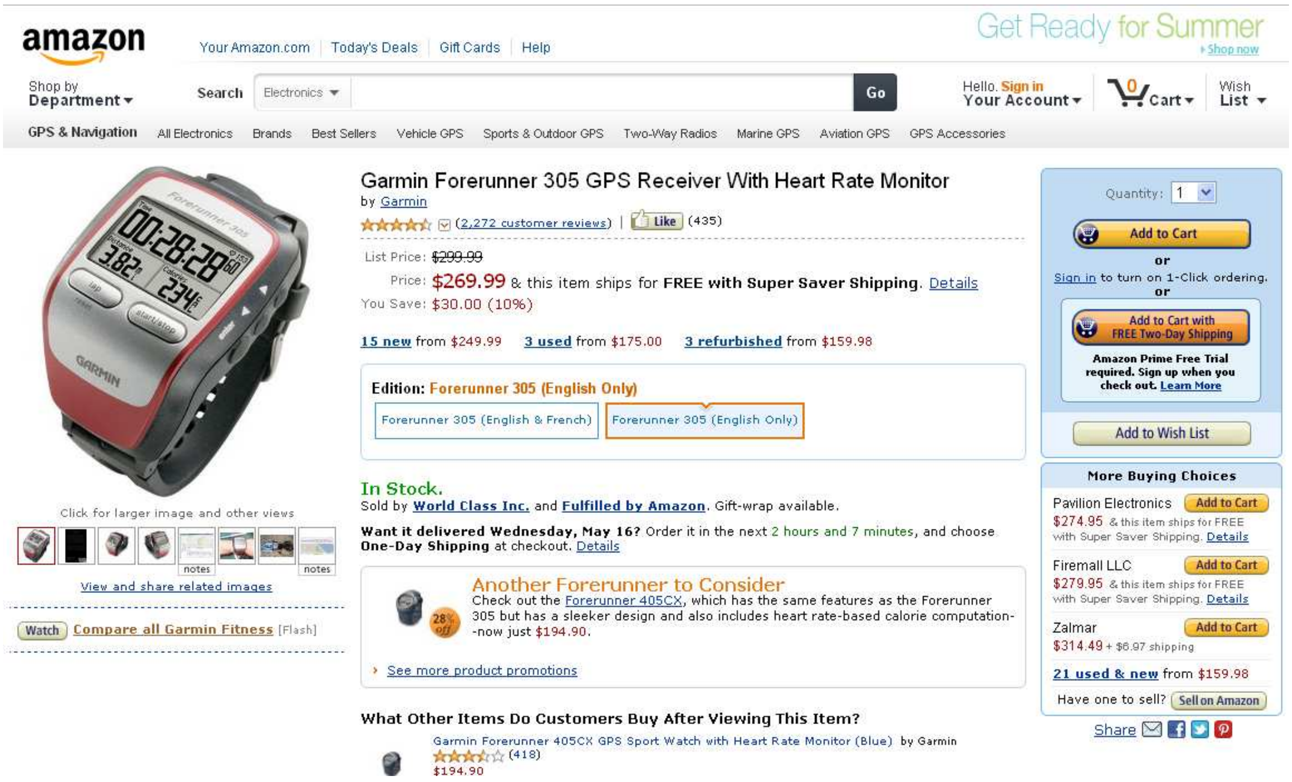
Spread to zoom in

Another new touch gesture is the flick, that is generally associated with the navigation to the previous or next page.

With all these new touch gestures, some common use mouse event are no more available in a touch smartphone. Right-click for example, that in desktop computer is often used, does not exist anymore, or better it is replaced with some other gesture or interaction. Furthermore even the mouse over event, very useful to highlight links in web pages when the mouse passes over them, is not available either.

Another way the user can interact with his smartphone is using the different sensors available with the phones. Camera, GPS, accelerometer, compass, microphone plus different wireless protocol such as Bluetooth or NFC make a smartphone more like a computer than a phone. As an example of how to use these sensors for user interaction, some car games use the accelerometer to understand the smartphone orientation and the user use its smartphone as the steering wheel.

Here it is a comparison of a web based interface and a mobile app one of the same company, amazon.com (Pictures' dimension are not real).



Garmin Forerunner 305 GPS Receiver With Heart Rate Monitor
by Garmin
(2,272 customer reviews) | Like (435)

List Price: ~~$299.99~~
Price: $269.99 & this item ships for FREE with Super Saver Shipping. Details
You Save: $30.00 (10%)

15 new from $249.99 3 used from $175.00 3 refurbished from $159.98

Edition: Forerunner 305 (English Only)
Forerunner 305 (English & French) Forerunner 305 (English Only)

In Stock.
Sold by World Class Inc. and Fulfilled by Amazon. Gift-wrap available.

Want it delivered Wednesday, May 16? Order it in the next 2 hours and 7 minutes, and choose One-Day Shipping at checkout. Details

Another Forerunner to Consider
Check out the Forerunner 405CX, which has the same features as the Forerunner 305 but has a sleeker design and also includes heart rate-based calorie computation--now just $194.90.
28% off
› See more product promotions

What Other Items Do Customers Buy After Viewing This Item?
Garmin Forerunner 405CX GPS Sport Watch with Heart Rate Monitor (Blue) by Garmin
(418)
$194.90

Click for larger image and other views
notes notes
View and share related images
Watch Compare all Garmin Fitness [Flash]

Quantity: 1
Add to Cart
or
Sign in to turn on 1-Click ordering.
or
Add to Cart with FREE Two-Day Shipping
Amazon Prime Free Trial required. Sign up when you check out. Learn More
Add to Wish List

More Buying Choices
Pavilion Electronics Add to Cart
$274.95 & this item ships for FREE with Super Saver Shipping. Details
Firemall LLC Add to Cart
$279.95 & this item ships for FREE with Super Saver Shipping. Details
Zalmar Add to Cart
$314.49 + $6.97 shipping
21 used & new from $159.98
Have one to sell? Sell on Amazon

