The Internet Store Project

Marcin Skubiszewski

19th February 2003

1 Overview

This document describes a simple yet complete web-based Internet store. As part of the *electronic commerce* course, we implement certain elements of the store.

At the beginning of the course, students receive a template implementing certain features of the store (in fact, a very small part of the store). The work of the students consists in understanding the template, and in extending it so as to implement more features. At any rate, students are not going to implement everything that is described in this document, because we do not have enough time for this.

Our store allows customers to search for articles for sale, to display detailed descriptions of selected articles, and to order articles. After ordering articles, a customer can check the status of her orders; specifically, she can check whether and when a given article has been shipped. The customer can cancel her orders as long as they have not been shipped.

The store includes a web-based interface that can be used by the store manager to update information about articles, about customer accounts, and about available stock.

For the sake of simplicity, the store exchanges information with its users exclusively through a web interface, and does not send or receive e-mail messages (in the real world, Internet stores do send e-mail).

The project runs on the top of the Linux operating system. The programming is done in the Java programming language, using Java Servlets and Java Server Pages (JSP), and using Jakarta Struts. Data are stored in the relational, SQL-based database management system PostgreSQL.

The software components that we use for the project (that is, Linux, Java, Java Servlet, Jakarta Struts, and SQL-based database systems) are widely used in real-world electronic commerce applications, including complicated, high-performance applications. Despite its relative simplicity, our store is organized and programmed exactly in the same way as many real-world online stores.

In Section 2 of this document, we describe how our Internet store behaves (how it interacts with customers and with the manager, what information it stores and provides to customers, etc.). In Section 3, we discuss the software environment used for the project. Finally, in Section 4, we state quality requirements that must be met by the work delivered by the students.

2 What the Internet store does

2.1 The articles for sale

2.1.1 How articles are classified and searched for

The store contains a number of **articles for sale**. Each article belongs to one or several **categories**. For example, we can have a category named ***boats*** and another one named ***fishing***. Boats especially suitable for fishing will be placed in both categories simultaneously.

Any Internet user can search our shop for articles for sale. The simplest form of a search consists in looking at the complete content of a given category: the user goes to a page containing the names of all the categories and clicks on a category name; in response, the store displays the complete list of articles in the category.

In the general case, a search is done through a Web form, and can use any combination of the following criteria:

- **categories:** the user is only interested in articles belonging to certain categories
- **category exclusion:** the user is only interested in articles not belonging to certain categories
- **price range**
- **plain text:** the user searches for articles containing certain words in their names or descriptions

Whenever a user performs a search using these criteria, the store responds with the list of all the articles satisfying the criteria.

For example, a user who looks for a fishing rod may use the following search criteria:

- group ***fishing***
- excluded group ***boat*** (so as to exclude fishing boats from search results)
- price range from 0 to 200 euro
- plain text "***rod***" (the user only wants to see articles containing the word ***rod*** in their names or descriptions)

2.1.2 How articles are displayed

For every article for sale, the Internet shop can display a ***mention***, a ***short description*** or a ***long description***.

Mention: The mention of an article only consists of its name and its code. For example, a mention may look like this:

Sevylor HF360 inflatable fishing boat (K77441234)

As explained below (Section 2.6.1), the mention of an article is clickable, and acts as a hyperlink leading to the long description of the article.

Short description: A short description is used whenever we want to display descriptions of several articles simultaneously, for example in search results or while showing the list of articles selected for purchase by a customer (this list is called ***the customer's shopping cart***). The short description contains an icon-size picture of the article (when available), its price, and a short descriptive text.

The short description contains a link labelled ***more info...***, leading to the long description of the same article, and a button labelled ***buy*** or ***add to cart***, that causes the article to be added to the customer's shopping cart (the shopping cart is described below, in Section 2.2).

For example, a short description can look like this:

[small picture here]	Sevylor FH360 inflatable fishing boat (K77441234)	size 3,40 by 1,50 meters, two fishing rod holders, can carry 400 kg	
	450,00 EUR	**add to cart**	**more info...**

Long description: The long description contains all the information available about an article, including a long descriptive text and a full-size picture (when available).

2.1.3 The information that we store about categories

Each category has

- a **name** (e.g., ***fishing***)
- a **description** (e.g., "everything you need to fish, including clothing and boats")
- a **code** (e.g., ***F12***); the code is guaranteed by the system to be unique and never to change; it is essentially used internally by the system, but it is likely to appear in URLs and can be displayed to the user.

