Int. Computer Symposium, Dec. 15-17, 2004, Taipei, Taiwan.

Web Service-Oriented Architecture for Pharmacy Management in Mobile Environments

Gilda Pour, Cindy Nguyen, Reginald Mangoba
College of Engineering
San Jose State University, San Jose, CA, U.S.A.

Abstract*-Ever-increasing demands for more efficient access to pharmacies on a 24x7 basis as well as rapidly growing needs for cost-effective pharmacy management systems have motivated research to develop new architecture that supports access of authorized users to pharmacy through both Web browsers and mobile devices. Our research[1] has focused on development of a new Web service-oriented architecture for pharmacy management systems in mobile environments. The new architecture allows access of authorized pharmacy system administrators, physicians, patients, and pharmacists to pharmacy through mobile devices as well as Web browsers. In this paper, we present our new Web service-oriented architecture. We have also developed a prototype of the pharmacy management system with the following key functions: prescription processing, medication knowledge-base management, pharmacy inventory management, and billing. In this paper, we also present the demonstration prototype that we have developed using the Java 2 Platform, Micro Edition (J2ME) and J2EE-based Web services.*

Keywords: Mobile applications, Web services, architecture, J2ME, J2EE

1. Introduction

The problems of unsatisfactory level of service and high rising cost of health care are associated with medical staff shortages, an aging population, high risks to patient safety, and cash-strapped hospitals. The use of mobile technology has emerged as a solution to the problem. The U.S. wireless market for health care is expected to increase from $330 million in 2003 to over $637 million by 2007 [1].

The ever-increasing widespread use of mobile devices and Internet, and high demands for access to pharmacies on a 24x7 basis through mobile devices have motivated the need for a new architecture that could support both Web browsers and mobile devices.

This need is addressed in our research project[2]. We have developed Web service-oriented enterprise architecture that supports the following key functions of pharmacy management in mobile computing environments and through Web browsers:

- Prescription processing: Creating and processing new prescriptions and refill orders.
- Medication knowledge-base management: Displaying information about a medication including its common uses and possible side effects.
- Pharmacy inventory management: Managing the inventory of medications and medical supplies.
- Billing: Automatically computing the total cost of a prescription order, and processing bills.

The mobility feature of the new system allows physicians, pharmacists, and patients to access the pharmacy management system anytime, anywhere using a Java-enabled mobile device. To support mobile application-to-application interactions on the Internet, we have developed Web services that are based on Java 2 Platform, Micro Edition (J2ME). J2ME is a version of Java aimed at devices with limited hardware resources, such as PDAs and cell phones.

This paper is organized as follows. Section 2 provides background and motivation. Section 3 presents our new Web service-oriented architecture. Section 4 provides the highlights of our prototype development. Section 5 presents final remarks and future directions for this project.

2. Background and Motivation

On-line pharmacies, such as those run by Kaiser Permanente (http://www.kaiserpermanente.org) and Longs Drugs (http://www.longs.com/), provide 24x7 assistance to patients with their prescription needs through Web browsers. However, there is a need for new enterprise architecture that supports access of authorized users to pharmacy through both Web browsers and mobile devices in a seamless manner. Such architecture will improve not only accessibility

[1] Our research has received sponsorship from IBM and Hewlett-Packard Company.

[2] Sponsored by IBM and Hewlett-Packard Company

but also efficiency and cost-effectiveness for parties involved.

To address this critical need, our research has focused on development of a Web service-based architecture that supports access of authorized users to pharmacy management system via mobile devices and Web browsers.

Web services are platform-, language-, and location-independent. As long as a Web service is publicly available, it can be accessed by authorized applications. Web services offer attractive features, such as extensive reusability and interoperability between various software applications running on different platforms [2].

Web-based applications have moved away from the data-centric nature of client/server applications towards a service-oriented architecture [2]. An enterprise application must have a standard form of communications, a common data representation and data exchange mechanism, standard meta-language to describe its services, and a mechanism to register and locate other Web-based applications.

Web services support interoperable machine-to-machine interactions over the network. The interoperability among Web services is provided by the following XML-based open standards: Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), and Universal Description Discovery and Integration (UDDI). The eXtensible Markup Language (XML) is a cross-platform, extensible and text-based standard for representing data. All forms of communications in Web services utilize XML format.