Share

Figure 1 – Amazon.com website page

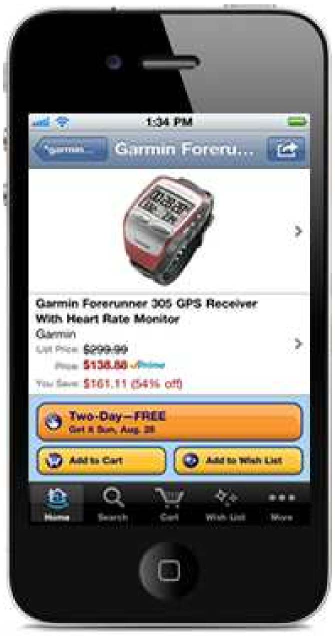


Figure 2 – Amazon.com iPhone App

From these example some differences are extremely highlighted. Just the most important product information for the mobile app, details can be found touching the object, while a lot of information in the web site. Furthermore there are big buttons in the mobile app, considering no mouse is available but only finger press, that are quite small in the website page. The last big difference is that in the mobile app the search bar is in another page, and that is due to the fact that the onscreen keyboard needs space and hides the already small product page.

To summarize here is a list of features to be considered during mobile application development:

- The content of the page should be readable without zooming in or out
- The content on the screen should be minimal, only the most important information.
- Links and buttons should be big and far from each other, the user should not press another link/button. Furthermore it should be clear that the text is a link or a button, there is no mouse over effect

Some navigation help should be implemented; mobile developer should implement the back/forward and home functionality that are commonly available in a browser

The smartphone has sensors that can be useful for achieve information (GPS position) or to ease user interaction (natural language voice control)

And what about tablet? The touch and multi touch user interaction is still valid to tablet; the main difference is the display size. With a 9.7’ display the page is absolutely bigger than the smartphone one. Content is no more an issue for tablet, and in fact books and magazine are successfully tablet apps. Tablets are the best replacement for paper, all the things that now we do using a pen and a sheet of paper could be done with a tablet, and much more.

4 Smartphone Development

More than forty per cent of cell-phone users have apps on their devices, and most use them regularly, according to some survey.
But building and deploying a quality app can be difficult. Most companies don't have deep mobile app dev skills in-house. There are multiple operating systems to develop for, and hundreds of OS-device combinations. As companies consider their mobile app development strategy, one critical decision is what type of app you'll deploy. A second critical decision is how you'll deploy those apps—link your development strategy with mobile device management.
There are three main ways to develop mobile apps (Information Week Source):

1. Develop a native application for each platform: Android, iOS, BlackBerry, etc.
2. Buy and use a cross-platform development framework, that leverages its APIs to write code once but have your app run on multiple platforms.
3. Use a mobile enterprise application platform, which provides prebuilt, enterprise-ready apps that integrate with your existing business systems via a vendor's framework, enabling rapid deployment of apps without much development.

Let's start by looking at each development approach, as each has notable pros and cons.

4.1 Native Apps

The main reason to develop a native smartphone application is because the app needs access to specific functionality provided by the device, such as an accelerometer, camera, or GPS, and needs the benefits of integration with other native apps and local processing. How many times have you downloaded an app on your phone and the drop-down boxes or check boxes aren't working like they do with the built-in applications on the device?

While it's easy to understand the pros of a native app, there can be many more cons. For example you have to develop a specific application for each different Operating System.

We have to test the native app on many different mobile devices. Different screen sizes, processors, and RAM can change the way an application functions.

Unless you've standardized on one device for your app, the large number of mobile platforms and devices makes the likelihood of finding an incompatibility high.

4.2 Cross-Platform Development Framework

"One codebase, every smartphone" is the mantra for Cross Platform development frameworks from vendors such as Rhomobile, Appcelerator, and WidgetPad. These platforms let developers create new mobile apps rapidly, usually using prebuilt user interfaces and drag-and-drop programming capabilities. While they do provide the capability to build powerful apps, most of these platforms are for apps that redisplay content from a central repository, such as a website, or for apps with a single function.

One advantage is that, while Android and iOS use Java and Objective-C, respectively, as their main programming languages, crossplatform development frameworks may use more common languages, such as JavaScript, HTML5, and CSS. That makes the job of finding developers easier and the time to develop an app shorter.

The main benefit of cross-platform development frameworks is module reuse. The famous Mythical Man-Month software development book states that the key to successful software development is the reuse of well-known and highly tested modules, since that reuse cuts the time to finish a project and increases reliability. Cross-platform development frameworks have forums and communities where developers can download prebuilt components such as drop-down boxes, grids of data, and integrations to social networks.

Stay on guard, however. Even though these modules are highly tested, that doesn’t guarantee they’re secure, so you’ll still need to analyze the quality of code being developed and the code of the cross-platform development framework itself.

If you’re mainly using an app to provide content and don’t expect users to do a lot of back-and-forth interaction in the app, Cross Platform development frameworks are the way to go.

4.3 Mobile Enterprise Application Platform

What if you already have a Web application at the core of your business, such as a customer service portal? In the early 2000s, Java-based Web apps were seen as the future. If you want to simplify and provide access to such a Web app but optimize it for the mobile experience, a mobile enterprise application platform might be the right route.

Unlike native applications or apps written with a cross-platform development framework, a mobile enterprise application platform is something like mobile middleware — a set of applications (and usually a limited set of the features found in Cross Platform development framework, such as APIs) that a business can use to integrate its software. For example, a MEAP might let a company provide a slimmed-down, mobile version of an enterprise CRM app with a subset of functions a field rep would need, along with integration with the full-blown on-premises version of the software. These applications are tested and backed by the MEAP vendor and used by thousands of customers, thereby increasing their reliability and functionality.

