EAI Endorsed Transactions
on Ubiquitous Environments
Research Article ICST.ORG

An Un-tethered Mobile Shopping Experience

Venkatraman Ramakrishna[1],*, Saurabh Srivastava[1], Jerome White[2]

[1]IBM India Research Laboratory, New Delhi – 110070, India
[2]IBM India Research Laboratory, Bangalore – 560045, India

Abstract

Smart phones with access to apps from online stores are ideal candidates to replace expensive hardware like Point-of-Sale terminals for retail. A standard set of shopper and retailer apps can replace the conventional retailer IT setup in settings ranging from rural areas with low connectivity to dense urban areas. We describe how we built such a set of apps for mobile shoppers and retailers equipped only with smart phones and tablets, and who require little to no training to use them. These apps are flexible enough to be used by small shops with small inventories as well as large grocery chains. Our apps enable retailers to manage their inventories and finances, and shoppers to discover retailers, match shopping lists, and make purchases. We describe how we conducted a user study of retailers in North India to ascertain their needs, and to understand the ecosystem in emerging markets. This helped us build a mobile shopping platform that can support a range of transactional scenarios.

Keywords: Mobile Shopping, Mobile Payment. Emerging Market, User Study.
Received on 28 February 2014, accepted on 13 August 2014, published on 17 November 2014.

doi: 10.4108/ue.1.3.e1

1. Introduction

"*There's an app for that*" was Apple's marketing slogan back in 2009, which tried to convey the impression that Apple devices and the apps running on them could support virtually any function desired by users. Though this slogan was the subject of jokes at the time, it was prescient in many ways. With other devices and operating systems (Samsung, Android, Microsoft, Windows, etc.) replicating Apple's smart mobile device success in the marketplace, we may indeed be heading towards a future where one can build an app for literally anything, and have it run on a combination of mobile devices, be they phones, tablets, or even spectacles (e.g., Google Glass) and watches. Also, such apps could benefit places and industries in which technology has not yet made an impact.

Our focus is on the retail industry, which is still growing at a fast clip in emerging markets where the consumer base continues to expand and increase its purchasing power. Yet, retailing is a very competitive business with high barrier to entry because of the relatively large capital investment necessary at the outset. Economies of scale provide such capital in the developed markets. But in emerging markets, there is a paucity of research and data on the retail ecosystem, the needs of retailers, and the mechanisms to promote growth. We conducted a user study in North India, an emerging market, to obtain exactly such data, and found that retail is predominantly carried out in small stores that almost exclusively use cash; inventory management and transactions are ad hoc, inefficient, and unreliable. Technology, i.e., hardware (e.g., PoS terminals) and software, is expensive and also requires reliable electricity and network connectivity; small corner shops and rural shops cannot afford this. Allowing retailers to run their shops using their personal mobile phones (which are ubiquitous, even in poor countries) or tablets will remove the primary entry barrier and provide a huge boost to the retail industry [9]. Even in mature developed markets, enabling store management through mobile devices would allow the large shopping chains to expand at lower cost into smaller towns and rural areas through franchising.

Not only will mobile devices help retailers run and expand their business, they will also provide customers with a more seamless and richer shopping experience. Traditional online payment systems using credit and debit cards have been augmented by *mobile money* systems [1][4][5][6][8], which enable customers to make purchases using a handful of clicks. Yet such systems are not going to be popular with customers, even in developed countries, unless retailers adopt systems that are compatible with those used by customers. Though such systems do exist, like Google Wallet [4], their adoption rate is low as they require the retailer to possess special hardware. If retailers were to run their stores out of their mobile devices, more customers

*Corresponding author. Email:vramakr2@in.ibm.com would use mobile money mechanisms without being tied to a service provider. In emerging markets, where the banked population is a small subset of the cell phone-using population, telecom operator-provided money accounts help shoppers make reliable purchases easily [8]. Introducing computers into the shopping experience enables long-lived customer-retailer relationships, with loyalty points; such features are currently available only in high-end retail.

In this paper, we describe the design and implementation of a suite of universal retailer management and shopping apps. These apps require no infrastructure other than mobile devices with the capability to run applications and to communicate with other devices. We show that the set of functions enabled by this suite is both minimal and complete for retail shopping scenarios. These apps are of use to shopkeepers and customers in a variety of settings, from urban to semi-urban to rural, and with intermittent to no connectivity. Our research contribution is twofold. First, we conducted a user study of small-scale retailers in two cities in North India to understand their current modes of operation, and to get an understanding of their pain points and needs that could be addressed by technology. Second, we used the knowledge gained from this study to determine the essential functions that retail applications need to perform for retailers and customers. Further we designed and implemented proof-of-concept applications on minimum-function smart phones to demonstrate that portable ubiquitous retail was possible without being tied to location or expensive specialized technology. Analysis of these apps reveals that shopkeepers and retail store clerks with no training can quickly incorporate them in their respective workflows.

As a suite of applications running on mobile devices, our retail platform is both extensible enough to incorporate additional functions, and generic enough to support a range of transactional scenarios, retail being the most prominent use case. By providing a technology-enabled shopping experience that is not tethered to location or to specialized difficult-to-port hardware, we make a contribution to the field of ubiquitous computing. Our solution takes advantage of the present and coming ubiquity of mobile devices and communication technology.

Here is an outline of this paper. Usage scenarios are described in Section 2, and a related work survey in Section 3. The results of our user study to ascertain the needs of retailers in emerging markets are presented in Section 4. The design of our platform and application suite to realize such scenarios is described in Section 5, with implementation details in Section 6. We evaluate our platform in Section 7, and make concluding remarks in Sections 8. (*Note*: we use the words *customer* and *shopper* interchangeably throughout this paper to refer to the consumer of a service.)

2. Motivating Retail Shopping Scenarios

In this section, we present two representative retail scenarios, and discuss other transactional scenarios that resemble retail in their essential aspects.

2.1. At the Corner Shop

Peter needs to purchase milk, cereal, spices, and other grocery items. He walks toward the small grocery store in the neighbourhood run by a retailer, John. At the store, Peter picks some of the items he can find. Other items are currently unavailable, and John is too busy tending to his accounting and managing other customers to suggest alternative stores. In his hurry, Peter forgot to bring adequate money, and so has to leave behind some non-essential items. While returning home, Peter suddenly remembers that he forgot to purchase milk, one of the essential items on his list.

What we see above is a mildly complex shopping scenario, the result of which is far from optimal or satisfactory to shopper or retailer. Maintaining a list of essential items that Peter will not forget, matching those items with John's inventory, getting suggestions for alternative stores, paying John, and store inventory management for John, are tasks that can be easily done using computers and networks. John can enhance his relationships with faithful customers by maintaining persistent accounts for them and giving rewards and discounts. Automating these tasks will reduce the cognitive burden on users, and this does not require either party to spend much money as such tasks can easily be performed on the cheap smart phones they already possess.

2.2. At the Supermarket

John bikes toward the supermarket to do his groceries. The store is crowded, and it takes John some time to discover where many of the items he needs are kept. He then waits patiently in line at a PoS-enabled checkout counter. Just when his turn arrives, he remembers that he needs biscuits. The clerk tells him that biscuits are indeed in stock, but John does not have the patience to go through the whole process again. After he pays cash, the clerk hands him a discount coupon valid for a week. On the way to his bike, John loses the coupon, and with it, the discount he is entitled to.

As in the previous scenario, we see various tasks that can be automated but are not. If the store possessed devices capable of communication, John's phone could automatically communicate with them to determine what items are available and in what room and rack; a small store map could be exchanged as well. The store could also alert John about the presence and location of biscuits, an item he forgot. To safeguard John's discount offers, persistent customer records could be kept in-store and on John's phone; this would make transactions easier, more pleasant, and reliable.

2.3. Other Example Scenarios

The above examples roughly cover the range of retail businesses and customer shopping habits, especially in emerging markets, though one could think of enhancements.

Yet these are not the only businesses that could benefit by automating interactions among service providers and consumers. The restaurant business is an example, where customers would love to be aware of all available choices before they decide to patronize particular restaurants, and where restaurants would love to establish long-lived relationships with their patrons. Bookstores offer another example. A customer often has to visit multiple stores to purchase all the books he/she seeks. Therefore, a store that proactively orders books, or places back-orders, for their faithful customers will see its business flourish.

