SmartRestaurant – Mobile Payments in Context-Aware Environment

Janne Lukkari
MediaTeam Oulu
University of Oulu, Finland
janne.lukkari@ee.oulu.fi

Jani Korhonen
MediaTeam Oulu
University of Oulu, Finland
jani.korhonen@ee.oulu.fi

Timo Ojala
MediaTeam Oulu
University of Oulu, Finland
timo.ojala@ee.oulu.fi

ABSTRACT
Mobility, context-awareness and payment combined provide a customer with a completely new setting of consuming services at any time and any place. We introduce SmartRestaurant service, which allows customers to use mobile devices for ordering and paying lunches from a nearby campus restaurant beforehand. Further, SmartRestaurant provides the restaurant with means of adjusting the sales with production capacity and prior knowledge of upcoming orders. We present a user evaluation of the system in form of a field trial in the real environment of use.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: Wireless communication;

K.4.4 [**Electronic Commerce**]

General Terms
Experimentation

Keywords
Context-aware, Mobile payment, Indirect mCommerce, B2C

1. INTRODUCTION
In recent times, there has been dramatic penetration of powerful mobile devices with ability to enhance the mobile environment. A number of services have been deployed for mobile terminals including news, directory services and payment services.

If we are paying or selling goods through a fixed or a wireless network, we can consider that as eCommerce (electronic commerce) or mCommerce (mobile commerce). ECommerce like mCommerce can be classified to indirect and direct (tangible and intangible) business [3], depending on type of the goods that are being sold via the fixed or wireless network.

Mobile services are available at any time and any place, which has made the mobile phone technology so popular. Usually the chargeable added value mobile services provide ringing tones, icons and other intangible digital goods, which are being downloaded to the customer's device. Shopping via mobile devices is projected to be a major business channel in the coming years.

Mobile payment (and also mobile payment systems) can be classified to remote and locale payments. The difference is obvious: in remote payments mobile network is used to route the transaction information between the payment system and user device as in local payments the user connects to the POS (Point of Sale) terminal physically or with a short range access protocol like Bluetooth or infrared. Both remote and locale payments systems are being used in various applications.

The remote payment systems are usually used in services that do not require handling of physical goods. In these content services the user's location is irrelevant. For example Sonera, the leading operator in Finland, provides several mobile services for its subscribers. Users can order different kind of SMS (Short Message Service) newsletters and digital content to their mobile phones [13].

In 2002 Sonera launched a mobile payment service Sonera Shopper [14]. It is an online mobile phone service, which is being used with SMS-messages. The mobile user needs to register to the service before the use. With the Sonera Shopper mobile payment method the users can pay their purchases in several stores in Helsinki and also parking fees in couple of cities in Finland. The payments are being made using the user's Sonera Shopper account or the user's credit card.

ZonePay [19] develops mobile ordering systems and payment solutions. For the time being their major product is the iWaitLess service. The service is directed to 'take away' restaurants and it can be used via mobile device or a web-browser. The customers can order and pay servings before hand from the restaurants, which are signed-up to the service. The service has been tested and deployed in the United States.

The local payment systems are used in user's proximity environment. The distance between the POS and the user depends on the technology that is used. Nokia and 2Scoot have tested RFID technology as a local payment system [9].

In mobile domain the service system can benefit from understanding the usage context of the users by automatically adapt the service to fit users' needs. The automated assistance is especially important with limited user interface devices like mobile phones. Generally, the context information can be understood as any information that can be used to describe the

situation of a person, place or object that has significance to the interaction between user and the service [2]. Useful context information exists in various levels of abstraction. Some examples of contexts are the location of the user, surrounding weather, user's current yearn, social relations with nearby users, bandwidth of the user's mobile device, screen size of the mobile device, etc. This information is eventually used to provide the user more valuable services like personalized news, guidance services, context-based directory services and other "smart" services.

In the previous mobile payment concepts context-awareness is not needed, since the user's location and contextual information do not provide any added value for the service. VISA presents a scenario of a traveling business executive who decides to get something to eat [18]. The person orders and pays a meal from a local restaurant using his wireless handheld computer. Using a carrier that calculates the customer's proximity to the restaurant and provides directions to the restaurant the customer gets his meal eventually. The paper does not present any actual implementation for realizing the scenario, but it just gives an idea how context-awareness could be used to provision more valuable services. Other related work on context-aware mobile commerce includes the context-aware and location-based mobile e-commerce server by Jin and Miyazawa [7] and Varshney's conceptual study of location management in mobile commerce applications [17]. Both papers remain somewhat theoretical, however, since no concrete deployment and its evaluation are reported.