2.1.4 The information that we store about articles for sale

Each article for sale has

- a **name**, e.g., "Sevylor FH360 inflatable fishing boat"
- a **short description**, e.g., "size 3,40 by 1,50 meters, two fishing rod holders, can carry 400 kg"
- an **additional description** that completes the short description, e.g.,

  > "This boat can can carry up to four adults. It can use a small electrical engine or a 2,5 kW gas engine. It takes approx. 30 minutes to inflate. Weight 20 kg."

- a **price** in euro, e.g., 450,00
- optionally, up to two **pictures**:
  - one **icon-size picture**, to be shown when the article is displayed in its short form
  - one **full-size picture**, to be shown when the article is displayed in its long form
- the **number in stock**: this number says how many articles of this type are in our warehouse, ready to be shipped to fulfill new orders (articles that are still physically present in our warehouse, but are going to be shipped to fullfill already-processed orders, do not count here).
- a **code** (e.g., ***K77441234***); the code is guaranteed by the system to be unique and never to change; it is used internally by the system, but it is likely to appear in URLs and can be displayed to the user as part of the description of the article.

2.2 Purchasing articles

2.2.1 Principle

The description of every article for sale contains a button entitled ***buy*** or ***add to cart***. When the user clicks on this button, the article is placed in the user's virtual shopping cart, i.e., is selected for purchase by the user.

When the shopping cart of a user contains one or more articles, the user can, at any time, choose to actually purchase the articles. For this purpose, the user follows the hyperlink labelled ***check out*** (as explained in Section 2.6.2 below, this link is present on every web page displayed by our Internet store).

The link leads to web pages that allow the user to verify the content of her shopping cart, to provide her name and her address, and to pay for the articles. The purchase becomes final after payment.

For payment, we use a cryptographically secure (secure HTTP) connection. The payment web page indicates the amount due, and asks the user to enter her credit card information (cardholder's name, expiration date and card number).

2.2.2 Customer accounts

To store the names and the addresses of customers, we use customer accounts: whenever a user enters her name and her address, a customer account is created for her, and the name and the address are attached to the account. Each account is identified by an **account name** and by a **password**.

In addition to storing names and addresses, accounts are used for storing information about orders previously placed by the corresponding customers.

Returning customers, i.e., customers who already have accounts, are encouraged to log into their accounts. In order to log into an account, it suffices to provide the account name and the corresponding password. While a user is not logged in, a ***login*** form is displayed on every web page sent to her, inviting her to log in. Similarly, while a user is logged in, a ***logout*** hyperlink is displayed on every page sent to her (see Section2.6.2).

While a customer is logged in, she is not asked to provide her name or her address; instead, the name and the address attached to the account are used by the Internet store.

2.3 Shipping

As soon as a purchase is complete, we ship all the articles purchased that are currently in stock. Articles that are not currently in stock will be shipped later, when they become available.

To cause an article to be shipped, our Internet store software generates a shipping label that contains

- the code of the article, and
- the name and the address where the article is to be shipped

When it is time to ship an article, the Internet store software generates the label and inserts it into a database table named ***shipmentTable***.

Once a shipment label is inserted into ***shipmentTable***, our Internet store software considers the shipment as done. We assume that some other piece of software will print out the label, and that the staff of our store will attach the label to the corresponding article, then actually ship the article.

2.4 Viewing and cancelling previously placed orders

While a user is logged into her customer account, she can, at any time, view a web page showing her past orders. The link **view/modify past orders**, present on every page displayed to the user, leads to that page (see Section 2.6.2 below).

The page in question shows

- all the articles that were ordered by this user, but not shipped yet, and
- N articles most recently shipped to this user (the choice of the value N is left to the students).

Next to each not-shipped-yet article, there is a button labelled ***cancel***. If the user clicks on this button, the order for this article will be cancelled, and the article will not be shipped.

In the real world, the cancellation of an order should result in the customer being reimbursed. In this project, hovever, there is no reimbursement, because payments are simulated.

Next to each already-shipped article, we display the date and time when the shipment was done.

2.5 Managing the store through a Web interface

Certain customer accounts are specially marked as **store manager accounts**. While a user is logged into a store manager account, she is a **manager**.