The above examples of restaurants and bookstores, not to mention retail stores, can be extrapolated to any number of scenarios that involve patron-provider relationships. By *renaming* the shopper and the retailer, we can create virtually identical transactional scenarios that are currently not being realized, but can be with the assistance of technology. We choose to focus on the retail scenario in this paper because of its increasing ubiquity in emerging markets, the possibility of acquiring large amounts of credible data through user studies, and the high prospect of field deployments and evaluations. In the remainder of this paper, we show exactly what storeowners and shoppers need through field surveys conducted by us, and present the design of a universal set of shopper and retailer apps to realize the scenarios described in this section.

3. Related Work

The proliferation of mobile devices has triggered significant work in building applications and services that provide value to customers as well as businesses. A large portion of this work involves building transactional applications, particularly shopping or retail. Yet the research or business focus has generally been on one or more parts of the shopping experience rather than the experience as a whole. Also, a lot of the practical work in this area focuses on either the customer or the retailer, but not both. In this section, we survey the work done in enabling mobile shopping through technology, and also examine user studies conducted to determine the needs of mobile shoppers.

Mobile commerce in particular is an area that is active in both the commercial and academic space. In the last few years, several companies have developed, and brought to market, commercial solutions for facilitating mobile payments. Further, they have done so in various markets around the world. Academia has also contributed to the area, usually by proposing systems that either address the end-to-end challenge of mobile commerce or solve a particular problem. While much of their focus has been on backend processing, work has also been done on interface design. This section takes a closer look at a wide range of these efforts.

3.1. Commercial Services

There are several wallet services that allow users to pay for goods via their mobile phone; e.g., Google Wallet [4], Square Wallet[†] [5]. In all cases, customers use their mobile devices to pay for goods via proxy. That is, the merchant and the customer both have accounts with the respective service. When a transaction is made, the underlying service handles the monetary exchange using pre-existing and verified traditional-bank details.

Boku is a service that allows users to pay for goods via their mobile phone by charging the transaction to a user's mobile service bill [6]. The concept, known as "carrier billing," allows customers without traditional methods of payment, such as credit cards or bank accounts, to purchase goods electronically. As such, the service is highly reliant on existing telecommunications carriers.

Airtel Money allows customers to pay bills, make purchases, and transfer money via their mobile device [8]. Users setup an account with Airtel that is debited each time they make a transaction through their Airtel SIM card. The service uses USSD, and is thus mobile agnostic.

Square Register is a specialized card-reader that allows merchants to accept card transactions via their mobile devices [5]. Requiring such specialized hardware, though, makes this solution less likely to be adopted in small shops, especially in emerging economies.

M-PESA is a popular mobile money solution, deployed widely in the developing world [1]. M-PESA itself is a branchless banking service: users register and transact physical funds through a network of certified M-PESA agents. The deployment, and interface, across mobiles is managed by participating telecom service providers; generally through SMS or USSD.

3.2. User Studies

Medhi et al. compared mobile money user interfaces amongst low-literate users [2]. The authors studied the usability, with respect to interface, of popular commercial mobile money services in several developing countries. Based on their initial ethnography, they developed and tested three interfaces across the target population. They concluded that speech- and picture-based interfaces, as opposed to text-based UIs, were more viable options for low-literate subscribers.

A study of money practices in rural Ethiopia revealed that existing mobile money research is biased toward technical contributions [7]; ideally it should consider users' monetary practices as well. In particular, the author advocates that when considering low-educated users in the developing world, religious and cultural practices should be considered alongside economic and social requirements.

These user studies focus on the needs of the mobile shopper and the mobile payment infrastructure provider. We

[†] Square Register and Square Wallet are developed by Square, Inc.

went one step further by conducting a study of retailers to determine their needs (Section 4), which helped us design the end-to-end shopping experience (Sections 5 and 6.)

3.3. System Design

M-Cash is a mobile money transfer service proposed for low-resource environments [3]. Transactions are made via an SMS interface and handled, at the SMS gateway, by a web-service middleware. The authors give very little detail as to how the service is unique, and, base their results on simulation rather than deployment.

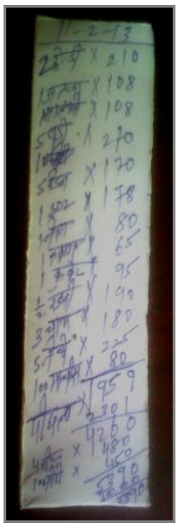
mFerio is a mobile payment solution that uses NFC to facilitate payment through the mobile phone [14]. To transfer money, parties use a "two-touch payment protocol" to negotiate a transaction. In the first phase, seller and buyer touch their devices to begin the payment process. They pull away, and then touch again to finalize the transaction. The authors show that the protocol strikes an optimal balance between usability and security. Their user studies took place in a controlled lab environment, using a population largely comprised of Singaporean undergraduates; thus, it is hard to say if their solution would be suitable for a low-literate Indian population.

Hassinen et al. present a mechanism for mobile payment with real or virtual PoS systems [10]. Rather than enforcing a strictly mobile-base PoS solution, the authors outline a rich set of communication options—SMS, Bluetooth, and traditional IP—to facilitate data transfer. Other solutions exist that utilize NFC and Bluetooth for money transfer (Monteiro et al., [11]; Pradhan et al., [12]; Zdravkovic, [15]). Massoth and Paulus outline an inventory management system in which the Blackberry platform is used to communicate sales information to a centralized server [13]. The primary focus of these systems is on a particular aspect of the shopping experience; namely payment or inventory. Unlike the work in this paper, they do not offer an end-to-end solution for mobile shopping. Further, none has been studied in the context of a low-literate population.

The goal of our project was not to investigate any one part of the shopping experience in depth, but rather to understand the emerging market retail ecosystem better through a user study and design a basic, yet practically usable platform. On that yardstick, our work clearly builds on and adds to existing research.

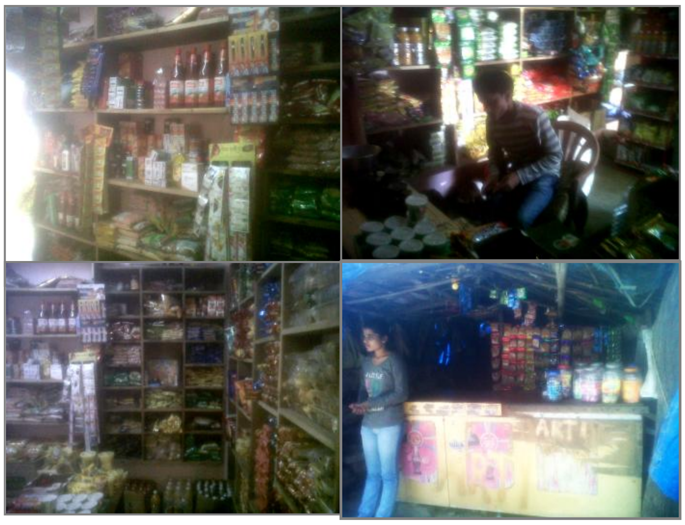
4. User Study

To understand the current ecosystem of small, medium and larger enterprises, we conducted a field study with 28 small-to medium-scale businessmen in two cities in North India: Delhi, a large city, and Lucknow, a smaller one. Five were college graduates, 9 were educated until Class 12 (K-12 system of Education), and the remaining 14 were illiterate. All of them possessed mobile phones. Their businesses ranged from mid-size grocery stores, betel shops, and street-side food shops, to medium size grocery stores. We were interested in the following: (i) *how* these businesses worked, (ii) *what types of transactions* they performed, (iii) *context* of individual businesses, and (iv) *challenges* faced in conducting their businesses. To help understand these questions we carried out a Contextual Enquiry: we interviewed each participant separately for an hour, and collected and analyzed their responses.