WSDL describes what a particular Web service can do, where it resides, and how it can be invoked. WSDL is in a standardized XML format. SOAP is a general-purpose protocol for sending XML messages between service endpoints over any transport layer. Most commonly used transport layer is HTTP. However, SMTP (Simple Mail Transport Protocol) and JMS (Java Messaging Service) can also be used. UDDI is a registry for connecting providers and users of Web services. A provider uses the UDDI to publish API to register a Web service. A user utilizes the UDDI to inquire API for locating Web services that satisfy a certain criteria. These standards provide a common and interoperable approach for defining, publishing, and using Web services.

As a result, Web applications have evolved from simple applications accessed through an intranet to enterprise applications with Web service-oriented architectures.

3. The Architecture

Figure 1 illustrates a use case model for pharmacy management system. As shown in Figure 1, the current system has four main types of users: pharmacy administrators, physicians, patients and pharmacists. A user type determines the set of operations that a user is allowed to perform on the pharmacy management system. The system allows an authorized user to login to the system, look up medication information, and update his/her profile (e.g. username, password, contact information, and email address).

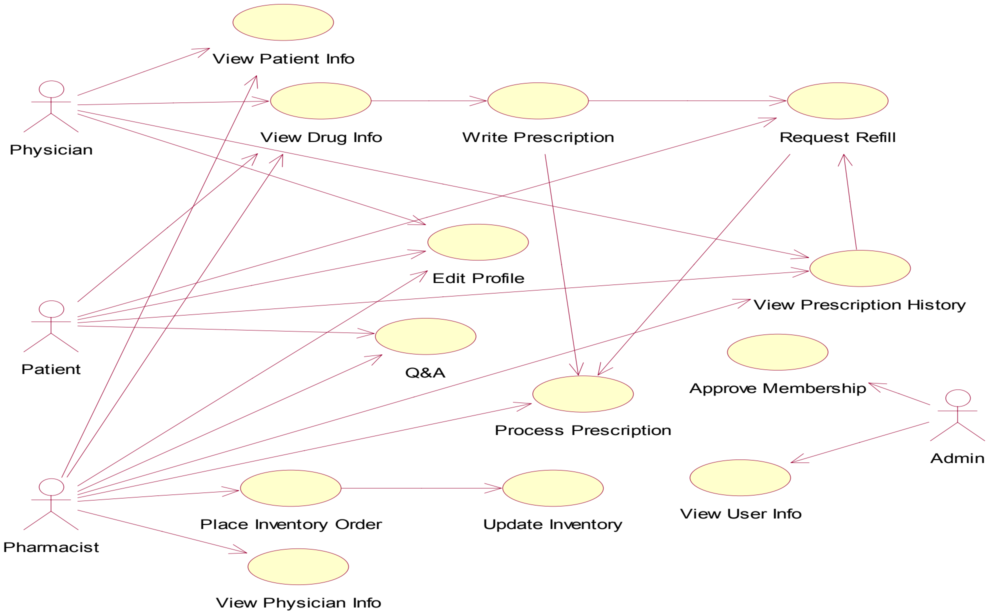


Figure 1. A Use-case Model for Pharmacy Management System

The system also allows:

- an authorized pharmacy administrator to login to the system, update membership list, search for a member on the membership list, and view contact information;
- an authorized physicians to search and view a patient's medical history, prescribe medications and authorize refills;
- an authorized patient to request a refill of his/her prescription(s), view his/her prescription history and corresponding medication information, and communicate with a pharmacist in charge of processing his/her request; and
- an authorized pharmacist to view prescription orders, process prescription orders, search and view patient's prescription history (when authorized), send a notification to each patient and prescribing physician (as required) once the order is processed, and update the pharmacy inventory.

We have designed the architecture with four major units; each performing one of the following key functions: prescription processing, medication knowledge-base management, pharmacy inventory management, and billing. The units in the new architecture are designed in the form of Web services and software components.

The new architecture supports two types of clients: (1) Web Browser clients and (2) J2ME clients. To support mobile devices, we have developed J2ME Web services. Java 2 Platform, Micro Edition (J2ME) specifications define a set of APIs and runtime environments for small embedded devices such as PDAs and Java-enabled cell phones. J2ME provides flexible user interfaces, robust security model, and a wide range of network protocols. The J2ME platform is deployed on various devices such as Java-enabled cell phone and PDAs.