The novel contribution of this paper is the deployment and evaluation of the SmartRestaurant service in the true environment of use with genuine end users. SmartRestaurant is designed for a context-aware mobile service system such as SmartRotuaari [11], allowing customers to use mobile devices for ordering and paying lunches from a nearby restaurant beforehand. Further, SmartRestaurant provides the restaurant with means of adjusting the sales with production capacity and prior knowledge of upcoming orders. An empirical evaluation is carried out in the real environment of use in a form of field trial involving one restaurant and 20 test users. The SmartRestaurant service is a new feature in the SmartRotuaari service system, which offers context aware mobile multimedia services to consumers at a city center [11].

One of the interesting features in the SmartRestaurant is the ability to entice the user to make the lunch order and payment from a distance to the restaurant. This behavior could be further stimulated by providing the user with the means and information needed for the purchase in a simple and intuitive way. This can be achieved with personalized mobile advertisements pushed to the user's mobile device. Kaasinen [8] has found out that the users' attitude towards push type mobile advertising is positive if the information is relevant. An implementation and empirical evaluation of a location-aware mobile advertisement system based on WAP Push is presented by Aalto *et al.* [1].

Context-awareness of the presented deployment of SmartRestaurant is limited to the implicit incorporation of the end-user's context when (s)he is placing the order. However, context-awareness could be enhanced in a straightforward manner by integrating SmartRestaurant seamlessly into a context-aware architecture such as SmartRotuaari [11], or by incorporating push type notifications using the aforementioned location-aware mobile advertisement system [1].

This paper is organized as follows. The SmartRestaurant system is introduced in Section 2. An empirical evaluation of the system is described in Section 3. Section 4 provides concludes the paper with a discussion on various aspects of the system and future work.

2. SMARTRESTAURANT SERVICE

SmartRestaurant is a web service for mobile users that has been designed to enhance a restaurant's production and delivery process. The SmartRestaurant actors are categorized to customers and employees.

The customers (also referred to as end-users) are normal customers except they use the SmartRestaurant to order and pay their lunch beforehand. They can browse the SmartRestaurant's menu with a mobile device, order and pay one or more meals, and schedule the delivery time of their order relative to their current context (time, location) so that the food will be hot and fresh when they enter the restaurant.

The employees of the restaurant configure the service and prepare the ordered meals. SmartRestaurant allows the restaurant to automatically adjust the sales with the production capacity. SmartRestaurant also provides the restaurant with a prior knowledge of upcoming orders.

2.1 Distributed architecture

SmartRestaurant's distributed architecture is illustrated in Fig. 1. Although the illustration contains references to specific local resources utilized in the field trial, the architecture is built of standard Internet components, which allows a robust deployment.

1. SmartRestaurant web service is installed on a host (named rotuaari.net in our case), which is connected to the Internet via a firewall.

2. The payment service is installed on a host, which is connected to the Internet via a firewall. In our system we use the e-maksu payment service [4] hosted by PPO, a local operator. The e-maksu service provides customers with accounts on which money is first transferred from a regular bank account. The money on the customer's e-maksu account can then be used for paying for content and services of service providers having subscribed to the e-maksu payment service. The payments are aggregated to the providers' e-maksu accounts and then transferred to their bank accounts.

3. Mobile users can use the SmartRestaurant with smartphones, which provide means for Internet access, e.g. GPRS (General Packet Radio Service) connectivity, and a web browser capable of showing XHTML pages. In the field trial phones used the GPRS network of Octopus, which is a local development and testing environment for new, innovative mobile applications and services [10].

4. Mobile users can use the SmartRestaurant also with a PDA (Personal Digital Assistant), which features wireless connectivity and a XHTML capable web browser. In the field trial PDA's connected to the Internet via panOULU, which is a local public access network comprising of a number of WLAN hotspots [12]. PanOULU provides wireless Internet access to accounted users, which are authenticated by an access controller.

5. Restaurant employees operate the SmartRestaurant service with a laptop equipped with WLAN connectivity and a standard web

browser. Laptops with wireless connectivity can be placed freely at the restaurant premises and they also allow the employees be mobile.