11-2-13
[illegible] X 210
[illegible] X 108
[illegible] X 108
[illegible] X 270
[illegible] X 170
[illegible] X 178
[illegible] X 80
[illegible] X 65
[illegible] X 95
[illegible] X 190
[illegible] X 180
[illegible] X 225
[illegible] X 80
1959
[illegible] X 2301
4260
[illegible] X 480
[illegible] X 450
[illegible]

(a)



(b)

Figure 1. (a) Supplier-Provided Inventory List for Low-Literate Shopkeepers. (b) The Grocery Stores

4.1. Study Findings

We found that *money management* is a prime concern for all businesses, irrespective of size. Our investigation revealed that tracking sales records and management of funds in spending, borrowing, repaying, investing, and savings was a routine task for most of the participants. Also, it was common for suppliers and retailers, especially in small-scale businesses, to sell certain items on credit. Though numerical literacy was good, most of the low literate retailers were dependent on the suppliers to maintain their inventories (Figure 1a). Credit and borrowed amounts were communicated verbally and memorized instead of maintaining persistent records. Some businesses, like ice cream or roasted peanut sales, were seasonal. Family grocery stores were common. These were small to medium stores offering daily commodities such as milk, mineral water, flours, spices etc. Most of these retailers were socially connected to their customers and had high levels of trust in each other. The larger stores typically kept central servers, associated with the point of sales kiosks, to maintain sales records.

Most of the participants emphasized the importance of *expiry management* of the commodities they sold. We found that a supplier often lent commodities to a retailer on trial basis, and the retailer had an option either to pay the supplier or to return the commodity if he could not sell it. Some study participants reported that they maintained notebooks to keep records of products that were close to expiration. We also observed that small and medium stores commonly maintained inventories that were lower than their estimated sales, especially of perishable goods; large stores with good supply chains maintained larger inventories. Other challenges faced by our study participants included lack of information about new offers on products and workforce management. Some participants expressed the desire to have multiple alternate vendors as fallback options and not depend on a single vendor to supply goods.

We categorized businesses based on their inventory volumes. Through an affinity mapping, we identified factors that influenced these categories of businesses. Our study results are summarized in Table 1. (In the future, we intend to augment our knowledge of the Indian retail industry by surveying large chain stores.) The results of our surveys give us insights into the features that are both useful and necessary in the apps we want to build, regardless of the size of the stores. In no particular order, these include: payment protocols, inventory management, and enabling many-to-many searches and associations between customers and stores.

Table 1. Factors Influencing the Retail Business, Categorized by Inventory Volume

	Perishable Inventory	**Low Inventory**	**High Inventory**
High Priority	Money management	Money management	Money management
↓	Expiry date management	Stock & estimation management	Expiry date management
	Stock & estimation management	Over-expected sales	Stock & estimation management
	Over-expected sales	Alternate vendors	New offers
	Workforce management	New offers	Over-expected sales
Low Priority	Alternate vendors	Expiry date management	Workforce management
	New offers	Workforce management	Alternate vendors

5. Platform Architecture and Protocol

The bare minimum hardware requirement for our target scenarios is a device that can run application code, communicate, and have the ability to be discovered. Any smart mobile phone or tablet, from low-end devices to sophisticated ones, will satisfy these requirements. Our software is designed to provide a seamless and uniform user experience regardless of the OS, be it Android, Windows, iOS, Blackberry OS, etc. We use *app stores*, which are offered by most phone system service providers, to deploy and update our platform apps. For communication, we use application-layer protocols that are independent of the underlying communication technology, which can range from IR, Bluetooth, and Wi-Fi to the broader Internet.

Our retail shopping scenario platform enables: (i) discovery of, and association to, stores by customers, (ii) exchange of product information, and (iii) financial transactions. A minimal instance of the platform software can be logically split into three parts: (i) a *shopper app*, (ii) a *retailer app*, and (iii) a *shopper-retailer protocol*. In addition, these applications may choose to link with remote databases for reliable data backup and synchronization. For larger commercial manifestations of our platform, separate financial services gateways can be used to manage money accounts and mediate shopper-retailer financial transactions. This architecture supports a minimal and complete set of functions required by a retailer and a shopper to transact with, and maintain a long-term relationship with, each other.

5.1. Shopper Application

The shopper's device runs an all-purpose application designed to perform all the transaction functions its user requires (Figure 2). In effect, the device acts as a localization engine, a search engine, and a mobile wallet all rolled into one.

The *discovery and communication* module locates retail stores that are reachable from the shopper's device, identifying stores it has a prior relationship with, and associating with a store as desired by the shopper or using a pre-configured policy. This module abstracts the low-level networking details from the core shopper app and offers a *send*/*receive* API for communication with the retailer app. Two of the abstracted details are: (i) *fault tolerance*: once associated, it tries to automatically reconnect if the original connection was broken or if the network adapter was reset, and (ii) *protocol type*: the devices may connect using Bluetooth, IR, WiFi, or even through the Internet, but the application logic does not change.

The *database* stores transaction and retailer information. This includes lists of products created by the shopper, known retailers, purchase history, and reward points. For additional reliability, this database may also sync with a remote database or cloud service.

The *negotiation* module contains the core shopping app logic for querying retailers and directing transactions. Once an association is made with a store, it sends a shopping list to the store to determine what items are available and at what price. It also determines how payments are to be made (i.e., applying reward points and discount offers in addition to money transfer) before triggering a payment.

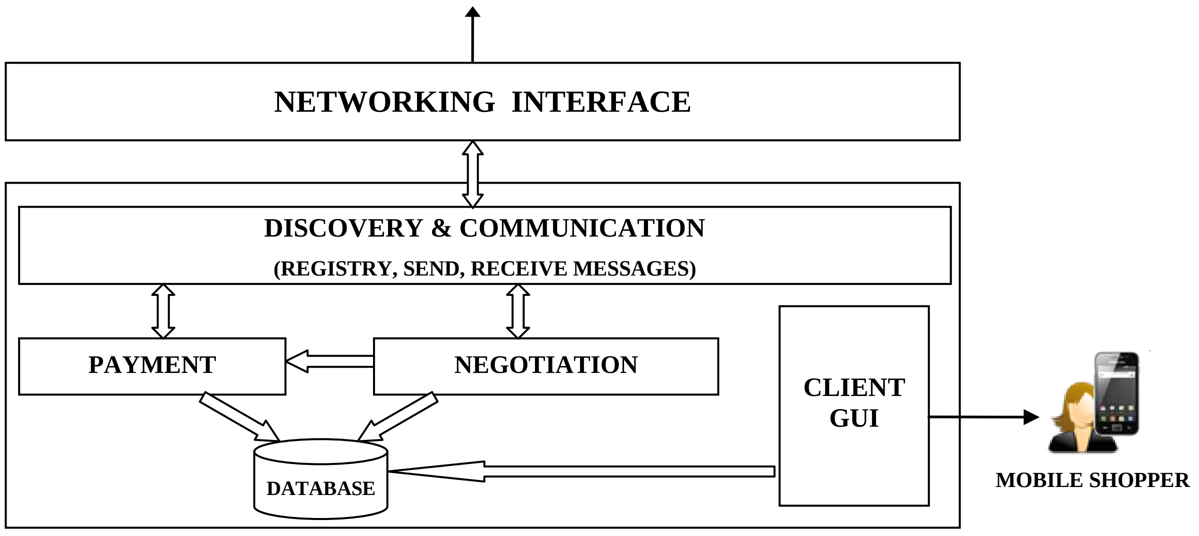


Figure 2. Shopper Application Architecture

The *client interface* is an interactive GUI allowing a shopper to create, update, and remove shopping lists, select stores to associate with, and to approve payments. The interface can be tailored to suit specific devices and operating systems, without changing core application logic; if based on HTML, it will be platform independent.

Lastly, the *payment* module is responsible for completing a reliable financial transaction when triggered by the negotiation module. It is configured to conduct tri-party protocols with the retailer and a remote financial entity (e.g., bank) for payments through bank accounts or credit cards. For ease of use, it may store the user's account or credit card information (accessed securely using appropriate authentication) as well. Optionally, it may maintain a *mobile wallet* for the user if both shopper and retailer apps are configured to use a *mobile money* protocol [8].

5.2. Retailer Application

The retailer's device runs an all-purpose application designed to manage the store as well as conduct transactions with shoppers, as illustrated in Figure 3.