Managers have the following special capabilities, in addition to those offered to ordinary users:

1. When a manager displays the long description of an article, the description is automatically followed with
   (a) the corresponding *number in stock*, and
   (b) with the list of all unprocessed orders for this article.
2. A manager can display the list of all customer accounts. The manager can mark or unmark each account as being a store manager account.
3. A manager can display the list of all unprocessed orders.
4. A manager can display the **shipmentTable**.
5. A manager can create and suppress categories of articles. She can change the names and descriptions of existing categories.
6. A manager can create and suppress articles for sale. She can modify the description of an article for sale. She can add or remove an article to/from a category.

In this document, we do not specify how these capabilities should be implemented. Implementation choices are left to the students. The template provided to the students already implements the creation and modification of articles for sale.

While designing management-related capabilities, the students should remember that the management of the store will be done by professionals, not by customers. It is therefore not necessary to make the management interface easy to use by novice users; instead, the management interface needs to be fast to use: every operation should be possible to do in as few clicks and as little text typed as possible.

2.6 Rules concerning hyperlinks

2.6.1 Everything is clickable

Whenever a web page displayed by our Internet store mentions an object that the store can describe in detail, the mention of the object is a hyperlink to the detailed description.

For example, when an article for sale is mentionned like this

KN4720 Sevylor HF-360 inflatable boat

the mention is clickable, and is a hyperlink to a detailed description of the article (specifically, to its **long description**, see Section 2.1.2).

Similarly, when, in a page displayed to the store manager, a customer's account is mentionned by name, the name is a hyperlink to a complete description of the account, including the customer's address and unprocessed orders.

2.6.2 Hyperlinks present on all Web pages

Every web page displayed by our Internet store should contain hyperlinks and forms that can be used to perform certain common operations. (In the template provided to the students at the beginning of the course, these links and forms are located in the left part of the screen, and are described in file `choices.jsp`.)

Specifically, the following links and forms should be present:

- **categories**: a hyperlink to the complete list of categories of articles in our store (see Section 2.1.1)
- **search**: a form that the customer can use to search for articles (see Section 2.1.1)
- **view cart**: a hyperlink to a web page that lists the content of the user's shopping cart (see Section 2.2)
- **check out**: a hyperlink to a web page that can be used to complete the purchase of the articles previously placed in the shopping cart (this link is not present when the shopping cart is empty; see Section 2.2)
- **login**: a form that can be used to log into an existing customer account, i.e., to identify the user as being a specified returning customer (this link is only displayed while the user is not logged in; see Section 2.2)
- **create account**: a hyperlink that leads to a form for creating a new user account
- **logout**: a hyperlink or a button that can be used to log out of a customer account; after logging out, the user becomes anonymous (unidentified) (this link or button is only displayed while the user is logged in; see Section 2.2)
- **view/modify past orders**: the meaning of this is obvious (this link is only displayed while the user is logged in)

2.7 Extensions

As described so far, our Internet store is quite simple, and can be completed and extended in many ways. Here are some example extensions:

- **Suggested articles**: the descriptions of articles could contain suggestions to buy complementary articles. For example, electrical devices could carry the suggestion to buy corresponding batteries, a fishing boat sold with no engine could carry the suggestion to buy an engine and fishing rods. Whenever a user buys an article that has suggestions, the store could automatically display the suggestions.
- **Inclusion between categories**: we could permit a category to be explicitly included in another one (for example, eBay does this). For example, the category ***fishing rods*** could be included in ***fishing***. ***fishing boats*** could be simultaneously included in ***fishing*** and in ***boats***. ***fishing*** and ***boats*** could both be included in ***sports***.

  Among others, the inclusion would provide the following benefits:

  - Every article belonging to a category automatically belongs to all the categories that englobe it. This would simplify the management of the store: for example, once an article has been added to the category ***fishing boats***, it does not need to be manually added to ***fishing***, to ***boats*** or to ***sports***.

- For each category, the list of included categories could be displayed to the customer, making it easy for the customer to find the right category for an object.

  For example, a user looking for a fishing boat does not necessarily know that our store has the category ***fishing boats***. The user may, for example, start by looking at ***boats***. In this case, she will rapidly spot the included category ***fishing boats***.

3 Software environment

3.1 Linux

Linux is the operating system that we use for the project. The operating system is what accomplishes the most basic tasks of the computer: accessing peripherals, establishing network connections, organizining data on disks into files etc.