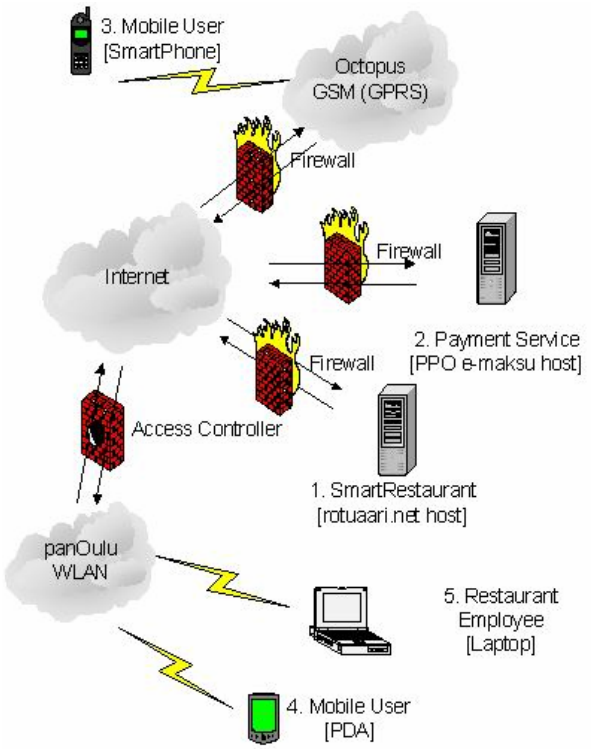


Figure 1. SmartRestaurant's distributed architecture.

2.2 Implementation

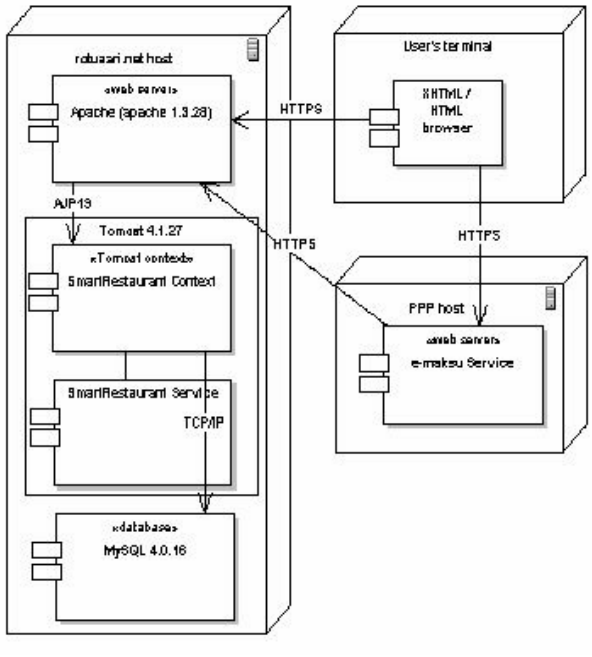
SmartRestaurant's run-time view is presented in Fig. 2(a). rotuaari.net is a multi-processor host that runs the SmartRestaurant service and other server components: Apache web server, Tomcat application server and MySQL database.

The SmartRestaurant Service component running on the Tomcat application server is implemented with JSP (Java Server Page) technology, which enables the service to serve dynamic web content. Before execution the JSP page is translated into a Java servlet, which processes the incoming HTTP requests and generates responses to the client [16].

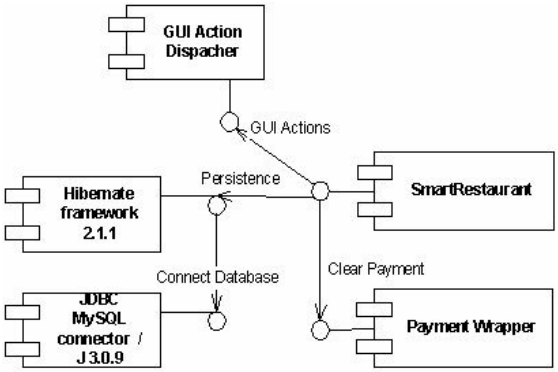
The e-maksu payment service is deployed on an IBM WAS (WebSphere Application Server), which runs on a multi-processor PPO host.

The GUI (Graphical User Interface) is realized as (X)HTML pages, hence a (X)HTML capable web browser is used at the customers' terminals.

All communication between the different entities is secure, as the HTTPS (HTTP over SSL (Secure Socket Layer)) protocol is used.



(a)



(b)

Figure 2. SmartRestaurant implementation: (a) run-time view of the system; (a) development view of the 'SmartResraurant Service' component.

The development view of the SmartRestaurant Service component is shown in Fig. 2(b). The component is designed according to the MVC (Model View Controller) design paradigm. MVC separates presentation from content and control, clarifies the roles and responsibilities of the system developer and the page designer [15], and makes the final application more compact and easier to maintain by preventing "spaghetti" code [16]. Utilization of MVC in the design of web applications has generated positive feedback [5].

The *SmartRestaurant* component is the *model*, including the business logic of the restaurant's production process. The *model* is mapped to the database with a Hibernate framework, which is a high performance object/relational persistence and query service

for Java [6]. Hibernate uses the JDBC (Java Database Connectivity) API to connect to the MySQL database. JDBC is a Java technology that provides connectivity to a wide range of SQL (Structured Query Language) databases.

