An Un-tethered Mobile Shopping Experience

Venkatraman Ramakrishna[1(✉)], Saurabh Srivastava[1], Jerome White[1], Nitendra Rajput[1], Kundan Shrivastava[1], Sourav Bhattacharya[1], and Yetesh Chaudhary[2]

[1] IBM India Research Laboratory, New Delhi 110070, India
{vramakr2, saurabhsrivastava, jerome.white, rnitendra, kushriva, sav.accharya}@in.ibm.com
[2] Indian Institute of Technology, Kharagpur 721302, West Bengal, India
yetesh.ch@cse.iitkgp.ernet.in

Abstract. Smart phones with access to apps from online stores are ideal candidates to replace expensive hardware like PoS 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 a user study of retailers in North India to understand the ecosystem in emerging markets, and ascertain their needs, helping us build a useful platform for mobile shopping.

Keywords: Mobile shopping · Mobile payment · Emerging market · User study

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.

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. In emerging markets, research and data on the retail ecosystem, the needs of retailers, and the mechanisms to promote growth, is lacking. We conducted a user study in North India, an emerging market, to obtain

I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 409–421, 2014.
DOI: 10.1007/978-3-319-11569-6_32

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 the 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, 4–6, 8], which enable customers to make purchases using a handful of clicks. Yet such systems are not going to be popular, even in developed countries, unless compatible systems are used by retailers. 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 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 how we designed and implemented a suite of universal retailer management and shopping apps that requires 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 can be used by shopkeepers and customers in a variety of settings (urban, semi-urban, rural, etc.) with intermittent to no connectivity. We also show how they can be operated easily by customers, shopkeepers and retail store clerks with no training. Scenarios are described in Sect. 2, and a related work survey in Sect. 3. The results of our user study to ascertain the needs of retailers in emerging markets are presented in Sect. 4. The design of our platform and application suite to realize such scenarios is described in Sect. 5, with implementation details in Sect. 6. We evaluate our platform in Sect. 7, and make concluding remarks in Sect. 8.

2 Motivating Retail Shopping Scenarios

We present below two representative retail scenarios, and discuss other examples.

2.1 At the Corner Shop

Peter needs to purchase grocery items that include milk, cereal, and spices. He walks toward the small grocery store in the neighborhood 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 needed 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 aren't. If the store possessed mobile 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. Other businesses could also benefit by automating provider-consumer interactions. The restaurant business is an example, where customers would love to be aware of all available choices in an ad hoc manner, and where restaurants would love to establish long-lived relationships with their patrons. Book stores are another example, where one often has to visit multiple stores to purchase all the required books. A store that proactively orders books, or places back-orders, for their faithful customers will see its business flourish. In the following sections, we show exactly what storeowners and shoppers need through field surveys, and design a universal set of shopper and retailer apps to realize the scenarios described in this section.

3 Related Work

Retail technology has been mostly driven by commerce, with little academic research.

3.1 Commercial Services

Square Register [5] is a specialized card-reader that allows merchants to accept card transactions via their mobile devices. Requiring such specialized hardware, though, makes this solution less likely to be adopted in small shops, especially in emerging economies. There are also several wallet services that allow users to pay for goods via their mobile phone; e.g., Google Wallet [4], Square Wallet [5], and Mobile Pay USA [6]. The merchant and the customer both have accounts with the respective service. When a transaction is made, the underlying service handles monetary exchange using pre-existing and verified traditional-bank details. Airtel Money [8] allows customers to pay bills, make purchases, and transfer money via their mobile device using USSD. Users setup an account with Airtel that is debited each time they make a transaction through their Airtel SIM card. Like Airtel Money, M-PESA [1] is another SMS- and USSD-based popular mobile money solution. Being a branchless banking service—users register and transact physical funds through a network of agents—M-PESA is deployed widely in the developing world. These mobile money solutions managed by telecom providers help ease the purchase process, but do not provide complete end-to-end shopping solutions like we do. We provide an extensible suite of mechanisms for shoppers and customers, of which seamless payment methods are but one aspect.

3.2 User Studies

A study by Medhi, Gautama, and Toyama compared mobile money user interfaces amongst low-literate users [2]. The authors studied 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 and concluded that speech- and picture-based interfaces, as opposed to text-based UIs, were a more viable option for low-literate subscribers. A study of money practices in rural Ethiopia [7] revealed that existing mobile money research is biased toward technical contributions; ideally it should also consider users' monetary practices. Our ethnography-based implementation work is in line with this approach.