Linux is *free software*: anyone is allowed to modify Linux as they please, and to distribute the modified version (to compare: only Microsoft is allowed to modify or distribute Microsoft Windows). As a result of this policy, Linux is distributed by dozens of different companies and non-profit organizations, and comes in dozens of different versions (differences between those versions are often minimal).

The brand of Linux that we use on students' computers is Mandrake Linux. This is one of the most popular brands of Linux, and is probably the easiest one to install and use on a desktop computer (for servers, RedHat or Debian seem more appropriate; in our case, students' computers serve both as desktop computers and as web servers).

Using a popular brand of Linux is important, because it facilitates support: while using a popular brand, whenever you encounter a problem, you are likely to find people who have previously solved exactly the same problem. This facilitates both for commercial support, that a company like Mandrake or RedHat can sell you, and support that you can get for free over the Internet.

3.2 Java 2 Software Development Kit

Our Internet store is written in the Java programming language. Java is a language developped by Sun Microsystems to facilitate clear, modular programming. Java removes the risk of programming errors to the extent possible (of course, this risk cannot be removed totally). As a result, Java is often considered as being the best language for writing Web applications.

Java is entirely portable. For example, if we develop a Web application in Java under Linux, this application is likely to run perfectly well under Microsoft Windows and under many other operating systems.

The most serious competitors of Java for writing Web applications are scripting languages (PHP, Visual Basic with ASP, or Perl). By definition, scripting languages are primarily intended for writing small pieces of programming (say, up to 200 lines) called *scripts*, as opposed to bigger pieces called *programs*, which are usually written in programming languages.

Scripting languages are easier to learn than Java. It is very easy to write short scripts in a scripting language, but these languages lack many features necessary for writing larger programs; for example, they have a very lax type control, which allows many programming errors to remain undetected. As a result, I believe that scripting languages are used for Web development for wrong reasons: not because it is easy to complete the project this way, but because it is easier to begin this way (learn the language and write the first 200 lines). I strongly recommend against using scripting languages for any serious Web development.

Another competitor of Java is the Microsoft .NET project. I prefer Java for both technical and non-technical reasons. The technical reason is maturity: Java is mature today, and Microsoft's .NET is a very young project, not mature yet. Therefore, I expect to run into many more technical problems with .NET than with Java. The non-technical reason is that Microsoft only offers .NET under the Windows operating system (it is a general policy of Microsoft to offer software only under their own Windows system, rather than under a variety of operating systems). As a result, while using .NET you are forced to use Windows. Windows is much more expensive than Linux, and less convenient for building and managing web servers.

The implementation of Java that we use is the Java 2 Software Development Kit, version 1.4.1_1, from Sun Microsystems.

3.3 Java Servlets, Java Server Pages (JSP), Tomcat, and Jakarta Struts

The Java Servlet and Java Server Pages (JSP) specifications describe how our Internet store application generates webpages, controls the webserver, and reads data that the user has entered in web-based forms. Jakarta Struts offer a set supplementary mechanisms.

The respective roles of Java, Java Servlet, JSP, and Jakarta Struts are as follows:

- Java is the programming language that defines the syntax that we use, and is also a programming environment that will perform for us many basic operations (e.g., operations on character strings or dates).
- The Java Servlet specification describes how our Java program can communicate with the webserver. For example, the specification describes how we can obtain from the server the values that a user has filled in a Web-based form.
- Java Server Pages (JSP) is a method for mixing Web page fragments written in HTML with program fragments written in Java. And this is exactly what we need to generate *dynamic webpages*, that is, pages describing the (dynamically changing) content of our store and the current state of the purchasing process.
- Jakarta Struts is a *library* (a set of program fragments) that groups together many program fragments useful when developing an electronic commerce application. For example, Jakarta Struts will automatically convert web-based forms, that customers fill in, into Java objects, that are very easy to manipulate by Java programs. Without Jakarta Struts, accessing web-based forms would be much more complicated. By using Jakarta Struts in addition to Java Servlet, you save a lot of time.

We use the Tomcat webserver, version 4.1.18. Tomcat is developped by the Apache Foundation, but is not to be confused with the Apache webserver, the flagship product of the foundation that holds approximately 60 % of the webserver market. Tomcat 4.1.18 implements the specifications Servlet version 2.3 and JSP version 1.2.