Using applications from a MEAP vendor that have been tested to integrate quickly with a monolithic system can let an IT organization move quickly.

Furthermore, because not much development is needed integrate a MEAP application with your legacy software, you probably won't need developers on staff or even developers at all, as MEAP providers offer integration as a professional service.

Looking down the road, one other force is likely to have a major impact on mobile app development: HTML5.

HTML5 has been heralded as the death of Adobe Flash, because it can provide drawing, video, and sound for Web apps. It even has the ability to per form geolocation and threading to increase performance. Companies are embracing HTML5 because it makes it easy to create applications and because of its ability to store data in a client-side cache (on the device), so that applications written in HTML5 can work offline. Some of the cross-platform development frameworks already embrace HTML5, and most mobile devices support it.

HTML5 will let companies rapidly develop mobile applications based on an existing Web app, and still provide rich content and user interfaces normally found only in Flash. If your application is Web-based and requires a lot of data input, updates, and other interaction, HTML5 is one of the quickest and easiest ways to convert that app for mobile devices. If your development team isn't up to speed on HTML5, it's time to get them training, as HTML5 is being embraced by Google, Apple, and other industry juggernauts.

4.4 Next Step: Development Essentials

Whichever route you take, there are a few pitfalls.

First, the whole point of these mobile app projects is to help the business in some way, so it's amazing how uninvolved business unit leaders and users often are in the development process. They need to be involved throughout the process to ensure that the proper features are taking shape.

Second, assess how you will support the app. Whether development is outsourced or done in-house, make sure the help desk is ready to handle user calls. Users will need training as well.

Third, once a mobile app is built, it still needs to be deployed. You can use a public application repository such as the iTunes App Store or Android Market. But more likely you'll want an in-house app store, which means you'll need mobile device management software to publish new apps and updates. Your IT department can even create bundles of apps that include public apps that IT allows on company-issued mobile devices—a great feature if your enterprise app relies on third-party apps. IT teams also can create role-based app groups—apps available to salespeople, for example, but not executives.

Vendors such as AirWatch, MobileIron, Good Technology, and Sybase provide mobile device management software. Their software includes management features that are a must-have if you're rolling out apps to more than a handful of devices. Each mobile device is essentially a computer that needs to be updated, maintained, and supported. Most MDM software also lets your help desk staff access a tablet or mobile phone remotely and see what end users are seeing on their screens.

While we have only briefly touched on information security, it's one of the top concerns when deploying mobile applications, and MDM plays a big role there as well. As companies think about which mobile apps they want to build, they need to mitigate the risk of data exposure from devices that are lost, stolen, or even exchanged at a wireless store.

MDM software lets administrators remotely wipe a device clean of data.

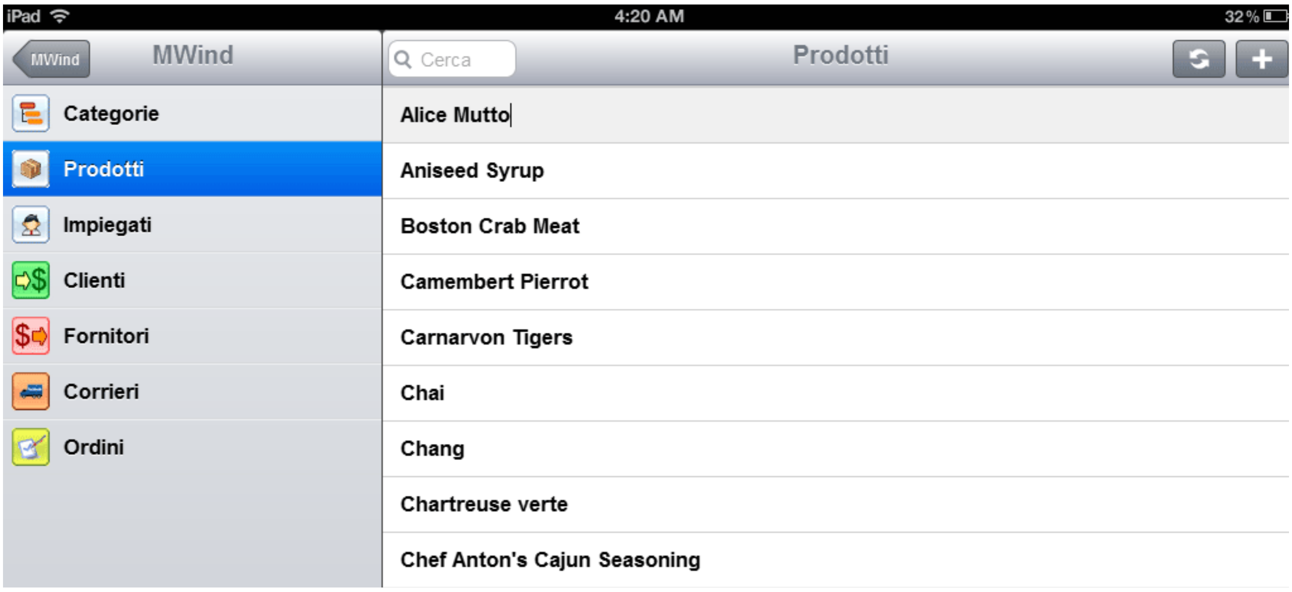
Mobility is now a fact of business life, but many IT teams aren't build for it. Outsourcing mobile app development isn't enough. You need a clear mobility strategy, from picking the right development approach to deploying and managing the devices to effectively securing the data.

5 Tablet Development

The tablet development, in spite of fact that uses the same Operating system than smartphone, is different. We have different Video Form Factor, different size.

Normally mobile applications developed for Tablet use for example the split view model when are displayed. In such a case, the space on the screen is divided into a left part where is present the main menu or a visualization for the objects' selection on which to operate, and a greater space on the right where will appear the opened screens.