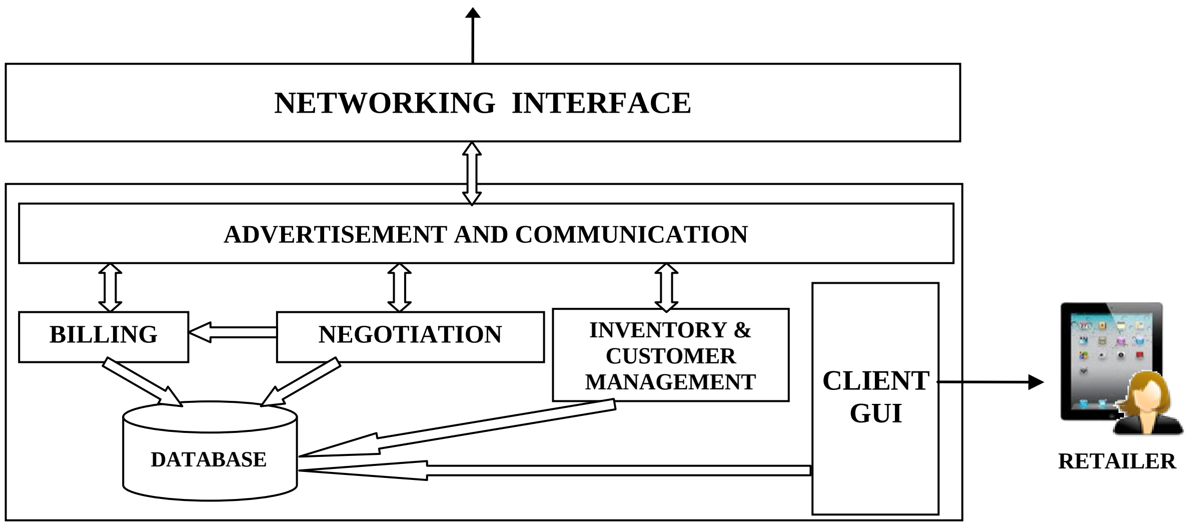


Figure 3. Retailer Application Architecture

The *advertising and communication* module is the counterpart to the shopper's discovery module. It ensures that the app is discoverable on a wireless channel (IR, Bluetooth, WiFi, etc.) or through a public REST API on the web. It is responsible for receiving incoming connection requests, and hides the low-level details of the channel by exporting a *send/receive* API to the core application, whose logic is independent of the communication details. This module ensures that the store continuously advertises its presence, or decides when to advertise based on a pre-configured policy.

The *database* stores information about products sold by the retailer, and information indicating whether they are available or on back-order. Transaction records are maintained for every identifiable customer who has conducted mobile transactions with the store. For additional reliability, this database may periodically sync with a remote database or cloud. Alternatively, a large retail store with multiple checkout clerks (each with a mobile device) may have a database running on an in-store server, which is used to sync with the clerks' devices, using a standard 2-phase commit protocol for transaction integrity.

The *management* module performs two functions. Through *inventory management*, the retailer can manage the products currently being sold, and identify the locations (e.g., room, rack) they are placed in. It also allows the retailer to query distributors for information on products not currently at the store, and to place orders. *Customer management* allows the retailer to manage his customers' accounts, view customers' purchase history, and add or remove reward points and discount offers.

The *negotiation* module is the counterpart to the shopper's negotiation module. Upon receiving a shopping list, it matches the items with the products in its inventory, determines what items are available, and returns this list to the shopper with associated pricing and reward points information. For a first-time association, it automatically creates an account for the shopper based on his device ID. For products currently unavailable, a notification may be raised for the retailer to eventually act on, or an order automatically placed with a distributor if a suitable API is configured.

The *client interface* is an interactive GUI that enables a retailer or store clerk to view inventory, customer, billing, and purchase information. It also allows users to change inventory contents, update customer records, and process bill payments.

Lastly, the *billing* module, when triggered by the user, sends a shopper's app a payment request for billed items. Subsequently it conducts a secure financial transaction with the shopper, the money being transferred to a pre-registered financial account. An invoice is sent to the shopper's app.

5.3. Shopping Protocol

The transaction protocol between the shopper and retailer consists of a mix of user actions (e.g., selection of items to purchase) and automatic processes (e.g., discovery of retailers, matching requested products with available items in the inventory). The complete workflow is illustrated in the flowchart in Figure 4.

We attempted to design a minimal and generic protocol for transactions in our target scenarios, and every intermediate step in the flowchart in Figure 4 provides a function that is necessary for a complete retail transaction to occur. While augmentations could be made—more security features, for example—such discussion at this level of detail is beyond the scope of this paper.

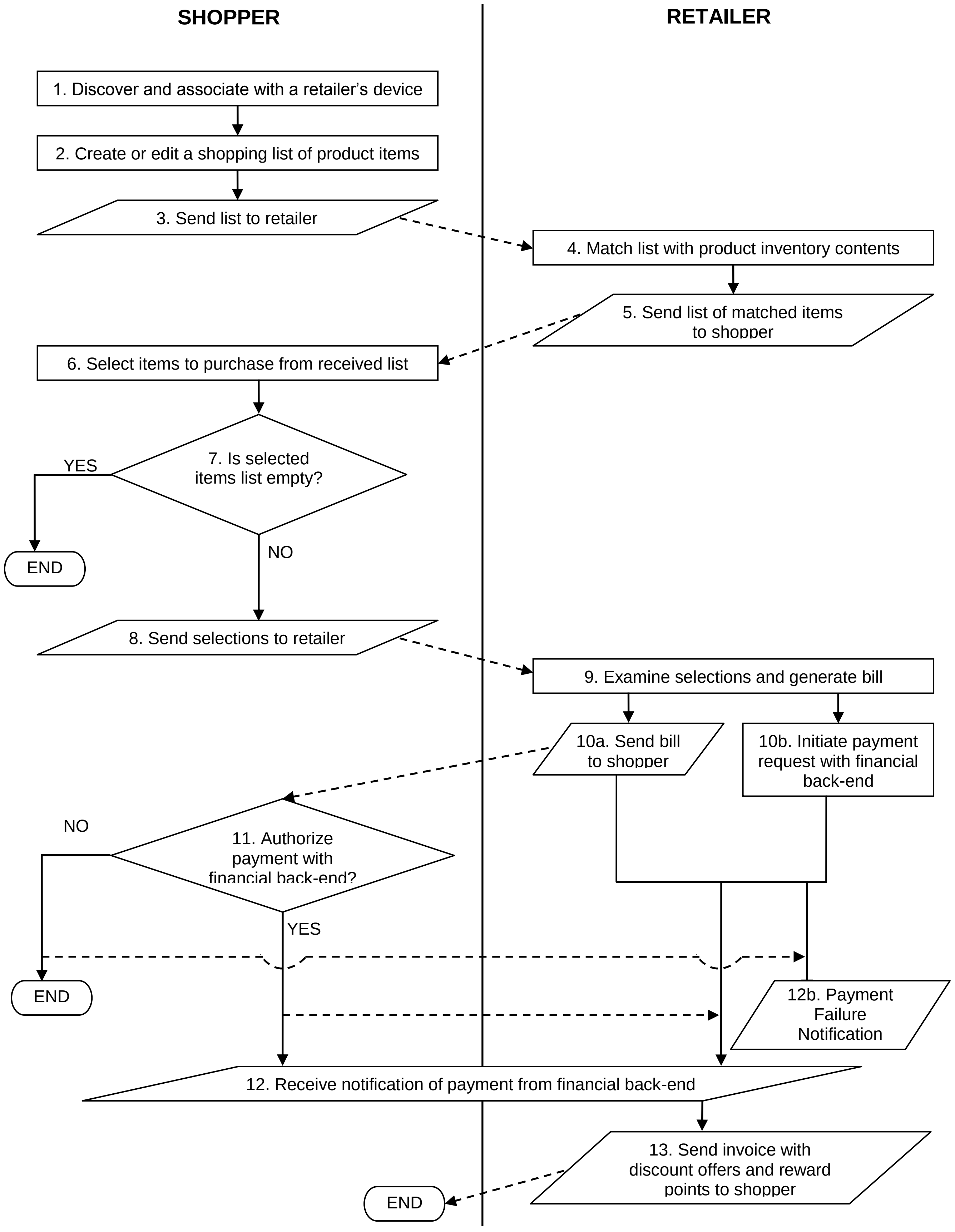

SHOPPER
RETAILER
1. Discover and associate with a retailer's device
2. Create or edit a shopping list of product items
3. Send list to retailer
4. Match list with product inventory contents
5. Send list of matched items to shopper
6. Select items to purchase from received list
7. Is selected items list empty?
YES
END
NO
8. Send selections to retailer
9. Examine selections and generate bill
10a. Send bill to shopper
10b. Initiate payment request with financial back-end
11. Authorize payment with financial back-end?
NO
YES
END
12b. Payment Failure Notification
12. Receive notification of payment from financial back-end
13. Send invoice with discount offers and reward points to shopper
END

Figure 4. Steps of a Generic Shopping Protocol

We also do not allow the billing process to be completely automated, as Step 6-9 indicate (the retailer's app waits for a user's selections before triggering the payment protocol.) This is because, in the most general case, stores must physically inspect items at checkout. Few stores currently have the technology to automate this process without being susceptible to shoplifting.

