

An iPad Order Management System for Fashion Trade

Ivano Baroni, Sonia Bergamaschi and Laura Po

Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia,
Via Vignolese 905, 41125 Modena, Italy

Keywords: Mobile App, Order Management System, Fashion Trade.

Abstract: The fashion industry loves the new tablets. In 2011 we noted a 38% growth of e-commerce in the Italian fashion industry. A large number of brands have understood the value of mobile devices as the key channel for consumer communication. The interest of brands in applications of mobile marketing and services have made a big step forward, with an increase of 129% in 2011 (osservatori.net, 2012). This paper presents a mobile version of the Fashion OMS (Order Management System) web application. Fashion Touch is a mobile application that allows clients and company's sales networks to process commercial orders, consult the product catalog and manage customers as the OMS web version does with the added functionality of the off-line order entering mode. To develop an effective mobile App, we started by analyzing the new web technologies for mobile applications (HTML5, CSS3, Ajax) and their relative development frameworks making a comparison with the Apple's native programming language. We selected Titanium, a multi-platform framework for native mobile and desktop devices application development via web technologies as the best framework for our purpose. We faced issues concerning the network synchronization and studied different database solutions depending on the device hardware characteristics and performances. This paper reports every aspect of the App development until the publication on the Apple Store.

1 INTRODUCTION

The world of mobile devices and applications is under an increasingly faster development. IDC prediction for 2013 says that mobile devices are the emerging markets that will drive growth¹. The surge in mobility will lead to mobile devices surpassing PCs as the method of choice for online access. The number of people accessing the Internet through PCs will shrink by 15 million over the next four years, while the number of mobile users will increase by 91 million.

Every day more and more people approach mobile technologies for simplifying and optimizing every aspect of their lives. Emotional factors caused by the adoption of mobile devices play an important role (Platzer, 2009). Users have showed a high attachment towards mobile services and possibilities these devices offer (Meschtscherjakov, 2009). At the same time, developers all over the world try to find new and different solutions to speed up the developing process of the applications.

This paper will explain the development of *Fashion Touch* app, a mobile version of a web Order Man-

agement System (OMS) developed by the software-house Quix for Fashion Trade companies. An OMS is a software product for sales force automation (SFA) used in a number of industries for order entry and processing. Orders can be received from businesses, consumers, or a mix of both, depending on the products. Offers and pricing may be done via catalogs, websites, or broadcast network advertisements.

Traditionally, an agent visits the customers with a paper catalog, the physical products of the collection, and writes down orders by hand. Agents that adopt an OMS can place online orders using the web application. Of course, in order to complete an online order you must have a computer, and a stable internet connection. Actually, the connection is not always available and reliable because some customers might be located in remote places.

Fashion Touch is an App that has been conceived for being used by the fashion agents that sell products of one or more brands in a more or less wide geographical area. By using *Fashion Touch*, the agent relies on an up-to-date catalog with high-resolution images (the quality is much better than the quality of images on a paper catalog), and an OMS aligned

¹ www.idc.com

with the ERP that can record orders even if the internet connection is not always available.

Broadly speaking, Fashion touch allows offline browsing of the full catalog of multimedia content (images items), it assists users in the order creation process through a step-by step procedure of customer, line and season selection and it displays customer information with the ability to send e-mail and geo-location of the same. Moreover, a wide set of functionalities for drafting and editing seasonal orders (custom keyboard to simplify and speed up the process of order entry; functionalities of viewing and editing the individual row order in details before processing it) are available. In the end, it is a multilingual app.

The application has been designed and desired by Quix s.r.l. that after developing the OMS web application, has felt the need to put on the market a Mobile version of the application. *Fashion Touch* is then the result of a collaboration between Quix and the DBGROUP of the University of Modena and Reggio Emilia. In particular, the analysis and development of *Fashion Touch* has been started during the internship of Ivano Baroni on 2012.