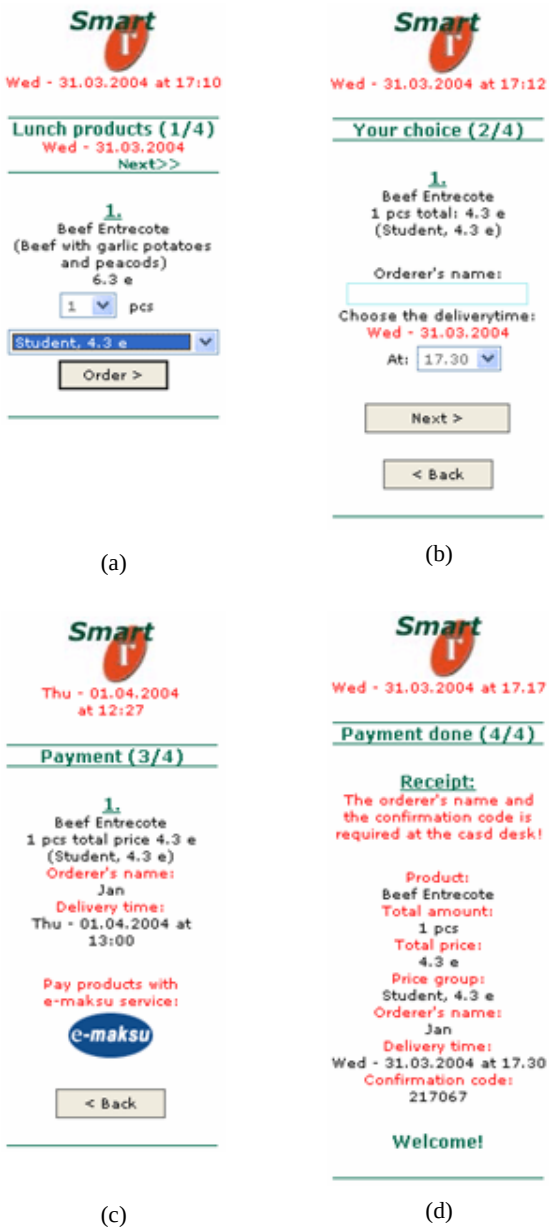


Figure 3. Screenshots of the customer interface.

The *GUI Action Dispatcher* acts as the *controller* and it updates the business logic by processing incoming requests.

The *Payment Wrapper* is an in-house solution that abstracts the payment transactions for the *SmartRestaurant* component, which generates the billing information and submits the information to e-maksu service. The *views* (i.e. user interfaces) are discussed in more detail in Sections 2.3 and 2.4.

2.3 Customer interface

The customer accesses the SmartRestaurant service simply by typing the URL (Uniform Resource Locator) of the service into the browser of the mobile device. The GUI of the service is realized with XHTML pages, which are illustrated in Fig. 3.

When a customer enters the SmartRestaurant service, (s)he is provided with a page showing the meals available that day (Fig. 3(a)). The customer can either advance to the next day, or select the type of the meal and the number of portions. Having done the latter the customer is provided with a new page, where (s)he enters her/his name and chooses the desired delivery time of the time slots still available in the restaurant's production process (Fig. 3(b)).

Next the customer clicks the e-maksu button (Fig. 3(c)) and is provided with a form, where (s)he enters her/his e-maksu user account name and password to authorize the payment. Finally, upon a successful payment the customer is provided with a receipt, which (s)he has to present at the restaurant to receive her/his order (Fig. 3(d)).

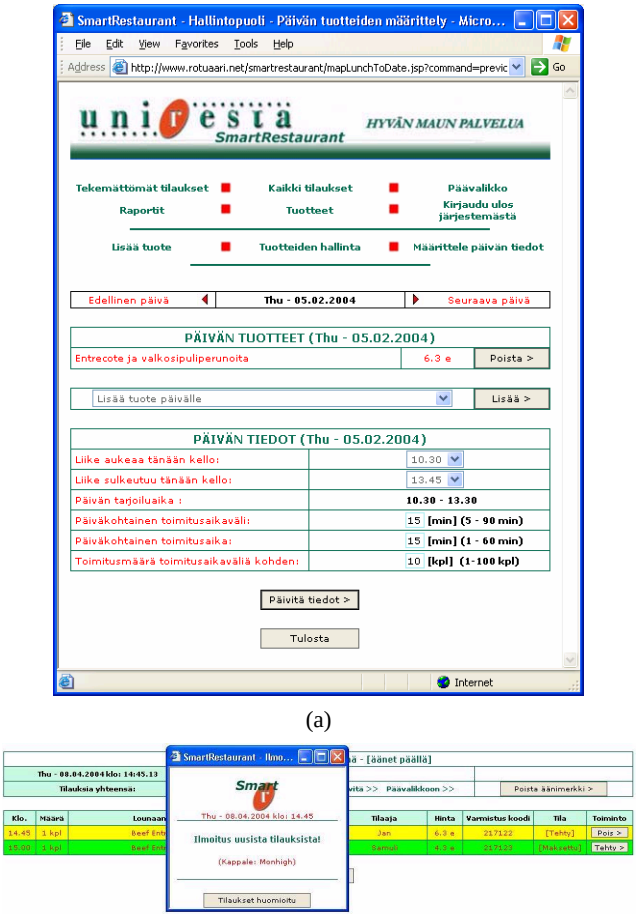


Figure 4. Screenshots of the restaurant employee interface.