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. Hassinen et al. present a mechanism for mobile payment with real or virtual PoS systems [10]. While it does not focus on the PoS being on the mobile itself, it provides rich communication options, ranging from SMS to Bluetooth to IP based data transfer. Other solutions exist that utilize NFC and Bluetooth for money transfer [11, 12], but these are limited to the payment mechanism and do not support the entire shopping experience that has inventory, cash register, loyalty points, etc. Zdravkovic discusses a wireless PoS terminal [15] that does not incorporate inventory and billing features. Massoth and Paulus outline an inventory management system where the Blackberry platform is used to communicate latest sales information to a centralized server, but they do not support interaction with a customer's device for billing [13]. The usability aspects of mFerio [14], which facilitates mobile payments through NFC and physical touch, were studied in a controlled lab environment using Singapore undergraduates. It is hard to say if their solution would be suitable for a low literate population.

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, 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 business? To help understand these questions we carried out a Contextual Enquiry: we interviewed each participant separately for an hour; data from the studies was collected and analyzed.

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 (Fig. 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

Fig. 1. (a) Supplier giving inventory list to low-literate shopkeepers. (b) The grocery stores

commodity if he could not sell it. Some study participants also reported that they maintained notebooks to keep a record 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 a lack of information pertaining to new offers on products and workforce management. Some also desired 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 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 virtually all smart phone system service providers support, 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 serve all the transaction functions its user requires (Fig. 2). In effect, the device acts as a location engine, a search engine, and a mobile wallet all rolled into one.

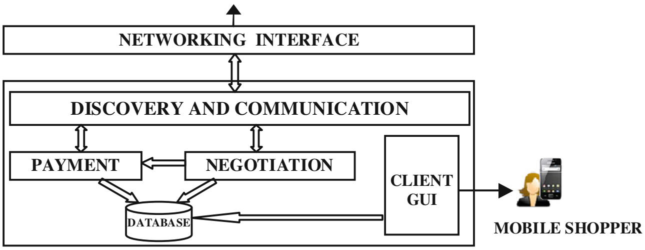

Fig. 2. Shopper application architecture

The *discovery and communication* module locates retail stores that are reachable from the shopper's device, identifying stores that 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.

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.

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 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 Fig. 3 below.

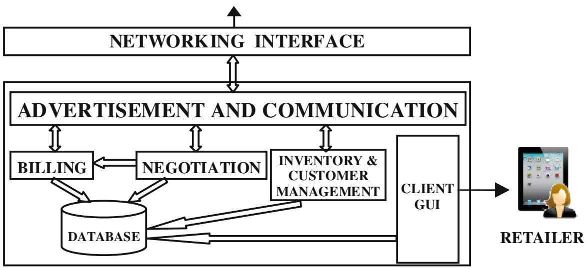

Fig. 3. Retailer application architecture

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 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 integrity.

The *management* module performs two functions. Through *inventory management*, the retailer can manage the products currently being sold, and set the positions (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 to the shopper with associated pricing and reward points information. For a first time association, it automatically adds 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 takes part in a secure financial transaction with the shopper, the money being transferred to a pre-configured financial account. An invoice is sent to the shopper's app.

5.3 Shopping Protocol

The transaction protocol between the shopper and retailer proceeds as follows:

1. ***Shopper***: discover a retailer's device and associate with it.
2. ***Shopper***: create or edit a shopping list of product items.
3. ***Shopper***: send list to retailer, requesting a match with its product inventory.
4. ***Retailer***: perform match and return list of available items with prices to shopper.
5. ***Shopper***: select items to purchase from the available products; if no items are selected, terminate the protocol; otherwise, send selections to retailer.
6. ***Retailer***: look up shopper's selections, generate bill, and send it to shopper; simultaneously initiate payment request with financial back-end.
7. ***Shopper***: examine bill and authorize/deny payment with financial back-end.
8. ***Shopper*** and ***Retailer***: receive notification of successful payment, or failure.
9. ***Retailer***: if payment was successful, send invoice to shopper with explicit mention of reward points or discount offers.

This is designed to be a minimal and generic protocol for transactions in our target scenarios. While augmentations could be made—more security features, for example—such discussion at this level of detail is beyond the scope of this paper.

We also do not allow the billing process to be completely automated, as we see the retailer's app waiting for a user in Step 6 above. 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 both sets of 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 (http://www.cyclos.org/), running on IBM WebSphere Process Server 7.0.