Quix S.r.l.² is a software house located near Modena founded in November 2000 as a web company. Quix S.r.l. has closely followed the ICT evolution adapting its offers to the marketing requirements and providing a set of products increasingly wide and complete that cover the specific needs of public and private sectors. The main application areas are enterprise portal, document management, information systems, business intelligence. The company has a close connection with the DBGROUP of the University of Modena and Reggio Emilia whereby often collaborates to study new technologies and applications.

There are some other solutions of mobile OMS on the market. In (Patrick Olson, 2010), authors describe a Mobile Order Management System (MOMS) that is designed based on a client-server model, with central database serving as a repository of data relating to customers and product information. The app has been developed for smartphones based on the open source Android platform. The application has been thought to be able to work even in an off-line mode. IBM, instead, proposed the Sterling Field Sales a caMobile³ application that provides timely access to product content, pricing and inventory availability information from mobile devices, such as Apple iPhone and iPod Touch. The problem of developing app for smartphones is that they are not the best devices if you want to offer customers a chance to look at an high

²www.quix.it

³www.sterlingcommerce.com

resolution image of the product on a sufficiently wide screen. *Fashion Touch* was developed for a tablet, iPad, as the capabilities of easily inserting orders by selecting items from a catalog of thousand of products is unfeasible on a smartphone.

This paper depicts the different phases that have led to the development and commercialization of *Fashion Touch* until the publication on the Apple Store.

The paper is organized as follows. Firstly, on Section 2, the OMS web application is depicted. Section 3 analyzes web technologies for mobile applications and their relative development framework making a comparison with the Apple's native programming language. Section 4 describes in details the mobile App developed, facing issues concerning the network synchronization and studying the different database solutions depending on the device hardware characteristics and performances. In the end, some conclusions and future work are depicted on Section 5.

2 THE OMS WEB APPLICATION

Quix s.r.l. implemented an OMS web application to meet the company sales force needs of having a centralized and automated collecting orders system connected to the ERP in order to enable effective stocks and production lines management. The OMS is a *Java Web Application* built on the *Apache Struts 2* framework and based on the *Central Authentication Service* protocol to ensure users single sign-on authentication. The application has been graphically customized for different clients. Since Quix is situated in a geographical area crowded of fashion industries, the OMS was designed to satisfy the needs of these commercial realities. The recipient of the application are the fashion companies, while the target users consist of the fashion agents who periodically visit the stores to collect sales orders.

The adoption of an OMS leads to unquestionable advantages, such as the chance to inspect the sales data almost in real time. With the traditional systems, agents delivered to the company the copies of the commissioned orders after few weeks, these data needed to be entered into the company's information system. Thus, the sales data could be analyzed at least 1 month after the effective sale operation. One month is a very long period in the fashion industry because of the nature of the product treated. By using an OMS, the company has the opportunity to conduct sales analysis in real time.

The web app is a solution created to manage the

whole process of customers' acquisition orders in fashion industries. The requirements imposed by the business are clear: to manage multiple and different clients in the same period of time and with the least amount of physical items; in practice, to reduce costs and raise revenues. The objectives set forth in the Web Application development were met with success. The Web Application manages the acquisition process of customer orders in the fashion world with speed and full integration with other processes in the company, it helps businesses to optimize operations and to reduce operating costs with a multiple choice of brand, franchise, outlets and distributors.

Important benefits have been reached by using the Fashion OMS, including the ability to handle more customers in showrooms in a period, placing orders of different types (seasonal, flash, re-assortment returnable, etc.) and in different modalities. The OMS also implements features like a practical reading barcode, the virtual showroom for a visual selection of the heads, and the generation of PDF documents (order report, customer report, order book, customer book, etc.).

The Fashion OMS is one of the Digital Enterprise solutions in line with the concept of *cloud* for perfect usability on all devices. It is a multi-platform Web App, multidevices, multi-lingual and multi-database, it can be used both as an *in house* solution (by acquiring a license) or as a SaaS (Software as a Service, with an annual subscription). The Fashion OMS can be integrated with any ERPs. The integration process usually takes place through export of data in csv file (but there are other alternatives that can be implemented).

