

Design and Development of a Push-based Point of Sale System (PUPS)

Bina Ramamurthy

CSE4/587 Information Structures

Due Date: /2005 by mid-night.

Purpose:

1. Design and develop a multi-tier distributed system offering remotely accessible services.
2. Understand the components, the core technologies, the architecture and the protocols enabling a J2 Enterprise Edition (J2EE)-based distributed system.
3. Design and implement system processes using Enterprise Java Beans (EJB).
4. Understand the process of preparing and deploying an interoperable remote service.
5. Design and develop a push-based point of sale system that functions by interoperation of several autonomous distributed systems.

Preparation:

1. Get a clear understanding of multi-tier distributed systems. (See lecture notes).
2. Understand the technology underlying the J2EE: its architecture and application models. See <http://java.sun.com/developer/onlineTraining/J2EE/Intro2/j2ee.html>
3. Learn how to use the XML-based build tool Ant at <http://ant.apache.org/>
4. Understand the role of deployment descriptors. The deployment descriptors are XML files used to configure runtime properties of an application thus relieving application to deal only with the programmatic details.
5. Learn to use the application interface to the Oracle database using embedded SQL and JDBC. Alternatively you may use a file-based simple database Cloudscape.
6. Download and install [Macromedia](#) JRun4 Developer edition and then the Updater2 (service pack). JRun4 is a J2EE compliant software environment for developing distributed systems. This can be done either or both in the project space that will be allocated to you and at your home, if you have the facility.

Technology details:

J2EE offers a suite of software specification to design, develop, assemble and deploy enterprise applications. It provides a distributed, component-based, loosely coupled, reliable and secure, platform independent and responsive application environment. It encompasses technology solutions for all tiers of a distributed system: client tier, web tier, (business) logic tier, and enterprise information system (database) tier. Sun Microsystems Inc. provides a reference implementation of J2EE compliant environment and many businesses offer fine products such a Macromedia JRun4 and BEA Weblogic for J2EE-based development. For this project, we suggest you use JSP (Java Server Pages) for the web-tier, EJB (Enterprise Java Bean) for the logic tier, and any relational data base (Cloudscape or Oracle) for the data-tier. An *enterprise bean* is a server-side component that contains the business logic of an application. At runtime, the application clients execute the business logic by invoking the enterprise bean's methods. Enterprise Java Bean architecture frees the application developer from having to deal with the system level aspects of an application. Developer can deal with the programmatic aspects of the application while the systemic needs of the application such

as data base driver and message queue can be specified declaratively. Ultimate goal of introducing J2EE at this point is to encourage the students to compare it to the grid technology that will be discussed later in the semester.

Assignment (What to do?):

Consider a very common operation of sales of a product, but with a twist. Instead of a customer walking or driving into a store to order, as the customer moves or drives around designated malls or wireless markets, the stores will be able to push the information on to a customers preferred communication device (cell phone, blackberry or custom screens on strategic locations). This will enable the customer to be more effective (and efficient) in shopping model. Major systems collaborating in this effort would be the (i) the customer database where the customers would have register to avail the push feature and (ii) shops with their product list and prices stored in a database. For simplicity, we will assume there is no explicit money transfer except the customer is charged with the amount spent in their registered account.

A person (customer) who wants to avail this push-based mall will register himself or herself with the mall organization. A personal profile including account for crediting and debiting money is set up in the profile of the customer. Then he/she will log into the system when they enter the mall. As they move around will be pushed the various information about graphical and/or text about the products the shops sell. Assume at least three different types of products/stores: music store, book store and anything else of your choice. Customer can respond to push messages by responding to it or by passing it. When customer responds, it could be for more details or for ordering. Customer can select items and place order. Customer's selection will be transferred to the shop's server. Order will be placed and customer account will be updated for the bill amount. The system will also allow a single point return of the item a customer bought.

Your assignment consists of three parts: (i) design and implement the databases, (ii) design and implement the user interfaces, and (iii) design and implement the tiers representing the push mall (PUPS). You will work in groups of not more than two people.

Analysis and Design:

Server side: Research and analyze the problem to understand the requirements. Represent the system requirements using UML (Unified Modeling Language) use case diagram. Identify the entities, processes and rules. Discover classes needed to implement the processes and entities. The rules are typically represented by methods in the classes. Represent the classes and relationship among them using a UML class diagram. Decide which among the classes will have methods that will be exposed to the users. Typically these will be implemented as enterprise components (EJBs in our case). Design a relational database to store persistent entities. The design document at the end of this phase will have use case diagram(s), class diagram(s), and a diagram (Entity-Relationship diagram) representing the database design. These documents have enough information to start coding.

Client side: Design and implement simple but creative user interfaces that will display the pushed information and allow the user to interact with the PUPS.

Implementation steps and details:

- 1. Getting used to building client-server systems:** When you implement a simple client side application program there are just two steps involved: compile and execute the code. In a client-server system, you will have to take care the server side as well as the client side. On the server side, you will compile the code, generate stubs or proxies using special compilers, deploy the service, register and publicize the service for the clients to use. On the client side you will prepare the client code with appropriate stubs, and during execution lookup the service needed and use it. You will notice that besides simple compile and execute, configuration and deployment of a service are important issues to be reckoned with.
- 2. Working with the relational database and embedded SQL:** In this project you will store the data in a relational table and access it using SQL statements embedded in Java language. Work on a simple java program to refresh your knowledge about accessing the Oracle database. See <http://www.cse.buffalo.edu/local/Consulting/Oracle.html> for examples and access details.
- 3. Building systems using build tools such as Ant:** In order to tackle complexities in configuration and deploying server-side applications, you will need to use special build tools. [Apache Ant](#) is a XML-based build tool which similar to “make” utility that most of you are familiar with. This topic will be covered during the recitation this week. Work on simple files to familiarize yourself with the Ant build tool.
- 4. Study and understand Enterprise Java Bean building and deployment details:**
 - a. You will use Macromedia JRun developer edition for the J2EE components. Download details will be discussed during the recitations. They are quite simple. You can work at home by downloading one into your personal computer and bring the deployable units to school for deployment.
 - b. Study the examples in the documentation that comes with the JRun installation.
 - c. For the database you may use the database that comes with JRun or Oracle database.
 - d. For the client-tier we suggest that you use JSP. We will cover JSP and servlet during the recitation.
 - e. While you have choice of technology in implementing data tier and client-tier, it is required that main exposed business logic should be implemented using EJBs. However, utilities supporting the business logic can be implemented using regular Java classes.
 - f. It is very important that you understand the concept of remote method call, name resolution, registering and lookup. The concept of component programming using

EJBs is also equally important. We will discuss these with examples during lecture.

5. **Design, implement and test PUPS server-side:** This is expected to be the most time consuming part of the project due to the novelty of the topic.
6. **Deploy the integrated system:** The various tiers can be deployed and tested individually.
7. **Practice good programming style:** Finally, practice all the good programming styles that you learned in the lower-level courses.

Submission Details:

Create a project1 directory and use that as the working space. Let the code directory tree be located in this directory. Let the design be represented by an integrated class diagram and presented in a file project1.pdf. Provide internal documentation using javadoc style comments. You will create a README file containing the details of the package and processing. Zip the project1 directory and submit the resulting zip file, project1.zip. Making sure that your current directory contains your project1 directory, you can create this file as follows:

```
zip -r project1.zip project1
```

Use the electronic submission program that corresponds to your class (cse4/587). For instance students in cse587 will submit by typing
submit_cse587 project1.zip
at the Unix prompt.

Documentation and Report: See [report details](#).