The choice of Tomcat for this project is non-essential: as long as the project conforms to the Servlets and JSP specifications, it can run under any webserver that implements these specifications. Several such webservers exist.

3.4 The relational database system PostgreSQL

Data related to our Internet store are stored in a relational database. They are accessed and modified through queries written in SQL (*simple query language*).

A relational database consists of a number of *tables*, also called *relations*. Each table consists of a small number of columns. The number and the types of the columns determine the format of the data stored in the table. For example, a table named *people* may contain a column *first_ name* of type *text*, a column *last_ name* of type *text*, and a column *birth_ date* of type *date*.

A table stores data as *lines*, and contains a potentially large number of lines. In our example, each line in the table corresponds with one person, and contains her first name, her last name, and her birth date.

A relational database makes it possible to search for data efficiently, and to retrieve arbitrarily complex pieces of information by combining together related lines coming from different tables (the operation of combining lines is called *join*).

In theory, data related to our store could be stored according to various methods, some of which are much better than relational databases. However, relational databases and SQL are today's *de facto* industry standard for storing business-related data: a vast majority of existing web applications uses relational databases and SQL, and students are very likely to encounter relational databases in their future professional lives. In addition, relational database systems are a highly mature area of computing: many high-performance, highly reliable databases exist. For these practical reasons, we use a relational, SQL-based database system to build the Internet store.

We use the database system called PostgreSQL, version 7.2, originally from the University of California in Berkeley, and today under development by an independent group (The PostgreSQL Global Development Group). We choose PostgreSQL because it is free, and of high quality.

4 Quality requirements

This section contains advanced matter. Do not worry if you have trouble understanding it, explanation will be provided during the course.

4.1 Compatibility with various Web browsers, with disabled users, and with slow connections

1. The Internet store must deliver well-formed XHTML or HTML text, in conformance with W3C recommendations.

2. The project must not make use of client-side scripts, except under the following special circumstancies:

   (a) the script is used to implement a functionality that cannot be easily implemented in a different way (all the functionalities required by this document are easy to implement without client-side scripts; therefore, scripts can only be used by those students who want to push their projects beyond what is expressed in this document); and

   (b) if the web browser refuses to run the script, this will only result in a limited degradation of the operation of the Internet store; for example, in the case of a script that checks the values entered by the user into a form, if the web browser refuses to execute the script, the user must still be able to fill in the form and send it to the server, although the values filled in may not be checked.

3. The Internet store must be accessible through various Web browsers, including the W3C reference browser Amaya and the text-only browser Lynx (Lynx is often used by handicapped persons). (If requirements 1 and 2 above are satisfied, then this requirement will automatically be satisfied.)

4. The store must be usable by those who cannot display figures.

   For example, whenever an icon is used to perform an action, the action must be described in an accompanying **ALT** attribute. If there is an icon for the ***add to cart*** link, then the corresponding **IMG** tag must contain an attribute like this:

```
ALT="add to cart"
```

4.2 Reliability

The serializable isolation level must always be used in PostgreSQL.

4.3 Code clarity and modularity

1. Every Java class must have a clear, easy to understand purpose and semantics. The same is true for every non-`private` and non-`protected` method in a class.
2. Every field or method in a class that is not intended for use by other classes must be marked `private` or `protected`.
3. Every Java name and every SQL table or column name must clearly say what it means, in English. For example, a boolean variable that says whether the shopping cart is empty should be named `cartIsEmpty`, and not `cartState` (this name is imprecise) or `cart` (this name is misleading) or `pustyKoszyk` (this name is not in English, and additionally it is misleading).
4. Every Java name must start with a capital if and only if it represents a class.
5. Actions (as defined by Jakarta Struts) and methods called by actions must only be used to generate results and to perform database operations, and not to describe the layout of webpages or to display anything.
6. JSP must only be used to display results previously computed by actions.
7. Graphical properties (fonts, sizes, colors etc.) of everything we display must be described in cascading stylesheets (CSS).
8. In the database schema, the **not null**, **unique** and **references** properties of columns must be used where appropriate.

4.4 Performance

All the operations of the Internet store must perform decently. Specifically:

1. The time taken by an operation must remain approximately constant regardless of the size of database tables in our store (except for plain text search operations, which will take a time roughly proportional to the number of articles among which we must search using plain-text criteria).
2. All indexes needed by the queries that we make must exist in the database.