3 WEB TECHNOLOGIES VERSUS MOBILE APPS

Writing applications for mobile devices is a challenging task as they can have multiple operating systems, and a wide range of hardware specifications. One solution is to provide Web applications using common web technology, as they can run cross-platform on mobile devices: in this case, there is no need to use proprietary technology (such as Objective-C with Cocoa on the iPhone). Another solution is the use of general frameworks with the capability of converting code in native software for a specific device.

PhoneGap and Appcelerator Titanium are both very popular open-source JavaScript frameworks for packaging and deploying mobile applications. Both frameworks have successful implementations on numerous platforms. However, for a right choice on the best framework for our purpose, we decided to com-

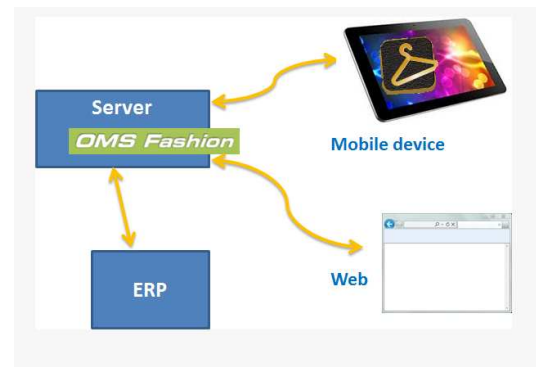


Figure 1: The communication among OMS web server, ERP and Fashion Touch mobile App.

pare the major significant differences between the two products. Our comparison started from the considerations reported in (Ohrt and Turau, 2012) about cross-platform mobile development tools. This section is intended to describe the differences between the two development frameworks, by highlighting the main features together with the positive and negative aspects that have been taken into account when we identified the best tool for the development of our App.

One of the fundamental difference between the two technologies is that PhoneGap is a web-based solution while Appcelerator Titanium is a pure JavaScript API that creates native code.

Appcelerator Titanium⁴ allows developers to utilize JavaScript to write a mobile application and compile down to native code for deployment. PhoneGap⁵ allows developers to utilize HTML5 standards to write mobile applications while also providing a JavaScript SDK to access native device capabilities. Developers use HTML code to define the GUI. PhoneGap ensures a wide portability on tablets and mobile platforms; it provides native source code or libraries for more device platforms than Titanium and it also includes a template for creating native smartphone apps that is extended with HTML and JavaScript files. However, the quality of GUI in PhoneGap Apps will vary based on the quality of the web view and rendering engine platform.

Appcelerator Titanium is a stand-alone SDK also built on the Eclipse platform. Developers write all source code in JavaScript by using APIs offered by Titanium. Titanium supports access to a wide set of native features and functionalities, from the GUI components to socket interfaces to system functions. During compilation, Titanium combines source code with a JavaScript interpreter into an app package, then

⁴<http://www.appcelerator.com>

⁵<http://www.phonegap.com>

at runtime, the interpreter processes JavaScript code. Since Titanium can be extended with visual components, developers are able to implement any GUI that is possible on the underlying native platform. The functionality gap between Titanium and a pure native apps is reduced approaching zero. However, the main limitation of Titanium remains that it is only available on the most popular mobile platforms: iOS, Android and Windows Mobile.

In conclusion, we opted for Appcelerator Titanium. First of all the target device was the iPad, so it was not necessary to select a framework with a strong inter-platform functionality, however, we decided not to develop in native to maintain the possibility of expansion on different platforms. The creation of the graphical interfaces in Titanium has a cost in terms of time significantly less than the time cost obtained using web-based tools. Titanium guaranteed the best compromise between quality and speed of software development and interoperability of different devices. Another crucial aspect we took into account for this choice is that Titanium can be extended with free or paid modules available on its Marketplace. So far, in the Fashion Touch App we have used two of these modules taken from the Marketplace (see section 4.1), and another one is intended to be used in the next future.

4 THE FASHION TOUCH APP

Fashion Touch is an iOS (iPAD) application for defining off-line orders for fashion industries. The application addresses the sales force of companies that wish to have an innovative approach and, at the same time, an optimized process with a simple and intuitive tool on which the information is always available.