Information Representation: We currently encode the communicated information as JSON strings, though we plan to use XML in the future. Our shopper and retailer apps were developed together with a common vocabulary, which enabled each to understand the other's message semantics. This solution may not work when the apps are developed independently. In the future, we plan to avoid this problem by publishing our data structure schemas in RDF and XML for public use and extension.

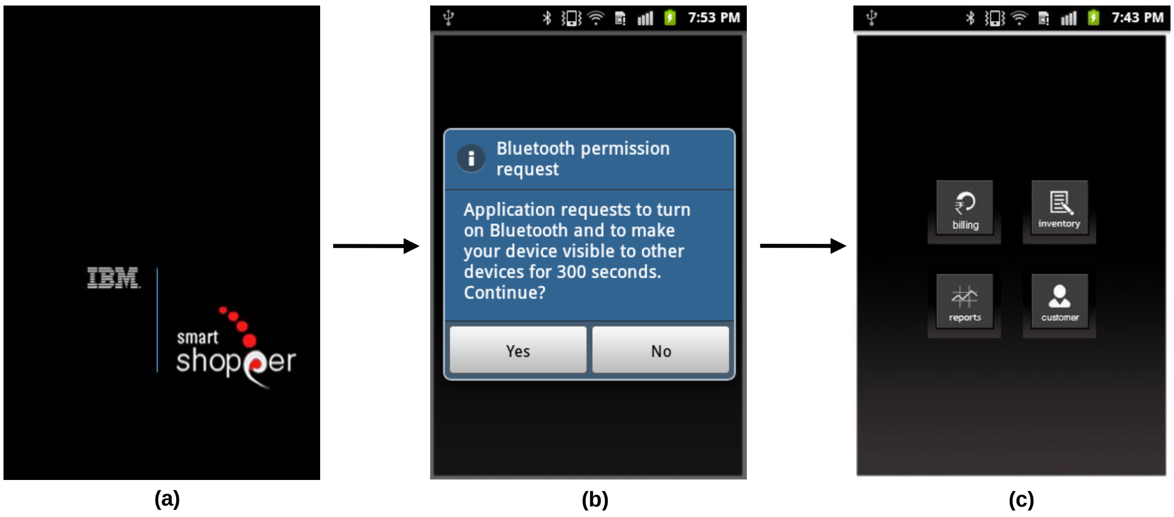
6. Application Suite Implementation

We implemented prototype Android applications for the shopper and the retailer. To run and test the apps, we used four phones running Android Linux v2.3 and communicating through Bluetooth: (i) Samsung Galaxy Ace S5830, (ii) Samsung Galaxy Y S5360, (iii) Samsung Galaxy Ace Plus S7500, and (iv) HTC Desire HD. The apps were developed on Eclipse using Java and HTML5. Though our current set of apps run only on Android, we developed the shopper application mostly in HTML5 using the IBM Worklight Studio 5.0.5; this will enable us to port the app easily to other mobile platforms (e.g., iOS). The SQLiteDatabase API was used for persistent local storage and MySQL Server 5.6 hosted on Apache Tomcat Server 7 for remote storage and sync; a REST API was exported for data lookup and manipulation. To process payments, we emulated a financial entity with the Cyclos 3.7 payment platform[i], running on IBM WebSphere Process Server 7.0.

6.1. Shopping Scenario Implementation

Our implementation of the shopping protocol described in Section 5.3 is illustrated in Figures 5-10. We walk through an instance of the protocol involving retailer '*Kamal Grocery*', showing selected screenshots. The shopper and retailer apps were installed as *.apk* files on two of the Samsung phones running Android.

The screenshots in Figure 5 illustrate how the retailer app is started, and the list of options available to the human (clerk/storeowner) operating the device (*Billing*, *Inventory*, *Transaction Reports*, *Customer Records*). The product inventory is loaded from the local database at startup, and the advertising module (*Bluetooth* in this example) is started so shopper's devices in the vicinity can discover and associate with the retailer.



Application requests to turn on Bluetooth and to make your device visible to other devices for 300 seconds. Continue?

Figure 5. Mobile Shopping Scenario: Retailer Instantiation and Presence Advertisement

Next, the shopper app is started, and it displays the list of shopping lists created beforehand (Fig. 6a). The user walks towards the retailer's device. When he is within Bluetooth range of the latter, a notification appears asking the shopper if he wishes to associate with the retailer (Figs. 6b-d). The shopper opts to do so, and Fig. 6e indicates the association of the shopper with the store '*Kamal Grocery*'.