2.4 Restaurant employee interface

The restaurant employees access the SmartRestaurant service via a web browser. After logging in with an employee account the employee is provided with the main menu, which provides access to following functions: configuration of the service, status page of orders (used by chef), listing of all orders (used by cashier) and reports (used by cashier).

Fig. 4(a) shows the configuration page, which allows for specifying the daily menu (meals and their prices) and the attributes of daily production process (opening hours, serving hours, delivery period, preparation time of a single order, and the maximum number of orders per delivery period). Based on these attributes, the SmartRestaurant automatically adjusts the sales with the production capacity, by keeping a record of booked delivery times and offering only available delivery times for new orders. In the screenshot the attributes are set so that at most 10 orders can be sold per a single delivery period of 15 minutes.

Fig. 4(b) shows the status page of orders, which is used by the chef in the kitchen. When a new order arrives, the chef is alerted with both an audible (randomly chosen from a set of audio clips, can be switched off) and a visible notification. A new entry displayed in green appears on the status page, showing detailed information of the order (delivery time, number of portions, type of meal). Having prepared the order the chef taps the "prepared" button on the corresponding row, and the entry turns into yellow.

3. EMPIRICAL EVALUATION

We evaluated the SmartRestaurant service in form of a field trial in the true environment of use involving one restaurant and 20 test users.

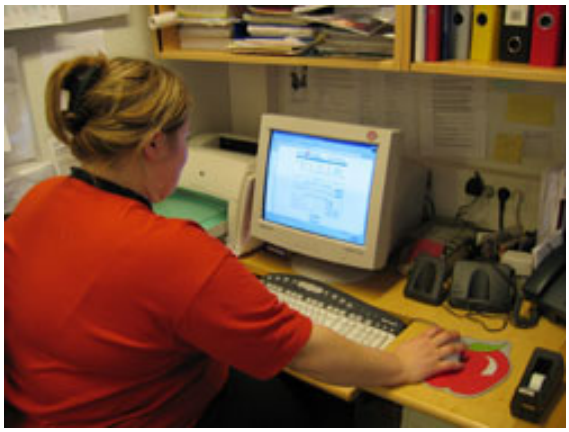
The operative goal of this field trial is to achieve results concerning end-users reactions with chargeable content due a mobile terminal in a context-aware environment, where the end-users evaluate their own time and location.

3.1 Setup of the field trial

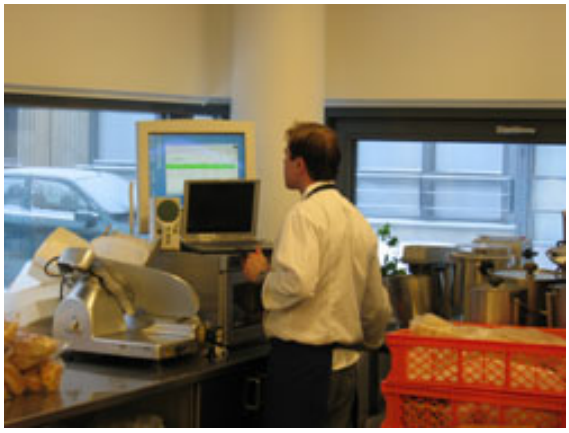
The field trial took place at the campus of the University of Oulu. SmartRestaurant service was offered by restaurant Kastari, which is one of the many campus restaurants operated by Uniresta Ltd. Kastari is a lunch restaurant offering two principal types of meals, a lunch buffet collected by the customers and a daily special lunch ordered and prepared individually on order-by-order basis. When the customer orders the special lunch, (s)he gets an order number and takes a seat in the lunchroom, waiting for her/his order to be prepared and served. Kastari is operated by three full-time employees, a chef and two cashiers working in turns in the kitchen and at the cash register.

Given the service model of Kastari, the SmartRestaurant service was configured to offer the daily special lunches. With SmartRestaurant customers were able to order and pay their special lunch beforehand, thus avoiding the need to queue up at the cash register. Further, customers could specify the preferred delivery time, avoiding the waiting in the lunchroom.