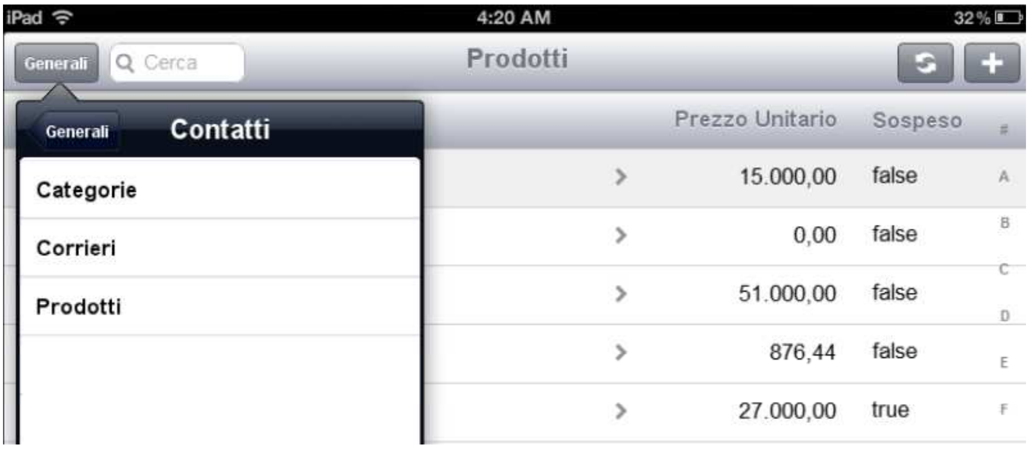
The following image shows an example of this structure. It should be take in mind that the display on the left side is not mandatory and you can also change the screen menu with a left docked. On the right side, however, if there are no open screens will display the file qhelp.htm.



Alice Mutto
Aniseed Syrup
Boston Crab Meat
Camembert Pierrot
Carnarvon Tigers
Chai
Chang
Chartreuse verte
Chef Anton's Cajun Seasoning

Figure 3 – Screen divided in the case of tablet used in horizontal

If the tablet is vertically rotated, the left side is automatically hidden, while the right side extends up to occupy all available space. In this case, in the toolbar that appears in the foreground of the screen you can touch a button that opens the left side in popover mode, as shown below:



Prezzo Unitario	Sospeso
15.000,00	false
0,00	false
51.000,00	false
876,44	false
27.000,00	true

Figure 4 – Example of the main menu open as popover

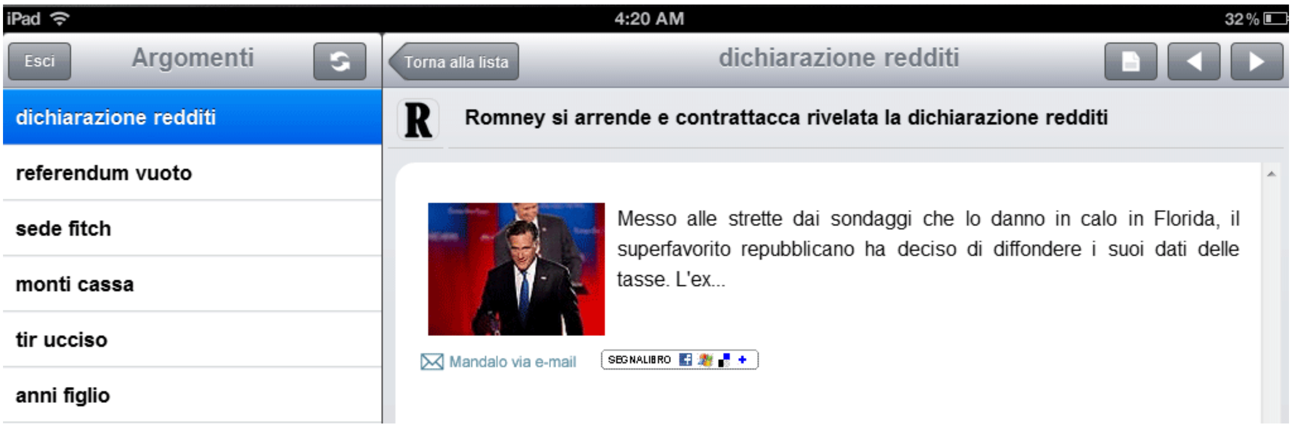
This button is available only if the current view is not dependent on other views above because in such cases on the left side of the toolbar there is a button to come back.

Now let's see what types of screens, in addition to the normal ones, can be used for a tablet.

5.1 Left Docked Screenful

A left docked screen appears on the left of the video when the tablet is used in horizontal, or as a popover if the tablet is vertical. In practice, a left docked screen occupies the same space of application's main menu.

These screens are very useful to show the objects' list on which to work. The image below, for example, uses a screen left docked to show the list of news topics of the day.



dichiarazione redditi
referendum vuoto
sede fitch
monti cassa
tir ucciso
anni figlio

Romney si arrende e contrattacca rivelata la dichiarazione redditi

Messo alle strette dai sondaggi che lo danno in calo in Florida, il superfavorito repubblicano ha deciso di diffondere i suoi dati delle tasse. L'ex...

A docked screenful may contain a list or panel in detail, a tree structure, a report, or even a combination of the above inside a tabbed view. Moreover, even if the docked screen appears instead of the main menu, it does not exclude a priori. When the operator closes the left docked screen, in fact, the main menu reappears.

5.2 Modal Screenful

A modal screenful has an user interaction closed in itself. Until it is closed, in fact, it is not possible to use other parts of the application. The following picture shows a screen to collect the signature of an order. Usually the modal screens are used when you need to perform the steps of a process in a fixed order. You should not use them extensively because they are not very common and require the user to follow a fixed route instead of leaving it free to use the application as he/she wishes. Although it is technically possible, it is advisable to open a second level of modal screen from a first level, because this is likely to confuse the user and maybe she/he is not able to fully understand the current process.