Fashion Touch is one of the first Italian mobile app for managing orders in the Fashion Trade⁶. It improves productivity, time and sales processes; it supports with additional information the sales network and creates an emotional connection with the brand and provides readily access to business information.

Following the graphical rules set up in (Ballard, 2007), the visual design has been thought to ensure usability and emotional response such as interest, excitement and pleasure. The application integrates seamlessly with the most popular information management systems and has a component devoted to the automatic communications of orders to the internal system. The sales network is automatically synchro-

⁶a video demonstration of the App can be found at <http://www.dbgroup.unimore.it/fashiontouch>



Figure 2: The order generation workflow in the Fashion Touch App.

nized with the ERP internal application and it relies on fresh and reliable content (see figure 1).

4.1 The Main Features of the App

The process of order generation (see Figure 2) starts from its creation, i.e. this happens at the time that an agent is visiting a customer and the customer decides to make an order of garments. The agent selects season, client, and brand. After this, on the mobile device it will be available the complete off-line catalog. Then, the agent supports customers in navigating the catalog up to the selection of items and the definition of the quantities for the different sizes. At the end of this stage, and also during selection, the order can be displayed in a summarized version for checking errors and applying revisions, if necessary. After this, the order can be confirmed. Thus, it will be signed by the customer and, in the end, sent to the OMS and ERP, when a connection is available.

Hypothetically, even customers (and not just agents) might use the Fashion Touch App; from the App point of view, it would be enough to create a profile for customers only. However, this profile has never been created since the fashion industry has never expressed this need.

Fashion Touch is divided into three sections: synchronization, orders and customers.

Synchronization allows to keep the device data consistent with those of the server (see figure 3); which is essential to avoid the inclusion of orders on old data, such as items that are no longer available for the current season. Fashion Touch, through a process of authentication, will download only the data that the user is authorized to access (the customers, the seasons and lines for which agent can make orders), diversifying for each agent the access to the data catalog.

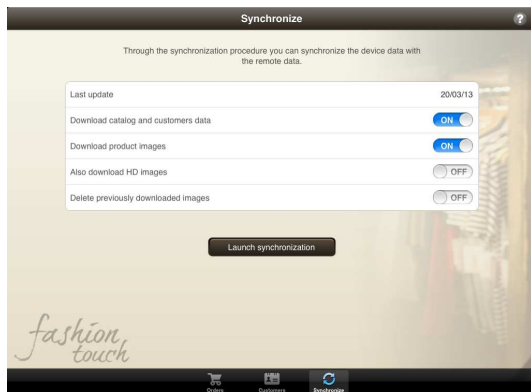


Figure 3: The synchronization tab.

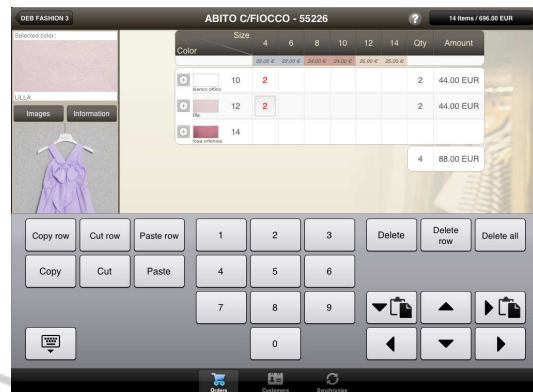


Figure 5: The use of the custom keyboard.

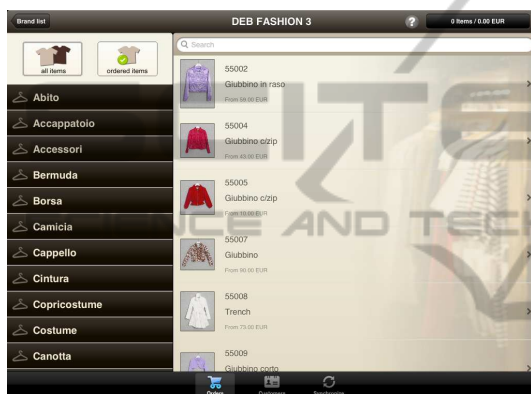


Figure 4: The product catalog.