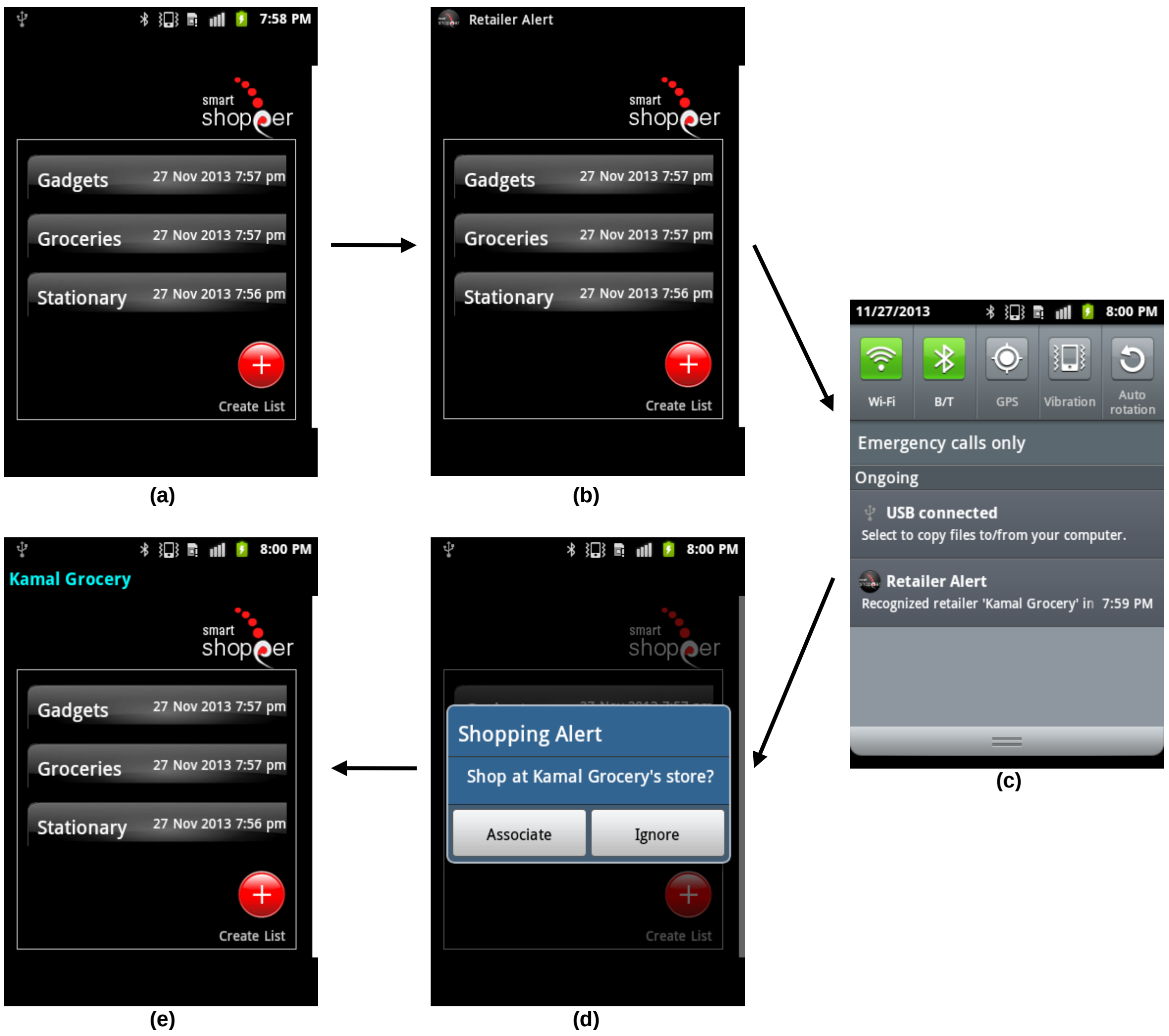
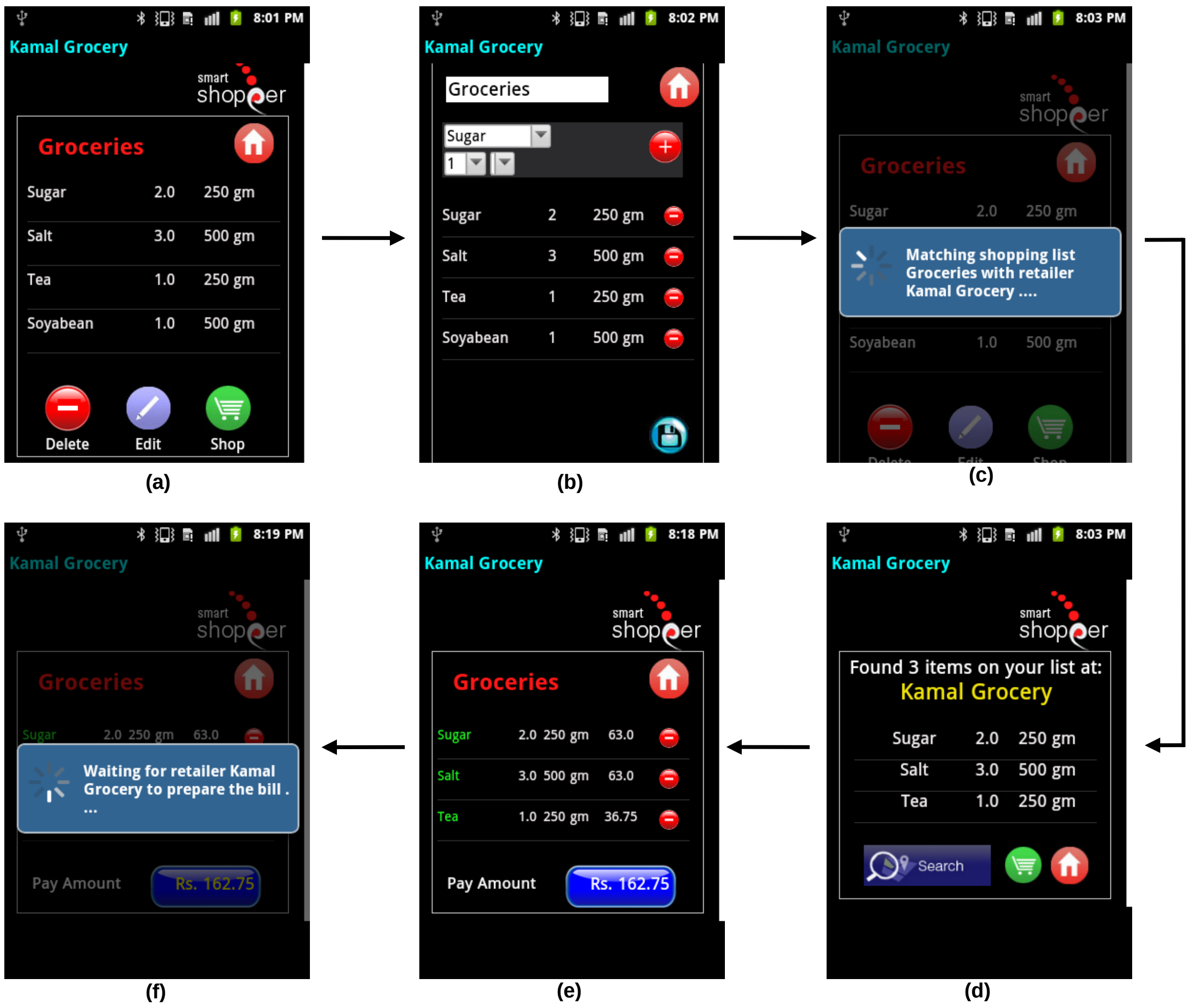


Figure 6. Mobile Shopping Scenario: Shopper's Lists and Store Discovery

Figure 7 illustrates the matching protocol, a process through which a customer's list is matched with the retailer's product inventory. The shopper selects the *Groceries* list, whose contents appear on the screen (Figs. 7a-b). He then sends the list to the retailer's device, and waits while the latter attempts a match with its inventory (Fig. 7c); the match involves no human/manual action. The list of available items is then returned to the shopper, who can view them and their costs on his device (Figs. 7d-e). He subsequently triggers a purchase request and waits for the retailer's device to respond (Fig. 7f).



(a)

Sugar	2.0	250 gm
Salt	3.0	500 gm
Tea	1.0	250 gm
Soyabean	1.0	500 gm

(b)

Sugar	2	250 gm
Salt	3	500 gm
Tea	1	250 gm
Soyabean	1	500 gm

(c)

Matching shopping list Groceries with retailer Kamal Grocery ....

(d)

Found 3 items on your list at: Kamal Grocery

Sugar	2.0	250 gm
Salt	3.0	500 gm
Tea	1.0	250 gm

(e)

Sugar	2.0	250 gm	63.0
Salt	3.0	500 gm	63.0
Tea	1.0	250 gm	36.75

Pay Amount Rs. 162.75

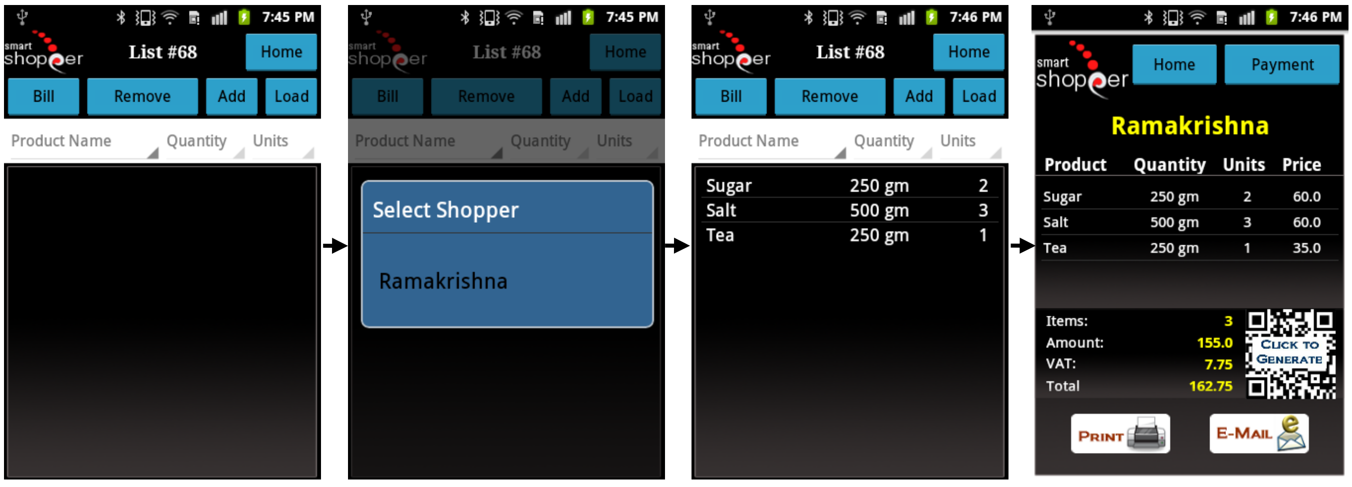
(f)

Waiting for retailer Kamal Grocery to prepare the bill . ...