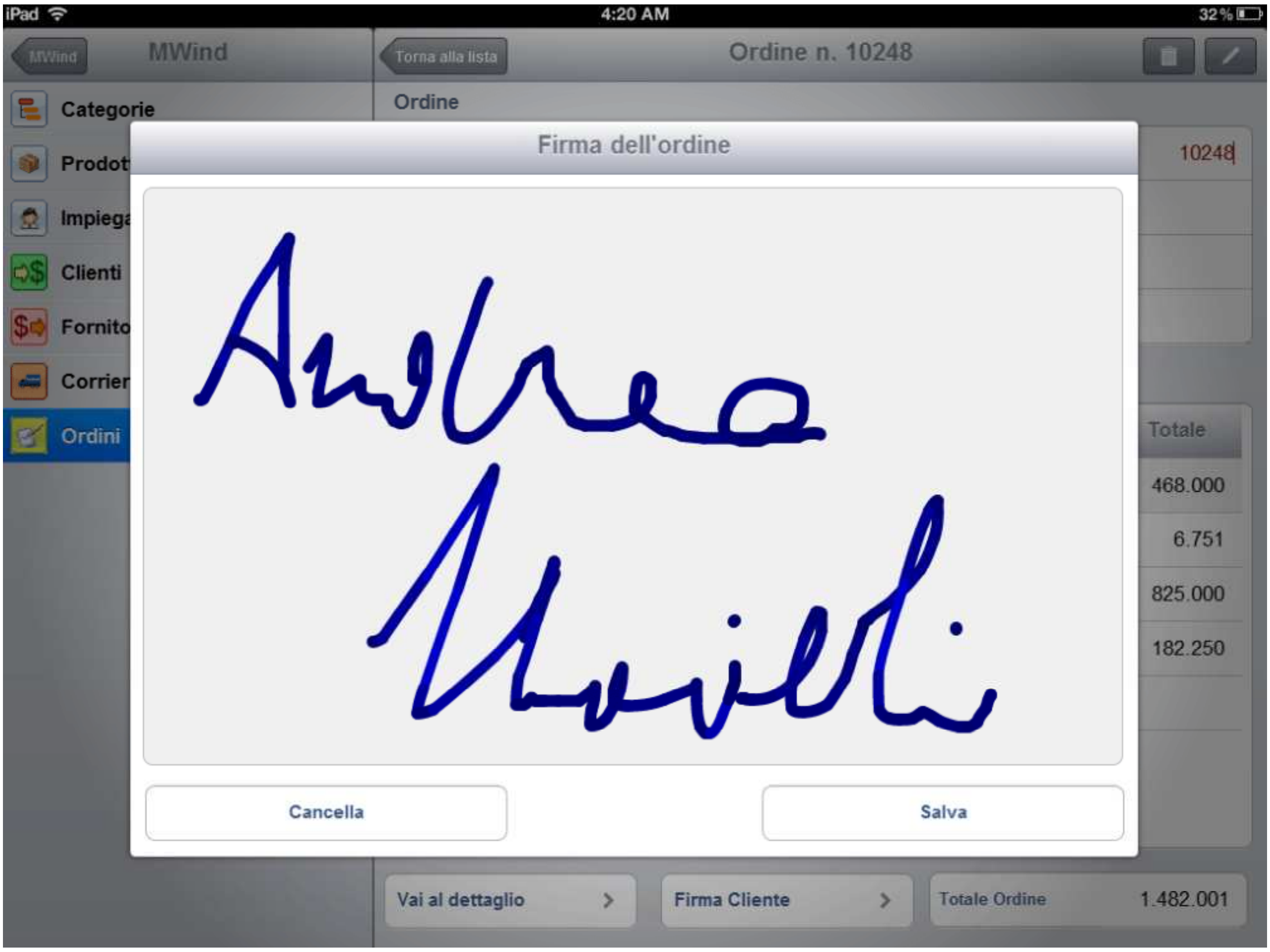
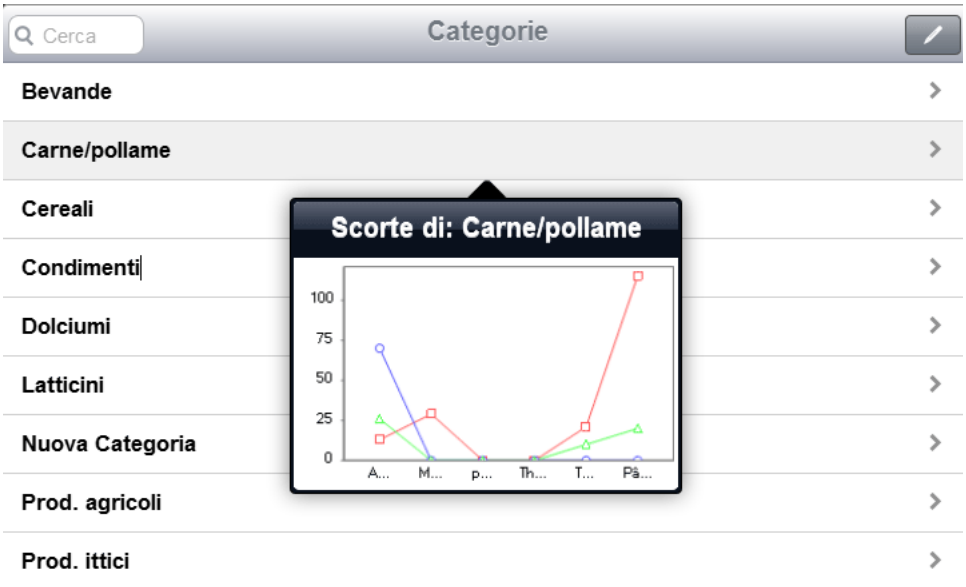


Figure 5 – Esempio di videata modale per la raccolta della firma

5.3 Popover Screenful

The popover screenful corresponds to popup desktop applications. They are used only in tablet and do not have a match in smartphones. They can be used to provide information in relation to an object of interest, and usually they are automatically closed when the user touches the outside of them. Use of popover screenshots is recommended only in special cases because normally the application is more complex to use. It may be interesting as "zoom" of summary data presented in the bottom of the screen.