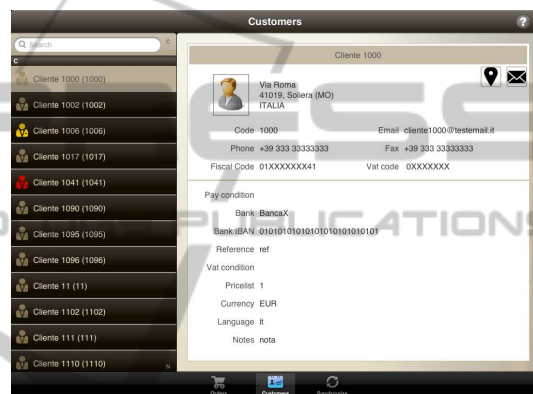


Figure 6: The management of customers.

The section of **orders entering** guides the agent through the customer, season and line selection. It allows to:

- browse the product catalog in different ways (google-style search, search by item code, search by category and by item name) (see Figure 4);
- choose the quantity and type of items;
- display and zoom in an item;
- get detailed article informations;
- finalize the order and send it directly to the corporate server.

The manual work of both the agent and the back office is thus dramatically softened.

In the section **customers** it is possible to see the complete list and display the main information for each of the customers. The main functions are:

- offline browsing of the full multimedia catalog (items images);
- users assistance in creating order through customer, line and season selection;

- features for preparing and editing seasonal orders and managing coordinate articles;
- a custom keyboard to simplify and speed up the order entering process: copy, paste, cut order line quantity and also copy several lines at once (see figure 5);
- the capability of viewing and changing the order details before processing it;
- for each item it is possible to view the images, the basic informations and the price ranges for each size;
- display informations about the customers with the possibility of sending e-mail and geolocalize them (see figure 6);
- multi-language support.

The application is completely integrated with the most popular information systems (in several cases the communication is performed through CSV files). The App is, thus, able to communicate orders automatically to internal business ERP. The continuous dialogue between the ERP and the application always guarantees the sales network to have an updated and reliable content.

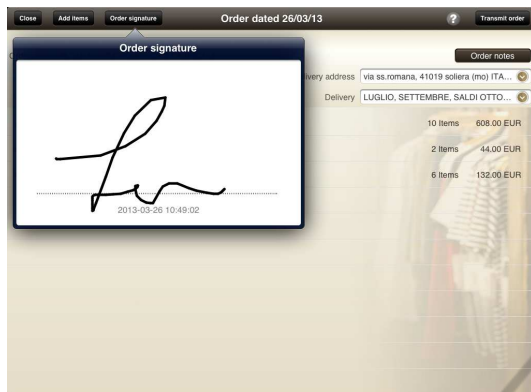


Figure 7: Signing an order.

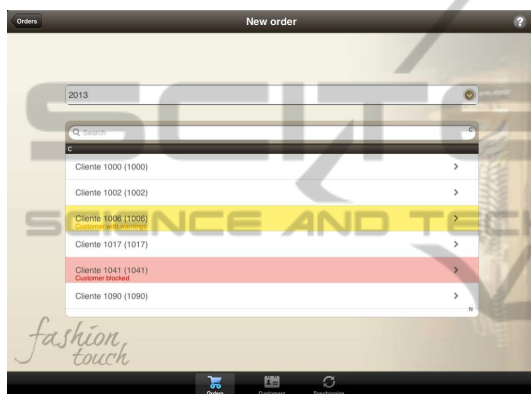


Figure 8: The management of blocked or warning customers.