Although the J2ME and Web browser clients may access different servlets and JSP components in a Web container, they often indirectly use the same EJB components for application logic and persistent data. J2ME Web Services Specification (JSR-172) defines a method for the use of Web services on J2ME devices [3]. It also allows developing Web services as J2ME clients. The two main components of the specification are compact versions of the Java API for XML parsing (JAXP) and JAX-RPC, which defines how J2ME devices can execute remote procedure calls.

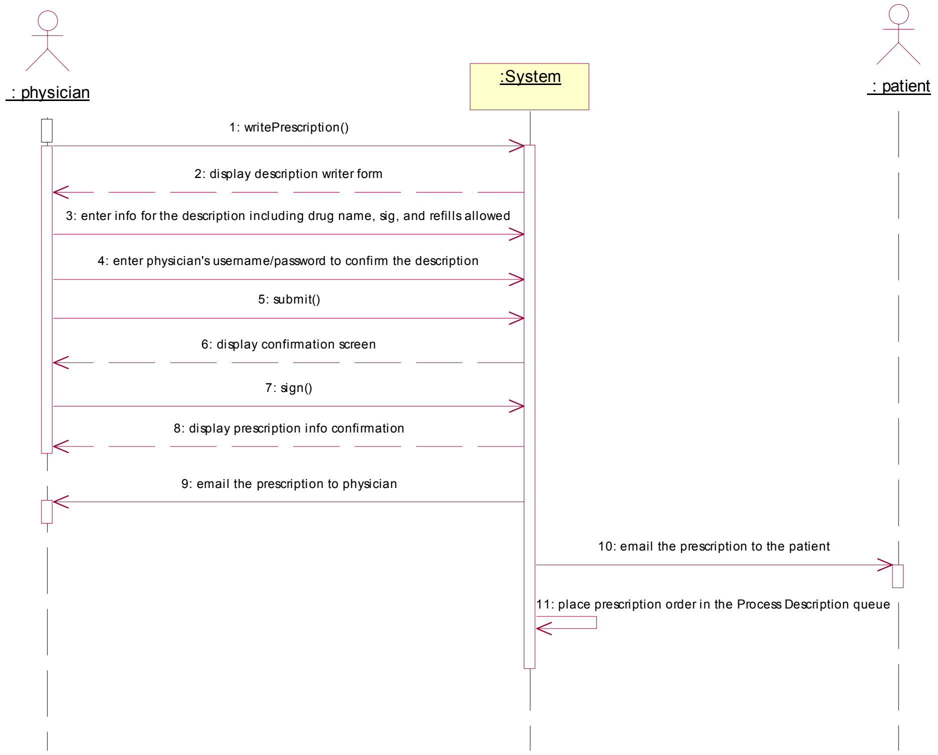

: physician
:System
: patient
1: writePrescription()
2: display description writer form
3: enter info for the description including drug name, sig, and refills allowed
4: enter physician's username/password to confirm the description
5: submit()
6: display confirmation screen
7: sign()
8: display prescription info confirmation
9: email the prescription to physician
10: email the prescription to the patient
11: place prescription order in the Process Description queue

Figure 2. A Sequence Diagram for Prescription Writing

Figure 3 shows Web service-oriented architecture with both J2ME and Web browser clients. As shown in Figure 3,

- J2ME clients and Web browser clients are located in client tier.
- Both J2ME clients and Web browser clients may access different Java servlets and JavaServer Pages (JSP) components in the Web container via a secure HTTP. The Web container is referred to as the "View".
- The EJB container is a "Controller". It is located in the application layer and directs the workflow.
- The Database server, located in the domain layer, is referred to as "Model".
- The servlets and EJB components may use different APIs to access enterprise information and services. For example, they use the JDBC API to access a relational database, or the JavaMail API to send an e-mail to a user.