Bevande

Carne/pollame

Cereali

Condimenti

Dolciumi

Latticini

Nuova Categoria

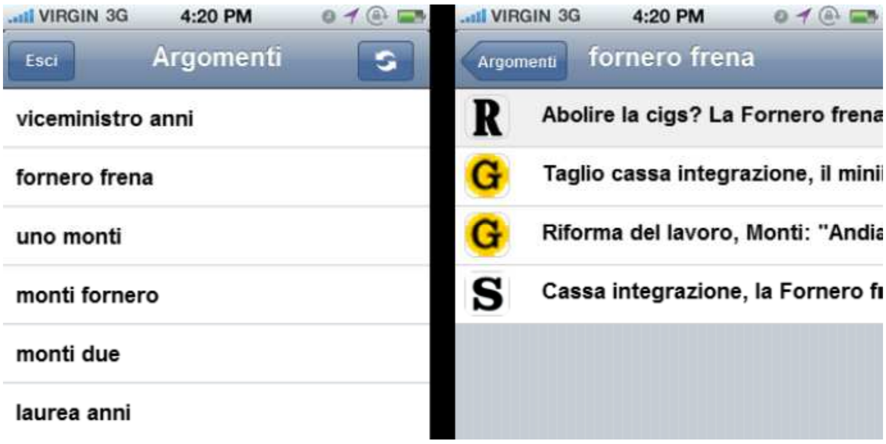
Prod. agricoli

Prod. ittici

Figure 6 – Popover Screenful Example

5.4 Linked Screenful

When an user action on a control due to open another screen, these two forms are connected. In such a case, in the title bar of the second one there is a button that allows you to return to the first, because when the user touches the second one this will be closed. The following illustration shows the sample application Newscloud view (in an iPad). The news' list appears when you touch a topic, then this is a screenful connected to the first. By tapping the button in the toolbar, the argument list closes and returns you to the previous year.



viceministro anni

fornero frena

uno monti

monti fornero

monti due

laurea anni

Abolire la cigs? La Fornero frena

Taglio cassa integrazione, il mini

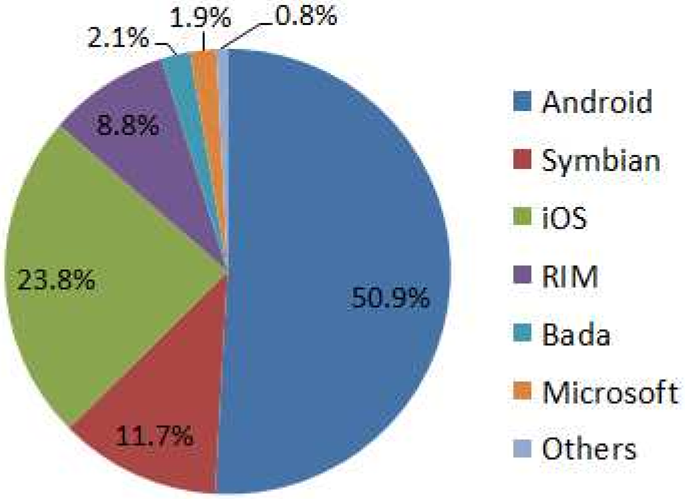
Riforma del lavoro, Monti: "Andia

Cassa integrazione, la Fornero fr

Figure 7 – Linked Screenful Example

6 More on Native Application versus Web Application

Currently, as shown in the graph in fig. 1, the most popular mobile software platforms are the following: iPhone OS, Android, Symbian, BlackBerry.



If you want to face development of a native mobile application to make it accessible to most users, you should realize at least 4 distinct applications, one for each of the platforms listed above. For example if you develop an application for iPhone or iPad you should buy an iMac, register as a developer Apple in the AppStore, download the development kit for iPhone and download the XCode tools. Developing for Android is certainly more simple: it is sufficient to download a plugin for Eclipse that includes the SDK. In any case it would become very expensive to develop this application for more than two platforms, as it would require a team decidedly important.

Fortunately, if the functionality does not require advanced graphics, you can subtract from that task by developing a network application. To date browsers for mobile devices have reached a maturity strong enough to support high-level mobile Web applications.

Until just some months ago, the reference seemed to be moving toward Framework jQTouch (http://jqtouch.com/), very easy to use tool but still very immature. Unfortunately jQTouch not evolved that much standing still at version 1 beta 4rc. Another limit is that it works only with jQTouch browser developed by the engine "WebKit". Meanwhile several others OpenSource Framework were born. Among these the one that seems to prevail over others, both in terms of ease of use and reliability in providing future updates, seems to be "jQuery Mobile" (http://jquerymobile.com/), even just for the experience provided jQuery framework to support the desktop. jQuery Mobile is based on the jQuery framework and provides a graphical user interface based on HTML5, unified for all the most popular mobile device platforms.

At this link: http://www.jqmgallery.com you can see a gallery of applications developed with jQuery Mobile.

As for the other frameworks, although difficult to fully assess without having them used in real projects you can without doubt say that they are deficient as regards the following aspects: limited functionality, difficulty of use, support only the "WebKit", uncertainty in the evolutionary development. Among these we will cite the following:

Zepto: http://zeptojs.com

Sencha: http://www.sencha.com

Wink Toolkit: http://www.winktoolkit.org

iUI: http://code.google.com/p/iui/wiki/Introduction

iWebKit: http://snippetspace.com/2011/03/03/iwebkit-6-preview

WebApp.Net http://webapp-net.com

7 Application features

The application to be developed will be made available to users working in a port.