The test of the app on real fashion industries has lead to implement some additional features:

- sign the orders - when the order is confirmed, it is also possible to sign it (see figure 7), and send the image of the signature to the server; although this is not legally binding, it represents a guarantee for the supplier;
- complete order management on iPad - error line and error header can be handled during the order transmission, this allows agents to complete the entire order process directly on the mobile device, the order is confirmed till the last step of transmission to the ERP -
- “mini” keyboard - the use of a simplified keyboard in the detail window and change order window supplies better usability;
- “tap” feature - an easy and quick feature to increase the quantity on a field, at each tap the amount increases by one, thus avoiding the use of the keyword for simple operations;
- double order lines - for the same item in the same color (using a + icon next to the line item details

in that “splits” line by creating a new equal);

- box management - a box is a predefined mix of items for the same article in different size; an article can be purchased in quantities chosen by the user, or in boxes;
- blocked or warning customers management - for control purposes, the fashion company wants to block the order emission for insolvent customers, the App has been refined in order to block the creation of an order or to advise about a critical situation regarding a customer (see figure 8).

Fashion touch is available for download on the Apple Store and can be tested by using a demo server configuration ⁷

The Fashion Touch App integrates two modules taken from the Titanium Marketplace⁸: one for the graphic editor for the customer’s signature and one for creating the zip json order, which is attached to an email to be sent to the assistance in case of problems while transmitting the order. For future implementations we intend to exploit another usefull module that allow reading barcodes.

4.2 The App Development

As already mentioned, the App has been developed using the *Appcelerator Titanium* framework. We have made use of the Titanium guidelines of good programming, organized the code in modules following the MVC (Modelling View Controller) pattern for web application development.



Figure 9: The software architecture of Fashion Touch App.

The software modules have been implemented according to the task that they have to perform in the App (see Figure 9):

- Database management module (DBMM): used to create and manage tables as well as open and close

⁷The App is available for download at <https://itunes.apple.com/it/app/id576756392?mt=8>

For receiving the test parameters please contact Quix s.r.l.

⁸<https://marketplace.appcelerator.com/>

database connections; it provides an abstraction of the db through the methods of data access;

- File System management module: used to manage the access to the file system. In particular, it is used for the images management;
- Network management module: used to manage and authenticate http requests to the server;
- User Interface management module: used to create the GUI; it consists of several sub-modules, one for each application sections;
- Orders management module: used to abstract the object related to an order;
- Customers management module: used to abstract the object related to a customer;
- Lines management module: used to abstract the object related to a line;
- Prices management module: used to abstract a price software object;
- Products management module: used to abstract a product software object;
- Season management module: used to abstract the object related to a season;
- Utilities management module: object that serve different utility tasks necessary to run the application.

Orders, Customers, Lines, Prices, Products and Season management modules use the DBMM for data abstraction.

In this section, we want to focus on the DBMM. The DBMM uses as database engine SQLite⁹. The DBMM faced several issues:

- The database schema of the web application was very complex and difficult to replicate in a mobile device: this highlighted a problem of choosing which data to import into the App and which data to omit;
- The web application database included different data types while *SQLite* supports only few ones;
- The type and amount of data stored in the App database have to allow maximum optimization in terms of performance when synchronizing with the server database and using it within the application.

In order to deal and solve these problems we acted on three different levels: we selected *JSON* (Javascript Object Notation) as the data format of the schema, we simplified the App database and, in the end, we unified different data types together.

⁹<http://www.sqlite.org/>

*JSON*¹⁰ is a text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. It is language-independent, with parsers available for many languages. *JSON* relies on two structures: (1) a collection of name/value pairs; (2) an ordered list of values. *JSON*'s basic types are number, string, boolean, array, object and null. Virtually all modern programming languages support these universal data structures in one form or another. A data format that is interchangeable within different programming languages might be based on these structures. The *JSON* format is the data format that has been used within the application, thus, all the complex data have been stored into the database tables as blob object in the *JSON* format.

To optimize the device performances and the developing complexity, we opted for a simplification of the App database schema. We decided to use only subset of tables; for each table we selected only the data strictly necessary for the needs of the App. In particular, the database tables were built reporting only the columns needed for the query, plus a generic column consisting of a *JSON* object containing the rest of the data.