Pay Amount Rs. 162.75

Figure 7. Mobile Shopping Scenario: Customer's Shopping List Match with Retailer's Inventory

The retailer's app now awaits human action to load and inspect the shopper's purchase request; an action performed by a small-store owner or large-store clerk. Figures 8a-c illustrate the loading of the selections made by the shopper (named *Ramakrishna*, who is known to have previously shopped with *Kamal Grocery* and thus has a record). The retailer then generates a bill (Fig. 8d).



Product Name	Quantity	Units
Sugar	250 gm	2
Salt	500 gm	3
Tea	250 gm	1

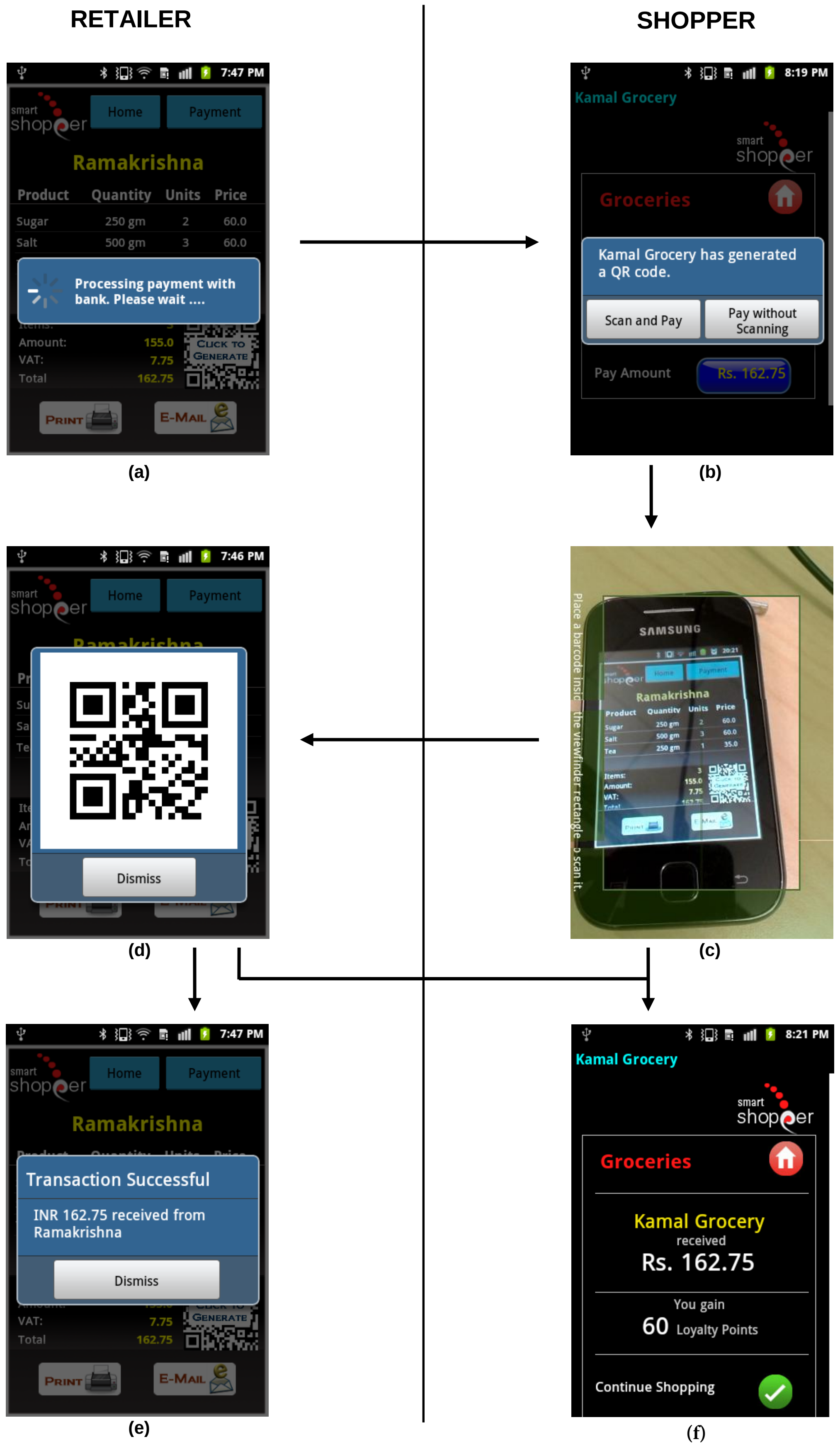
Ramakrishna

Product	Quantity	Units	Price
Sugar	250 gm	2	60.0
Salt	500 gm	3	60.0
Tea	250 gm	1	35.0

Items:	3
Amount:	155.0
VAT:	7.75
Total	162.75

Figure 8. Mobile Shopping Scenario: Retailer Processing Shopper's Bill at Checkout

The payment protocol is illustrated in Fig. 9. First, the retailer triggers a payment request (Fig. 9a). To make the protocol more secure, the retailer app generates a QR-code containing all the billing information, which the shopper's device may scan directly to verify and authorize the payment. Thus, the shopper's device subsequently displays an option to scan the QR-code (Fig. 9b). The shopper elects to do so; the shopper app starts the camera (Fig. 9c) and scans the QR-code generated on the retailer's device (Fig. 9d). A notification of successful purchase is displayed on the retailer's (Fig. 9e) device, and simultaneously an invoice appears on the shopper's (Fig. 9f) device. The actual payment is processed transparently through the Cyclos server behind the scenes.



7:47 PM

Ramakrishna

Product	Quantity	Units	Price
Sugar	250 gm	2	60.0
Salt	500 gm	3	60.0

Processing payment with bank. Please wait ....

Amount: 155.0
VAT: 7.75
Total 162.75

8:19 PM

Kamal Grocery

Groceries

Kamal Grocery has generated a QR code.

Pay Amount Rs. 162.75

7:46 PM

Transaction Successful

INR 162.75 received from Ramakrishna

8:21 PM

Kamal Grocery

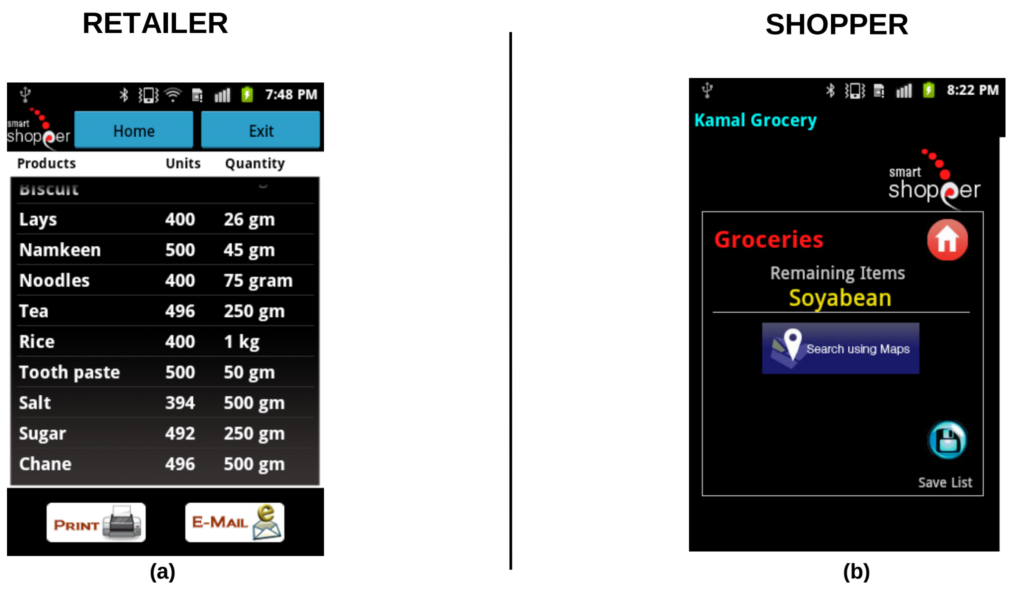
Groceries

Kamal Grocery received Rs. 162.75

You gain 60 Loyalty Points

Figure 9. Mobile Shopping: Payment Protocol

Finally, the updated states in the aftermath of the transaction are illustrated in Fig. 10. The retailer's inventory, purchase, and customer records are suitably updated; Fig. 10a shows a snapshot of the inventory. The shopper app displays the list of items that were unavailable at *Kamal Grocery*, and allows the shopper to suitably update his shopping list if he so chooses (Fig. 10b).