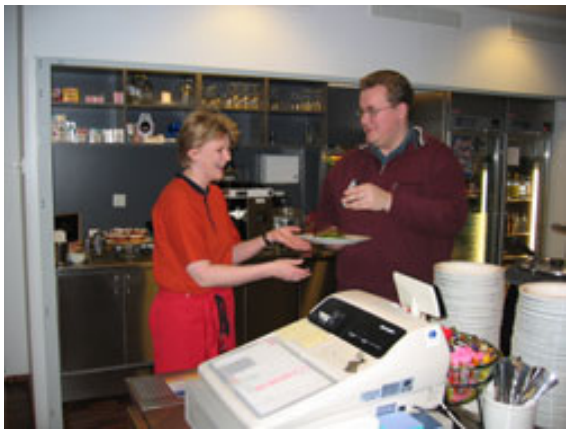
The restaurant employees operated the SmartRestaurant with three computers. A laptop equipped with WLAN connectivity and a large touch screen was placed in the kitchen for the chef. Another laptop with WLAN connectivity was placed next to the cash register for the cashier. The employees configured the service with a standard desktop PC available in the office, as shown in Fig. 5.



(a)



(b)



(c)

Figure 5. SmartRestaurant in action: (a) restaurant employee configures the service; (b) chef checks out new orders; (c) customer presents the receipt in his mobile phone to the cashier and gets his lunch.

The chef configured the SmartRestaurant service so that at most 10 lunches could be sold for a single delivery period of 15 minutes, first period of the day scheduled for 30 minutes before Kastari was opened. The maximum number of portions per order was set to five. The SmartRestaurant service was set to automatically close at the same time as Kastari was closed.

The restaurant employees were trained to operate the SmartRestaurant service and the new delivery process was rehearsed before the start of the field trial.

3.2 Test users

In total 20 test users participated in the field trial, 15 males and 5 females. All test users were Finnish, due to recruiting being carried out in Finnish. The majority of the test users were in their twenties, 8 (40 %) were 18-24 and 11 (55 %) 25-34 years old. There was only one person (5 %) in 35-49 age group. 11 (55 %) test users were students and the remaining nine worked in the University of Oulu.

18 test users reported to visit the Kastari restaurant regularly, at least once per week. One person (5 %) used to visit the restaurant only once per month and one had never visited the Kastari before the field trial. Only six persons (30 %) informed that they usually eat the special lunch at least once per week. Four persons had never tried the special lunch at Kastari.

All test users owned a mobile phone, while only four (20%) test users had their own PDA. Test users were frequent mobile phone users, for only seven people informed that they use SMS messaging only one or two times a week or less often. Other mobile services were used less often, as illustrated in Fig. 6.

18 test users reported to pay over 75 % of their bills with Internet banking. However, only two (10 %) of the test users had paid bills via a mobile phone.

Some of the test users did not consider downloading logos and ringing tones as mobile payment and were surprised to hear that downloading digital content to their mobile phone is classified as such. This reflects the transparency of the post-payment system employed by network operators in contrast to e.g. the e-maksu system utilized in SmartRestaurant, where the customer has to deposit money to her/his account beforehand.

Almost every test user (18 out of 20, 90 %) told to know the difference between GPRS, LAN and WLAN network technologies.

13 test users used the SmartRestaurant service with a smart phone equipped with a SIM card providing access to the Octopus GPRS network. The phones were different Nokia models: 3650 (5), 6600 (2), 7650 (4) and N-Gage (2). The Opera web browser was installed into the phones.

Seven test users used the SmartRestaurant service with a PDA (Compaq iPAQ running Pocket PC 2002) equipped with a cradle and a WLAN card. They were provided with a user account to the panOULU WLAN network available at the campus.

12 (60%) test users reported that they had not used the assigned mobile device before the trial. The mobile devices and network connections were free for the test users. The test users paid for the lunches with their own money.

All test users were given a quick guidance on the use of the mobile device and the SmartRestaurant service.

3.3 Results

The field trial was run for three weeks, from January 20th to February 6th, 2004. Data was collected via questionnaires, diaries, interviews and automatic logging of every GUI event at the server side. Test users were advised to fill a diary sheet every time they used or tried to use the service. The questionnaires and interviews were organized after the field trial. The employees of the Kastari restaurant and the Uniresta CEO filled also a staff questionnaire.

Table 1. Successful and failed lunch orders per device type.