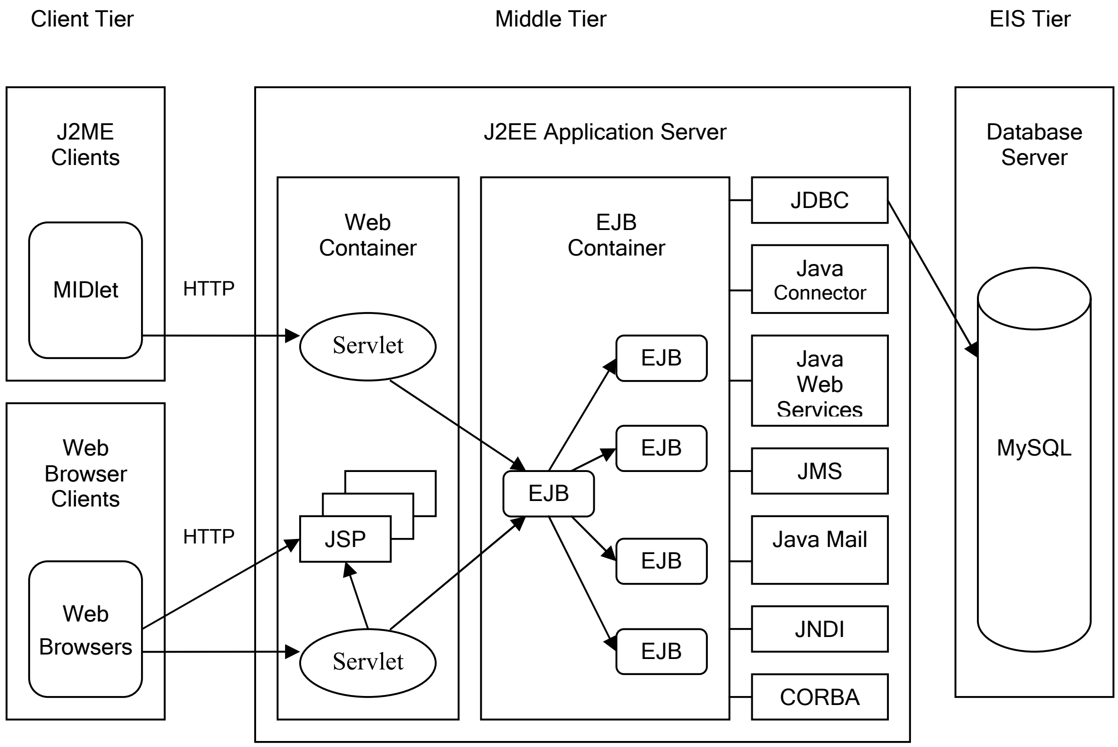


Figure 3. The Web-Service Architecture for both J2ME and Web Browser Clients

Figure 4 illustrates the high-level architecture of a J2ME-based mobile application. As shown in Figure 4,

- The client tier consists of a MIDlet, which provides a user interface on a mobile device. MIDlet communicates with a Java servlet, usually via HTTP, over a secure channel when necessary.
- The servlet in the Web tier interprets a client's request received from MIDlet, and dispatches the request to EJB components. Once the client's request is fulfilled, the servlet generates a response to MIDlet.
- In the EJB tier, the EJB components encapsulate the application logic. Both servlets and EJB

components are located on a J2EE application server.

- The Enterprise Information Systems (EIS) tier supports access to the application database through the JDBC API.

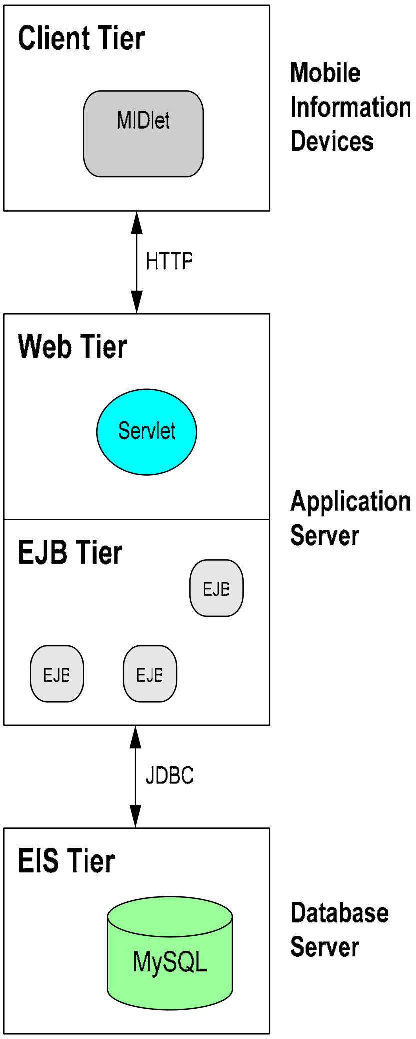


Figure 4. J2ME Mobile Application Architecture

4. Prototype Development

We have developed a prototype of the system with the Web service-oriented architecture presented in the previous section, J2ME and Java 2 Platform, Enterprise Edition (J2EE).