Also, integrating data types was a convenience choice to improve the speed of the application development. All fields of the implemented tables contain a *string data type*. This choice was also related to the decision to use network services that return as output data in *JSON* format and so having fields in string data format.

In the end, by the network management module the App was able to perform every operation that needs the server synchronization. The network management module implements several HTTP calls ensuring the user authentication to the server to which the call is made.

5 CONCLUSIONS AND FUTURE WORK

This paper depicted all the phases leading to the development of an innovative mobile application for fashion trade. Fashion Touch is an effective application which improves and eases the commercial agents working approach by speeding up the process of orders acquisition, removing all forms of paperwork (with all the advantages in terms of costs and environmental impact), enabling a rapid and handy cata-

¹⁰<http://www.json.org/>

log consultation and ensuring agents to deal with data up to date w.r.t. the ERP.

The app optimizes the process of orders acquisition by sales agents working in the fashion industry, appreciated and used by several hundreds of users. Its implementation has become a point of strength among Quix's products. Furthermore, the development of more advanced functionality is an opportunity for continuous work and commitment, as well as for the study of new solutions of implementation.

Fashion Touch participated at the Smau Mob App Awards, which took place in Bologna June 7, 2012, by competing in the *Task Specific* category. The Fashion Touch App ranked between the four finalists among hundreds of participants.

The application is currently being used by three companies with a network of around 200 agents, therefore it is continuously updated and integrated with new features following the clients' needs.

Some of the more interesting features under development are:

- *Copy of an order* (for transmitted or confirmed orders): the user selects an order and click on the copy button, a popup appears where it is possible to select the new customer, then the system creates a new order as a copy of the selected one with the proper prices for the selected customer;
- *Printing Orders*: at any time of the order creation process it is possible to visualize the pdf file, even if the order has not yet been transmitted; this is useful in case the customer wants to keep a copy of the order and confirm it later;
- *Management of the minimum number of items*: it is possible to set the minimum purchase quantities for an article (or an article in a specific color), therefore the constraint is that the agent will have to purchase an amount greater or equal to the minimum established for that item;
- *Search for articles by reading bar codes* (this functionality will be available thanks to a module from the Titanium Marketplace), the agent rather than navigate the standard catalog, scans a bar code through the camera of the iPad (which identifies an item in a particular color), the reading and identification of the code re-directs the agent towards the insert page where he has to insert the quantity for that item; the agent continues to enter new items and so on.

ACKNOWLEDGEMENTS

The work hereby presented has been possible thanks to the internship at Quix s.r.l. of Ivano Baroni master student in Computer Engineering at the Department of Engineering "Enzo Ferrari" University of Modena and Reggio Emilia. Particular appreciation goes to Andrea Prandini and Sara Quattrini, respectively one of the founder and an employee of Quix s.r.l., for their suggestions and collaborations during the internship and the valuable comments on the paper.

REFERENCES

- Ballard, B. (2007). *Designing the Mobile User Experience*. John Wiley & Sons, 1. Auflage edition.
- Meschtscherjakov, A. (2009). Mobile attachment: emotional attachment towards mobile devices and services. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '09*, pages 102:1–102:1, New York, NY, USA. ACM.
- Ohr, J. and Turau, V. (2012). Cross-platform development tools for smartphone applications. *Computer*, 45(9):72–79.
- osservatori.net, I. . M. (2012). Mobile marketing & service: finalmente un'accelerazione. In *Proceedings of the 2010 Annual Conference & Exposition*. Politecnico di Milano, Dipartimento di ingegneria Gestionale.
- Patrick Olson, Mudasser Wyne, A. O. (2010). Android based mobile order management system. In *Proceedings of the 2010 Annual Conference & Exposition*, volume 1, pages 4605–4617. American Society for Engineering Education Annual Conference and Exposition 2010.
- Platzer, E. (2009). Emotional factors influencing technology acceptance particularly with regard to mobile service usage. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '09*, pages 95:1–95:1, New York, NY, USA. ACM.