Device	ok	%	failed	%
SmartPhone	40	47,06	5	38,46
iPAQ	33	38,82	7	53,85
PC	12	14,12	0	0,00
NA	0	0,00	1	7,69
total:	85	100	13	100

Table 1 shows the statistics of successful and failed lunch orders for each device type. Although SmartRestaurant was supposed to be used by a mobile device, some test users also used a PC at

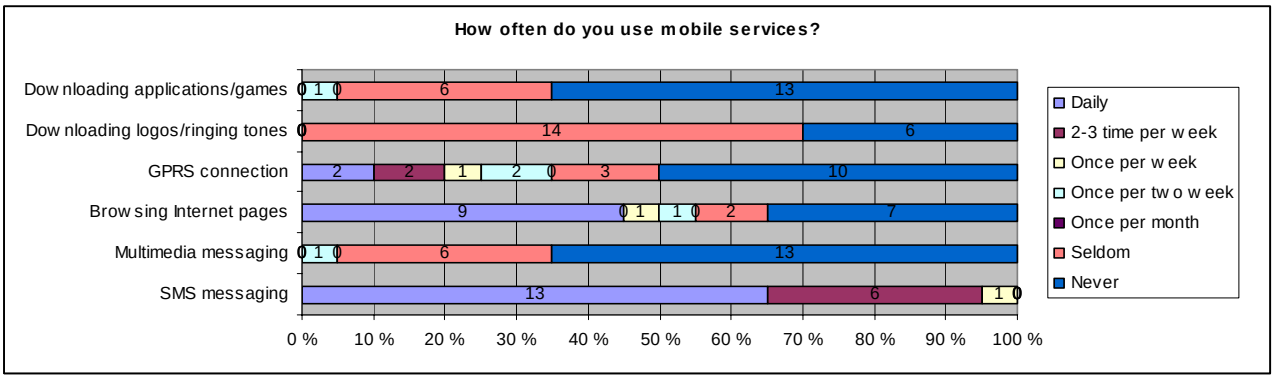


Figure 6. Test users' background in using particular mobile services.

times. In total 98 lunch orders were initiated, of which 85 succeeded, while 13 failed for various reasons (user entering an incorrect password and refusing to proceed, payment not cleared by the e-maksu payment service). In total 96 lunches were sold in the successful orders.

Table 2 shows the statistics of the transaction times logged automatically at the server side and estimated by the test users. Column 'whole process' corresponds the time the user spent in the service in total (including e.g. browsing daily menus before initiating an order), whereas 'best time' refers to the time the user spent on actually making an order (steps 1-4 in Fig. 3). Column 'user estimate' corresponds to the times estimated by the test users in the questionnaire. Subcolumns 'whole' and 'payment' correspond to the time spent in the ordering sequence (steps 1-4) and in the e-maksu payment service, correspondingly.

Table 2. Statistics of transaction times.

		Best time		Whole process		User estimate	
		Whole	Payment	Whole	Payment	Whole	Payment
IPAQ	Average	02:40	01:05	03:48	01:57	05:18	02:02
	Std dev	02:24	01:16	03:59	02:22	04:55	03:38
	Median	01:44	00:42	02:23	01:13	03:00	01:00
	Min	00:44	00:18	00:44	00:18	00:30	00:10
	Max	10:10	07:46	19:10	10:19	20:00	18:00
Smart-Phone	Average	04:44	02:08	07:19	04:00	05:28	02:11
	Std dev	03:36	00:49	06:53	05:27	02:43	01:18
	Median	04:03	02:00	04:37	02:15	05:00	02:00
	Min	00:25	00:10	00:25	00:10	02:00	00:30
	Max	22:33	03:40	28:03	25:23	12:00	06:00
PC	Average	00:51	00:24	02:06	01:13	04:05	04:02
	Std dev	00:22	00:16	02:33	01:57	06:04	08:43
	Median	00:49	00:21	00:54	00:28	02:00	01:00
	Min	00:23	00:10	00:27	00:13	01:00	00:20
	Max	01:30	01:10	09:35	07:03	22:00	30:00
Total	Average	03:23	01:29	05:13	02:49	05:13	02:23
	Std dev	03:12	01:10	05:46	04:13	04:10	03:59
	Median	02:38	01:17	03:09	01:48	04:00	01:15
	Min	00:23	00:10	00:25	00:10	00:30	00:10
	Max	22:33	07:46	28:03	25:23	22:00	30:00

We observe a great variation in the transaction times of smart phone and PDA in comparison to PC. Further, the average times spent in placing the actual order (steps 1-4) (0:51 for PC, 2:40 for PDA and 4:44 for smart phone) reflect the relative overall usability of the devices (PC >> PDA >> smart phone).

Every test user reported some kind of usability and/or technical problems. Since most of the test users were not familiar with the mobile devices used in the field trial, much less familiar with using them for Internet services, it is difficult to attribute the usability problems into a particular feature of the device instead of the SmartRestaurant service and vice versa. For example, Nokia 3650 has a very unconventional circular keyboard, which test users found quite awkward. Similarly, the Opera browser's excessive use of shortcut keys and modes may be useful to expert users, but caused problems to novice users. Generally, using a web service with the browser of a mobile phone was found rather cumbersome.