J2EE version 1.4 has evolved for developing Web services [4]. The J2EE platform is being used to develop Web services through the JAX-RPC API, which allows the development of service endpoints based on SOAP.

With J2EE, developing Web services does not require extensive knowledge of XML-based standards and protocols such as WSDL and SOAP. The mapping between Java language and such XML-based standards is handled by the Web service runtime system. J2EE platform also support portability, scalability and reliability.

As shown in Figure 3, we have used Java Server Pages (JSP) technology. JSPs are server side modules that execute on a JSP-enabled Web server such as Apache Tomcat server (http://jakarta.apache.org/tomcat/). The JSPs produce and display dynamic HTML content viewable on a Web browser. JSPs can also act as a client to create Java objects, EJB components or Web services.

We have used JavaScript for validation on various JSPs throughout the application. JavaScript is a Web scripting language allowing Web authors to create dynamic pages that react to user interactions. JavaScript functions are embedded in the HTML produced by the JSPs.

The key difference between JavaScript and JSPs is that the browser executes JavaScript. JavaScript is used for various operations, including event handling and client side validation of user forms.

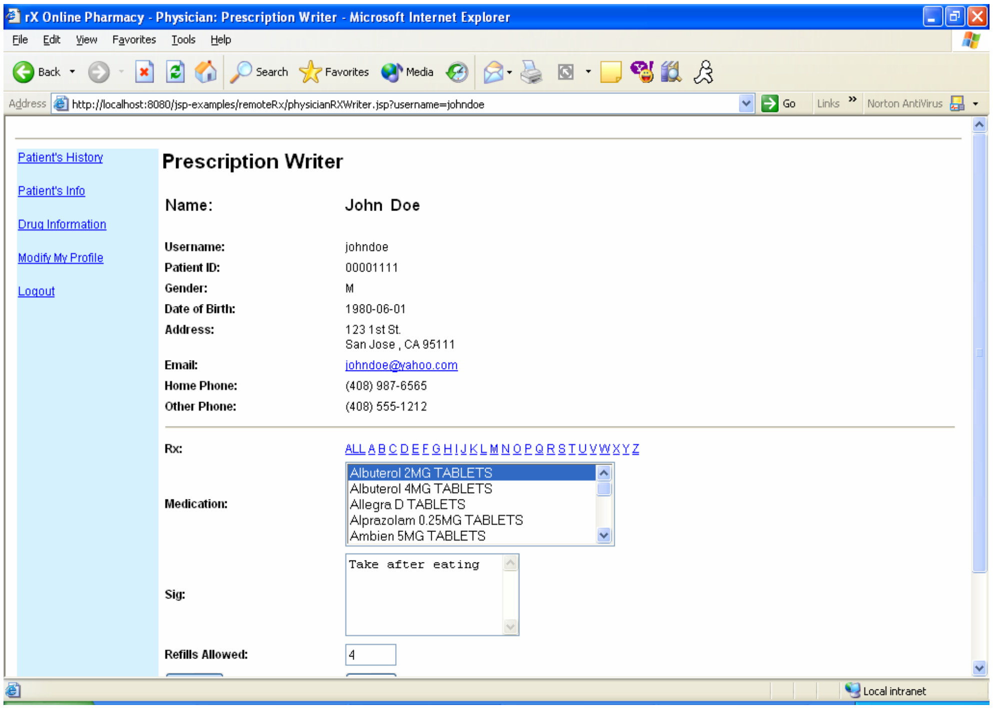
We have used webMethods Glue for creating and deploying applications with Web services, JSPs and servlets. It makes it easy to publish a Java object or EJB as a Web service. It includes a compact, high-performance implementation of important standards such as HTTP, Servlets, XML, SOAP, WSDL and UDDI. It interoperates with Microsoft .NET, IBM WebSphere, BEA WebLogic, Apache Axis and other Web service platforms.

In this project, the Web services call Java object methods that access the database and send emails.