Products	Units	Quantity
Biscuit		
Lays	400	26 gm
Namkeen	500	45 gm
Noodles	400	75 gram
Tea	496	250 gm
Rice	400	1 kg
Tooth paste	500	50 gm
Salt	394	500 gm
Sugar	492	250 gm
Chane	496	500 gm

Figure 10. Mobile Shopping Scenario: Retailer Inventory and Shopping List Changes

6.2. Variations

A generic retail store was represented in the above scenario, to which we can add variations and extra features. If a shopper cannot find all the items he needs in a store, a search could be triggered for another that does contain those items, and a map of stores in the vicinity displayed. The only extra implementation this would involve is launching the maps app resident on the phone. Alternatively, the store itself may suggest other options to maintain customer loyalty; the only extra implementation this will involve is exposing new web service APIs for retailers to communicate with each other. As an extension, the shopper-retailer protocol could be augmented so that retailers may suggest alternative products or advertise new products to shoppers.

7. Evaluation and Discussion

As we have seen in sections 5 and 6, our shopper and retailer apps are designed to be location-independent, and reliant only on mobile devices and communication technology. Communication can be short range peer-to-peer, or conducted over ubiquitously available cellular and WiFi networks. These apps are not tied to each other: using the shopper app on a mobile device, one can shop with any retailer who runs a compatible app on his mobile device. The reverse is also true: a retailer app can interact with and conduct transactions with any potential customer who has downloaded and installed the app on his personal mobile device. To examine whether our apps are ready for widespread use, we need to evaluate them on a number of dimensions: *usability*, *security* and *privacy*, and *scalability*.

7.1. Usability

Ease of use and low interaction time are critical factors from an end-user's perspective for the shopper and the retailer app. Table 2 shows the number of clicks and average time that it takes a typical user to perform certain tasks. As seen in the table, most tasks can be performed in less than a minute, with very few clicks.

The combination of ease of use and low transaction times ensure that a low-skilled retailer can manage his store's PoS terminal, thereby reducing a shopper's waiting time and improving the shopping experience. If PoS can be implemented on a mobile device, a large store can afford to have more checkout terminals. At the same time, low-end stores such as those shown in Figure 1b can feasibly use the app. In addition, large consumer products companies have informally indicated to us that they would be interested in using the sales data recorded at such stores to improve their sales and delivery. With this data, they can perform analytics that will help them understand the effect on sales for particular offers better and to understand gaps and rectify them faster.

From a larger perspective, our application suite provides the minimal functionality one needs to run a store. A shopper too has the minimal support and information he needs to purchase items without forgetting some or being unable to pay for lack of funds. Devices can use whatever communication technology is available. If WiFi or cell networks are not present, Bluetooth could be used; in our test runs, we discovered that shopper's devices reliably discovered retailers' devices close by. These apps can be ported to non-retail business scenarios easily in a short development cycle; only the information semantics and the application logic to process it must change.

Table 2. Number of clicks and average time taken for each task

Task	# Clicks	Average Time (seconds)
Shopper completes a transaction, given the list	5	10
Retailer creates bill from the shopper's list	2	4
Retailer creates fresh bill (5 items)	22	30
Retailer checks inventory	1	2
Retailer and shopper apps connect to each other	2	5
Retailer looks at shopping summary for a specific day	2	10

7.2. Security and Privacy

The goal of our project was to provide a broad and extensible platform. Therefore we did not implement the entire gamut of features one could think of when realizing our retail scenario. Security and privacy features fall under this category. The shopper-retailer transactions could be secured by encrypting communication, providing more security for the retailer's data store, and supporting secure discovery with privacy preservation. We can leverage existing research in these areas or invent new solutions, but demonstrating these features was beyond the scope of this paper. Our apps are designed to be easily extensible, and we will augment them with security and privacy features in future work.

7.3. Scalability

There is nothing in our platform that limits its scalability. The retailer app can simultaneously advertise to, and conduct transactions with, multiple customers. Shoppers can, should they choose to, simultaneously browse the wares of multiple retailers. Our design features do not limit the transactional load, defined by the number of shoppers interacting with a retailer, or the number of retailers interacting with a shopper. It depends entirely on the amount of resources possessed by the retailer's back-end and by the customer's phone. Therefore, we can justifiably claim that our platform is as scalable as the state-of-the-art in computing technology.

8. Conclusion and Future Work

The high cost of technology is a barrier to entry in retail, especially in emerging markets. As a result, most small scale businesses suffer from inefficiency and unreliability, as was evident from a user study we conducted in two cities in North India. In this paper, we demonstrated how we could eliminate the entry barrier, and make shopping easier and more reliable, by building universal shopper and retailer apps that perform a minimal set of required functions. The only hardware requirement is a smart phone or tablet, which is ubiquitous even in non-prosperous societies. Retail stores can be truly un-tethered from geographical location, and can access back-end systems (if required) through widely available cellular or WiFi Internet connections. Our apps can be extended to support a wide array of features, and can also be adapted for other transactional scenarios with little investment of a developer's time. A natural next step is to launch these apps in a variety of settings in India, both in small stores with the assistance of telecom operators, and in larger outlets with the help of commercial retailers. Data collected from these deployments will provide new insight into usage patterns. For interoperability, we must configure the apps to support universally adopted information representation mechanisms. This will enable new shopper apps to work with legacy store inventory systems, and new retailer apps to work with existing payment apps like Square.

References

[1] Hughes, N. and Lonie, S. (2007) M-PESA: Mobile money for the "unbanked" turning cellphones into 24-Hour tellers in Kenya. *Innovations* 2(1-2): 63–81.

[2] Medhi, I., Gautama, S.N. and Toyama, K. (2009) A comparison of mobile money-transfer UIs for non-literate and semiliterate users. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems (ACM): 1741–1750. doi:10.1145/1518701.1518970.

[3] MIREMBE, D.P., KIZITO, J., TUHEIRWE, D. and MUYINGI, H.N. (2008) A model for electronic money transfer for low resourced environments: M-cash. In *Proceedings of the Third International Conference on Broadband Communications, Information Technology & Biomedical Applications, BROADCOM '08* (Washington, DC, USA: IEEE Computer Society): 389–393. doi:10.1109/BROADCOM.2008.37.

[4] Google Wallet, http://www.google.com/wallet.

[5] Square Inc. (US), https://squareup.com.

[6] Boku, http://www.boku.com/.

[7] MESFIN, W.F. (2012) Mobile information systems architecture for everyday money practice. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems* (ACM): 205–212. doi:10.1145/2457276.2457316.

[8] Airtel Money, http://airtelmoney.in.

[9] WAGNER, M. (2009) Keys, money, and mobile phone. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications (ESWC)*, Heraklion, Greece, 31 May–4 June, 2009 (Berlin, Heidelberg: Springer-Verlag).

[10] HASSINEN, M., HYPPÖNEN, K. and HAATAJA, K. (2006) An open, PKI-based mobile payment system. In *Emerging Trends in Information and Communication Security* (Springer), 86–100.

[11] MONTEIRO, D.M., RODRIGUES, J.J. and LLORET, J. (2012) A secure NFC application for credit transfer among mobile phones. In *Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on* (IEEE): 1–5.

[12] PRADHAN, S., LAWRENCE, E. and ZMIJEWSKA, A. (2005) Bluetooth as an enabling technology in mobile transactions. In *International Conference on Information Technology: Coding and Computing* (IEEE), 2: 53–58.

[13] MASSOTH, M. and PAULUS, D. (2008) Mobile acquisition of sales operations based on a blackberry infrastructure with connection to an inventory and ERP management system. In *Mobile Ubiquitous Computing, Systems, Services and Technologies* (IEEE): 413–418.

[14] BALAN, R.K., RAMASUBBU, N., PRAKOBPHOL, K., CHRISTIN, N. and HONG, J. (2009) mFerio: the design and evaluation of a peer-to-peer mobile payment system. In *Proceedings of the 7th international conference on Mobile systems, applications, and services* (ACM): 291–304.

[15] ZDRAVKOVIC, A. (1998) Wireless point of sale terminal for credit and debit payment systems. In *Electrical and Computer Engineering, 1998. IEEE Canadian Conference on* (IEEE), 2: 890–893.

[i] Cyclos—Online & mobile banking software: http://www.cyclos.org.