In particular, the software will serve to streamline mechanisms such as acceptance of the drivers at the port, providing a tablet to the guard responsible for the input so that they can pass the device to driver that must fill out the form (with name and surname) to obtain the temporary permission input, in the event that he is not in possession of any authorization tool (badges, fingerprint recorded, ...).

The same procedure is also valid in case of operators who are not drivers of trucks, but people traveling by car, other means of transport or on foot.

Another feature lets drivers (or forwarder) to make, even before arriving at the port, the preparation of notice of arrival. This is the case in which following a report made by the terminal operator of goods ready for pickup, followed by an acceptance of the assignment by the driver.

In another case the driver (or forwarder) could modify an existing preannouncement having received reports that have changed one or more associated authorization information.

As a last feature we describe what is available to the user terminal: he will have the opportunity to enter an authorization "on the fly", i.e., by accessing the list of preannouncement inserted, he may authorize any of these. It's a feature provided for emergency cases, when the terminal operator behind the arrival of a truck realizes that he has not authorized the mission through.

8 Application Architecture

Although this is a prototype application it was considered appropriate, in the development phase, to use a Database. The Database has allowed us to make a more realistic application, especially in the management of the states of notices and requests for access to the port. The Database used is the Microsoft SQL Server 2008. See all application architecture represented in the diagram below.

The architecture of the mobile application consists of a component HTML / Javascript and a web service that encapsulates access to the data. The pages in pure HTML are not tied to any server-side technology. They are invoked only once, and all subsequent interactions with the database are made using Ajax calls to the Web Service above. Whereas it is a prototype, to simplify the architecture, the Business Logic layer has been incorporated within the same Web Service. This Web Service, to perform CRUD operations on the database uses a library developed by Optisoft and called "OptiDAL".

Please note that we developed two libraries "OptiDAL" in C# for a Windows / IIS environment and the other for the Java / JBoss environment.

In development of mobile application in question, having no special directives and just for internal reasons of ease of use, it was decided to use the C # version. As a result of this choice, the development of the web service is supposed to do in C # for IIS. If you wish to use the application on Java / JBoss, you need only to make a simple porting of the web service in JBoss environment and at this point just rely on the library "OptiDAL" developed for JBoss.

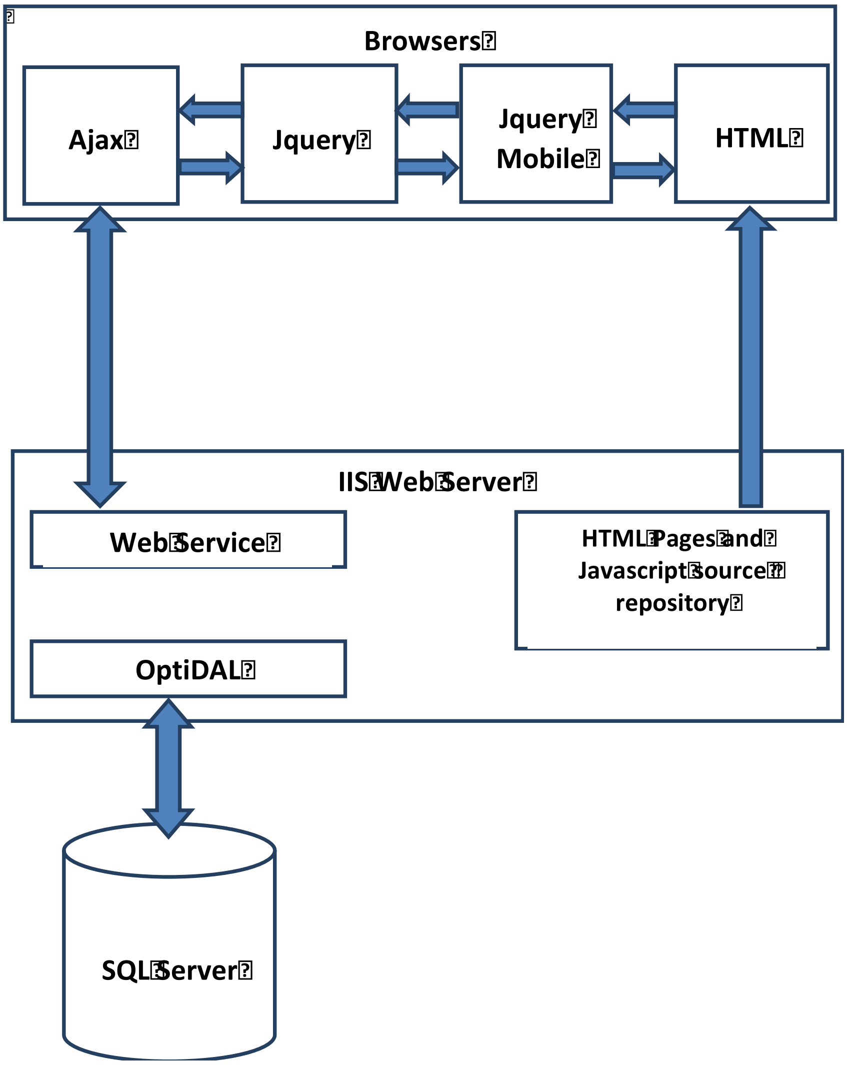


Figure 8 – the home page

9 Development Tool

The choice to realize a network application (and not a native application) has been dictated by the need to cover the largest number of mobile devices, as already stated. Should not be underestimated even the possibility to use the application in a PC Desktop environment, using the most popular browsers (except Internet Explorer 8). The fact that it was a managerial application, essentially based on forms, corroborated the choice, as the development of a native application for this type of applications, it would provide no added value.

To facilitate application development it was decided to use a javascript framework. In particular, the choice fell on Jquery Mobile. In our opinion (and not only ours), the best tool on the market as stability, reliability, and support for developers.