In the enterprise information systems (EIS Tier), we have used MySQL database server, the popular open source database. Web services and EJB components, located in middle tier, access MySQL database using a datasource. The datasource supports portability while improving performance.

In addition, MySQL supports ANSI SQL syntax and query caching. In order for the Web services and Java objects to interact with MySQL database, MySQL Connector/ODBC is used. MySQL Connector/ODBC allows database connections using the ODBC database API.

Figures 5-8 illustrate some of the results of our prototype development.


rX Online Pharmacy - Physician: Prescription Writer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/jsp-examples/remoteRx/physicianRXWriter.jsp?username=johndoe
Go Links Norton AntiVirus
Patient's History
Patient's Info
Drug Information
Modify My Profile
Logout
Local intranet

Prescription Writer

Name: John Doe

Username: johndoe
Patient ID: 00001111
Gender: M
Date of Birth: 1980-06-01
Address: 123 1st St.
San Jose , CA 95111
Email: johndoe@yahoo.com
Home Phone: (408) 987-6565
Other Phone: (408) 555-1212

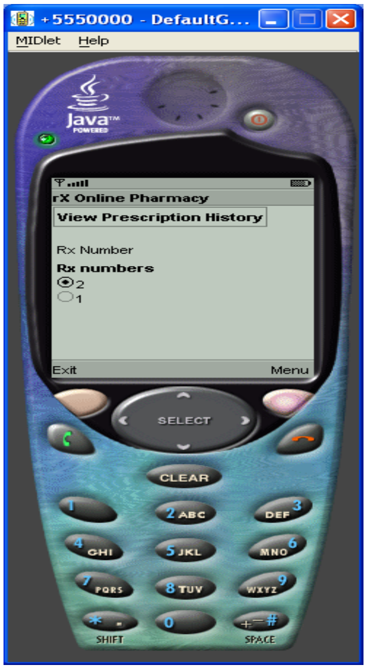
Rx: ALL A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Medication:
Albuterol 2MG TABLETS
Albuterol 4MG TABLETS
Allegra D TABLETS
Alprazolam 0.25MG TABLETS
Ambien 5MG TABLETS

Sig: Take after eating

Refills Allowed: 4

Figure 5. Screenshot of a Physician Writing a Prescription



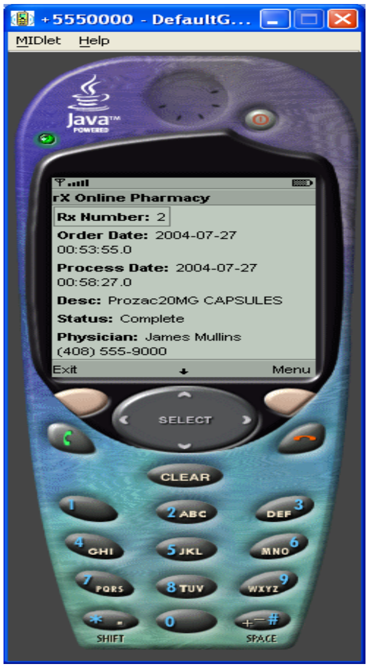
rX Online Pharmacy
View Prescription History