Most of the technical problems were related to wireless connectivity. According to some test users the GPRS call setup was too long and the connection speed was found slow. The PDA users disliked the separate authentication into the panOULU network.

The users found the e-maksu payment service too complicated, since it requires a registration, uploading of funds from the bank account and a separate authentication during each transaction.

Regarding the SmartRestaurant service, test users would have liked to have access to complete daily menus instead of just the special lunch. Similarly, test users would have liked to use the SmartRestaurant service in all restaurants at the campus. Some test users also requested for the possibility to reschedule or cancel their lunch order in case something prevented them to have lunch at the requested delivery time.

However, despite the aforementioned problems test users regarded the SmartRestaurant as a potential service. They could order their lunch beforehand at any place and at any time.

The restaurant employees found the system to be very helpful. The chef could start preparing the lunches before the customers came to the restaurant. The cashiers thought that it was easy to serve the SmartRestaurant customers because the payment had been taken care of beforehand and the lunches were ready waiting for the customer.

4. CONCLUSIONS

This paper introduced the SmartRestraunt service, a web service for ordering and paying lunch orders in a context-aware mobile environment. We also presented a practical deployment and empirical evaluation of the service in the real environment of use. SmartRestaurant was proven to be a functional concept useful for both customers and the restaurant.

The biggest shortcoming in the current service is that it requires the SmartRestaurant customer to eat at the chosen delivery time. It would be handy if the customers could place immediate order, say just when leaving their office towards the restaurant. Alternatively, the customers could be given the possibility to reschedule or cancel lunch orders.

5. ACKNOWLEDGEMENTS

The financial support of the National Technology Agency of Finland and the GETA graduate school is gratefully acknowledged.

6. REFERENCES

[1] Aalto, L., Göthlin, N., Korhonen, J. and Ojala, T. (2004) Bluetooth and WAP Push Based Location-Aware Mobile Advertising System. *Proc. Second International Conference on Mobile Systems, Applications and Services*, Boston, MA, 49-58.

[2] Dey, A. K. (2001) Understanding and Using Context. *Personal and Ubiquitous Computing, 5, 1,* 4-7.

[3] European Commission (1997) A European Initiative in Electronic Commerce. Communication to the European Parliament, the Council, the Economic and Social Committee

and the Committee of the Regions, COM(97)157. http://www.cordis.lu/esprit/src/ecomcom.htm.

[4] e-maksu (2004) https://www.e-maksu.com [referenced 1 Apr 2004]

[5] GuangChun, L., Lu, W. and HanHong X (2003) A Novel Web Application Frame Development by MVC. *ACM SIGSOFT Software Engineering Notes, 28, 2*, 7.

[6] Hibernate (2004) http://www.hibernate.org/ [referenced 1. Apr 2004]

[7] Jin, L. and Miyazawa, T. (2002) Context and Location: MRM server: a context-aware and location-based mobile e-commerce server. *Proc. 2nd International Workshop on Mobile Commerce*, Atlanta, GA, 33-39.

[8] Kaasinen, E. (2003) User needs for location-aware mobile services. *Personal and Ubiquitous Computing, 7, 1*, 70-79.

[9] Nokia and 2Scoot (2001) http://press.nokia.com/PR/200101/803668_5.html [referenced 4 Apr 2004]

[10] Octopus (2004) http://www.mobileforum.org/octopus [referenced 1 Apr 2004]

[11] Ojala, T., Korhonen, J., Aittola, M., Ollila, M., Koivumäki, T., Tähtinen, J. and Karjaluoto, H. (2003) SmartRotuaari – Context-aware mobile multimedia services. *Proc. 2nd International Conference on Mobile and Ubiquitous Multimedia*, Norrköping, Sweden, 9-18.

[12] panOULU (2004). http://www.panoulu.net [referenced 4 Apr 2004]

[13] Sonera (2004). http://www.sonera.fi/ [referenced 4 Apr 2004]

[14] Sonera Shopper (2004) http://www.sonera.fi/shopper [referenced 4 Apr 2004]

[15] Seshadri, G. (1999) Understanding JavaServer Pages Model 2 Architecture http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html [referenced 4 Apr 2004]

[16] Sun Microsystems (2004) http://java.sun.com/developer/technicalArticles/javaserverpages/servlets_jsp/ [referenced 4 Apr 2004]

[17] Varshney, U. (2003) Location Management for Mobile Commerce Applications in Wireless Internet Environment. *ACM Transactions on Internet Technology, 3, 3*, 236–255.

[18] VISA U-Commerce. http://www.corporate.visa.com/av/ucomm/u_whitepaper.pdf [referenced 4 Apr 2004]

[19] Zonepay (2004). http://www.zonepay.com [referenced 4 Apr 2004]