6.1 Shopping Scenario Implementation

We walk through an instance of the shopping protocol (Sect. 5.3) involving retail '*Kamal Grocery*', showing selected screenshots. The shopper and retailer apps were installed as.*apk* files on two of the Android phones. The retailer app is started, and the product inventory loaded from the local database. Now the shopper app is started, and it displays the list of shopping lists created beforehand. The user walks towards the retailer's device. When he is within Bluetooth range, a notification appears asking the

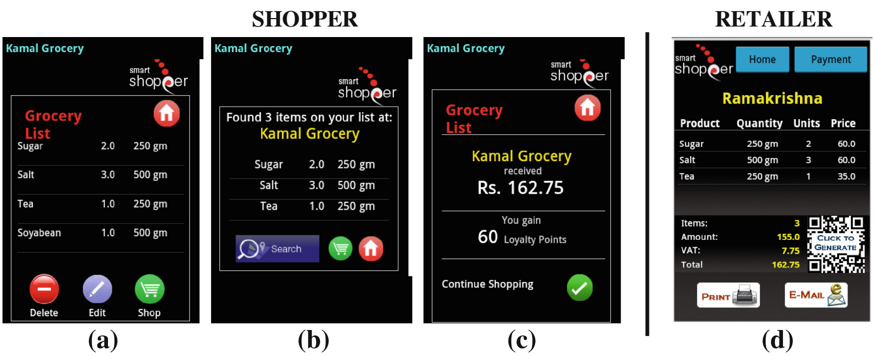

Fig. 4. Mobile shopping: querying and transaction protocol

shopper if he wishes to associate with the retailer. The shopper opts to do so and selects a list, whose contents appear on the screen (Fig. 4a). He then requests a match with the retailer's inventory, and sees the result on the screen following a silent protocol (Fig. 4b). He subsequently triggers a purchase and waits for the retailer's device to respond. The retailer loads the shopper's list and requests a payment (Fig. 4d). The payment is processed transparently through the Cyclos server, with an optional step where the shopper scans a retailer-generated QR-code of the bill, and an invoice appears on the shopper's device indicating a successful transaction (Fig. 4c). The retailer's inventory, purchase, and customer records are suitably updated.

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

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 the customer waiting time and improve the customer 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 the one shown in Fig. 1 can feasibly use the app. In addition, large consumer products companies have informally indicated to us that they would be

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

Task	# Clicks	Average time (s)
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

interested in using the sales data recorded at such stores to improve their sales and delivery by performing analytics that result in understanding the effect on sales for particular offers better or 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.

8 Conclusion

The high cost of technology is a barrier to entry in retail, especially in emerging markets. As a result, most of the small scale businesses suffer from inefficiency and unreliability, as was evident from a user study we conducted in 2 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. Our apps can be adapted for other businesses with little investment of a developer's time. We plan to launch our 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 give us new insights into usage patterns. We will also conduct research in universal 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., Lonie, S.: M-PESA: mobile money for the "Unbanked" turning cellphones into 24-hour tellers in Kenya. Innov. Technol. Gov. Globalization **2**(1–2), 63–81 (2007). MIT Press
2. Medhi, I., Gautama, S.N., Toyama, K.: A comparison of mobile money-transfer UIs for non-literate and semi-literate users. In: CHI 2009, Boston, MA, pp. 1741–1750 (2009)
3. Mirembe, D.P., Kizito, J., Tuheirwe, D., Muyingi, H.N.: A model for electronic money transfer for low resourced environments: M-cash. In: BROADCOM 2008, Pretoria, South Africa, pp. 389–393 (2008)
4. Google Wallet. http://www.google.com/wallet
5. Square Inc. (US). https://squareup.com
6. Mobile Pay USA. http://www.mobilepayusa.com
7. Mesfin, W.F.: Mobile information systems architecture for everyday money practice. In: MEDES 2012, Addis Ababa, Ethiopia, pp. 205–212 (2012)

8. Airtel Money. http://airtelmoney.in
9. Wagner, M.: Keys, money and mobile phone. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, p. 4. Springer, Heidelberg (2009)
10. Hassinen, M., Hyppönen, K., Haataja, K.: An open, PKI-based mobile payment system. In: Müller, G. (ed.) ETRICS 2006. LNCS, vol. 3995, pp. 86–100. Springer, Heidelberg (2006)
11. Monteiro, D.M., Rodrigues, J.J.P.C., Lloret, J.: A secure NFC application for credit transfer among mobile phones. In: CITS 2012, Amman, Jordan, pp. 1–5 (2012)
12. Pradhan, S., Lawrence, E., Zmijewska, A.: Bluetooth as an enabling technology in mobile transactions. In: ITCC 2005, Las Vegas, NV, USA, vol. 2, pp. 53–58 (2005)
13. Massoth, M., Paulus, D.: Mobile acquisition of sales operations based on a blackberry infrastructure with connection to an inventory and ERP management system. In: UBICOMM 2008, Valencia, Spain, pp. 413–418 (2008)
14. Balan, R.K., Ramasubbu, N., Prakobphol, K., Christin, N., Hong, J.: mFerio: the design and evaluation of a peer-to-peer mobile payment system. In: MobiSys 2009, Krakow, Poland, pp. 291–304 (2009)
15. Zdravkovic, A.: Wireless point of sale terminal for credit and debit payment systems. In: IEEE Canadian Conference on Electrical and Computer Engineering 1998, Waterloo, CA, pp. 890–893 (1998)