In the application's development we don't encountered particular difficulties. The only hitch it happened in the use of Safari on the iPhone. For example when the operator inserted a new preannouncement, the element just added did not appeared in the filtered list items. This error happened when we used the version of Safari for iOS6 that used to perform the caching of POST via Ajax. To overcome this problem it was necessary to change the signature of web services by adding a parameter whose value needs to be calculated as a function of time.

An inelegant but effective solution.

As an alternative to Safari you can possibly use Chrome.

A compatibility problem occurred using a widget to represent dates, developed for jQuery Mobile and not compatible with the native Blackberry browser. To obviate this drawback it should represent dates with three fields of edit separated. There were also problems of alignment between the labels and the corresponding fields of edit. In smaller device (smartphone in particular) the tendency of the browser is to bring the head to the edit field without taking into account the percentage of jobs defined by CSS.

Glossary

Term	Description
Customs Agency	Stakeholder:
Customs Authority	Stakeholder: responsible for collecting and safeguarding customs duties and for controlling the flow of goods including animals, transports, personal effects and hazardous items in and out of a country. Depending on local legislation and regulations, the import or export of some goods may be restricted or forbidden, and the customs agency enforces these rules.
Customs Police	Stakeholder: is essentially responsible for dealing with financial crime and smuggling; it has also evolved into Italy's primary agency for suppressing the drugs trade.
Bayplan	Document: A schematic diagram of cargo bays within a shipping vessel. The bays represent vertical divisions of the entire stowage area and are numbered from bow to stern.
Customs Declaration	Document: it is a paper document with information related to importer, exporter, goods description and codes (TARIC),…, container id.
Ship Freight Forwarder	Stakeholder: is a person or company that organizes shipments for individuals or corporations to get large orders from the manufacturer or producer to market or final point of distribution. Forwarders will contract with a carrier to facilitate the movement of goods. A forwarder is not typically a carrier, but is an expert in supply chain management. In other words, a freight forwarder is a "travel agent," for the cargo industry, or a third-party (non-asset-based) logistics provider.
Bill of Lading	Document: A legal document between the shipper of a particular good and the carrier detailing the type, quantity and destination of the good being carried. The bill of lading also serves as a receipt of shipment when the good is delivered to the predetermined destination. This document must accompany the shipped goods, no matter the form of transportation, and must be signed by an authorized representative from the carrier, shipper and receiver.
Loading/Unloading Order	Document: List of containers to load/unload. Data declared: ship name and voyage; containers list (id, seal, type, port of provenance, weight, …)
Forwarder	Stakeholder:
Haulier	Stakeholder:
Harbour Master's Office	Stakeholder: in accordance with the Navigation Code and other special laws, carries out a policing and safety role as well as those administrative duties which don't come under the port authorities

	ambit. The Harbour Master’s office activities can thus be summarised as follows: • coast guard, search and sea rescue • safety of navigation (licensing and controlling vessel safety certificates, controlling on safe sailing conditions of the vessels) • maritime police (the regulating of traffic in and out of the port’s waters, berthing and maritime traffic; organisation of ancillary services and relevant tariffs; inquiries and investigations on navigation accidents) • naval property and administrative regulation of vessels • marine environment safety and protection.
Bindello	Document: it is the weight ticket used to notify and certify the quantity of goods actually loaded on the trucks
Delivery Order	A document that contains information about the Ship name and voyage; container list (id, seal, type, weight, forwarder, …)
Customs Clearance	Declaration references, container list, MRN, …
Customs Declaration	Importer, exporter, goods description and codes (TARIC), … , container id
Customs Declaration (Exit Certification)	Document: it is similar to Customs Declaration, but with Customs Police’s stamp to certify goods exit from the port area
Discharging Order	Ship name and voyage; container list (id, port of discharge, weight, type, seal, booking number, …)
CRM	Document: Goods description, truck destination, containers id, …
CRM (Entrance Certification)	Document: Similar to CRM, but with Customs Police’s stamp to certify goods entrance in port area. The document enables the presentation of the customs export declaration.
Port Gate Entrance Notice	Document: Container id, date/time of entrance, booking number, hauler, …
A3	Document: temporary custody produced by the Ship Manifest
Kemler code	It is a code giving advice on the nature of the hazard presented by the goods. It is printed on the Hazard Warning Panel
UNDG number	It classifies the dangerous goods. It is printed on the Hazard Warning Panel

Acronyms

Acronym	Description
VPA	Venice Port Authority
PCS	Port Community System
APC	Adriatic Port Community
ITS	IT System
CRM	Convention relative au contrat de transport international de marchandises par route. It is equivalent to the Italian DDT (Documento Di Trasporto)
MRN	Movement Reference Number
MMA	Manifesto Merci in Arrivo equivalent to ACM
ACM	Arrival Customs Manifest
MMP	Manifesto Merci in Partenza equivalent to DCM
DCM	Departure Customs Manifest
TOS	Terminal Operating System: Computer system used by the terminal user to manage the activities and the operations at the Terminal
CPRE	Customs police Risk Evaluation: it is a system available from the Customs Police to the system for verifying the risk level for the driver, the truck, goods.
ACS	Access Control System: it is the SaFE component. A security application that identifies people and vehicle (by the plate number)
AGS	Automatic Gate System: the main system, the subject of this document

Referenced documents

Code	Title

ACVP_1103	Assessment delle procedure per l'accesso al porto 25ommercial di Venezia (marzo 2011) Assessment of the Procedures for accessing the Commercial Port of Venice (2011 March)
ITS_NAPA	Napa Port Organization and Processes Assessment
APC_VE_SUM	APC Venice Port Summary (2012 March)
APC_ITS_NAPA	APC vs ITS NAPA – a clarification/description of the relationships between the projects
APC_USR_REQ	User Requirements (at Venice Port level)
APC_SYS_SPEC	System Specifications at Venice Port level
APC_MOB_REQ	Specifiche per le applicazioni "mobile" Requirement specifications for mobile applications