Rx Number

Rx numbers
● 2
○ 1

Exit Menu

Figure 6. Viewing Prescription History on a Cell Phone

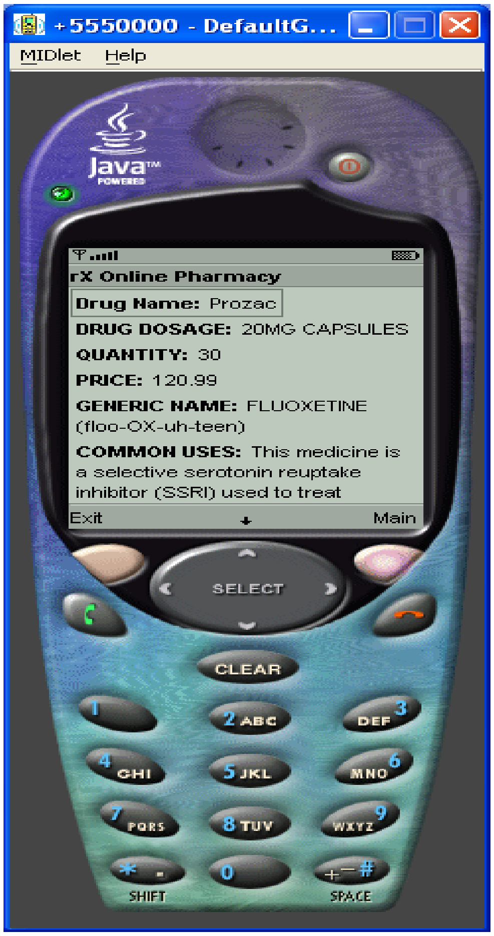


rX Online Pharmacy

Rx Number: 2
Order Date: 2004-07-27 00:53:55.0
Process Date: 2004-07-27 00:58:27.0
Desc: Prozac20MG CAPSULES
Status: Complete
Physician: James Mullins (408) 555-9000

Exit Menu

Figure 7. Ordering a Prescription on a Cell Phone



rX Online Pharmacy

Drug Name: Prozac

DRUG DOSAGE: 20MG CAPSULES

QUANTITY: 30

PRICE: 120.99

GENERIC NAME: FLUOXETINE (floo-OX-uh-teen)

COMMON USES: This medicine is a selective serotonin reuptake inhibitor (SSRI) used to treat

Exit Main

Figure 8. Refill Request Confirmation on a Cell Phone

5. Final Remarks

Our research has addressed the need for new architecture to support access of authorized pharmacy system administrators, physicians, patients, and pharmacists to pharmacy through both Web browsers and mobile devices. In this project, we have developed a new Web service-oriented architecture for pharmacy management systems in mobile environments.

We have also developed a demonstration prototype of a pharmacy management system using the new Web service-oriented architecture.

The results of the prototype evaluation have been promising. Our research results can be used to develop Web-based enterprise systems for mobile environments in some other application domains, not necessarily in pharmacy management.

In the next phase of this project led by the author, we plan to enhance the adaptability and flexibility of the system using agent components [5-11]. Our enhancement plan includes extending the research using mobile agent components and developing a demonstration prototype with JADE-LEAP. JADE-LEAP is the combination of Lightweight Extensible Agent Platform (LEAP) [12] and Java Agent DEvelopment Framework (JADE) [13]. Our planned research project is directed toward the pervasive computing world--next-generation computing.

References

[1] Wireless Market Shows Signs of Life in Health Care. Retrieved March 2004, http://www.ihealthbeat.org/index.cfm?Action=dspItem&itemID=101508.

[2] M. Chitnis, M., Tiwari, P., & Ananthamurthy, L. (n.d.) Introduction to Web Services Part 2: Architecture. Retrieved March 2004 from http://www.developer.com/services/article.php/1495091.

[3] Designing Wireless Clients for Enterprise Applications with Java Technology, 2003.http://java.sun.com/blueprints/earlyaccess/wireless/designing/designing.pdf.

[4] K. Topley, *Java Web Services in a Nutshell*, O'Reilly & Associates, 2003.

[5] G. Pour, "Multi-Agent System Architecture for Quality Control: Moving Toward Autonomic Systems for E-Commerce," *E-Commerce Series*, 2004.

[6] G. Pour, "Web-Based Multi-Agent Architecture for Distributed Collaborative Software Engineering," *Applied Computing Series*, 2004.

[7] M. Griss and G. Pour, "Accelerating Development with Agent Components," Cover feature article, *IEEE Computer*, Vol. 34, No. 5, 2001.

[8] G. Pour, "Multi-Agent Autonomic Architecture for Water Quality Control," *Intelligent Agents, Web Technologies and Internet Commerce,* 2004.

[9] G. Pour and X. Wei, "A Multi-Agent Framework to Support Web-Based Collaborative Environment for Drug Prescription Management," *Intelligent Agents, Web Technologies and Internet Commerce,* 2004.

[10] G. Pour, "Internet-Based Multi-Agent Architecture for Service Delivery," *Internet Computing Series*, 2004.

[11] G. Pour, F. C. Yao and C.C. Yu, "A Mobile Agent-Based Architecture for Mobile Systems Supporting Distributed Software Project Management," *IEEE SoftCOM*, 2004.

[12] http://leap.crm-paris.com/.

[13] Bellifemine, F. et al, 1999. JADE: A FIPA-Compliant agent framework. *Proc. Practical Applications of Intelligent Agents and Multi-Agents.* pp. 97-108.