[ TEAM \* ]

**Price Finder and Ordering for Consumer Electronics**

**PFOCE   
  
Software Requirements Specification**

**By**

**\*\*\***

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 02/07/01 | 1.0 | Initial Version | Team2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Purpose 1

1.1. Definitions, Acronyms, Abbreviations 1

1.1.1. Definitions 1

Customer- The person that is browsing the web site to browse or purchase products from a vendor. 2

1.2.2 Acronyms 2

1.2. References 2

1.3. Overview 2

2. Overall Description 3

2.1. Product Perspective 3

2.2. Product Architecture 3

2.3. Product Functionality/Features 3

2.4. User Characteristics 3

2.5. Constraints 3

2.6. Assumptions and Dependencies 3

3. Specific Requirements 4

3.1. Functional Requirements 4

3.2. External Interface Requirements 4

3.3. Internal Interface Requirements 4

3.4. Internal Data Requirements 5

3.5. Design and Implementation Constraints 5

4. Non-Functional Requirements 6

4.1. Safety Requirements 6

4.2. Security and Privacy Requirements 6

4.3. Environmental Requirements 6

4.4. Computer Resource Requirements 7

4.4.1. Computer Hardware Requirements 7

4.4.2. Computer Hardware Resource Utilization Requirements 8

4.4.3. Computer Software Requirements 9

4.4.4. Computer Communication Requirements 10

4.5. Software Quality Factors 11

4.6. Packaging Requirements 11

4.7. Precedence and Criticality of Requirements 11

Computer Hardware Resource Utilization Requirements 12

5. Qualification Provisions 13

From Shopper 14

From Vendor 15

From System 16

From Admin 17

6. Requirements Traceability 18

6.1. Upward Traceability and Downward Traceability 18

# Purpose

This document will describe the project “On-Line Store” (referred to as the project). Furthermore, this document will provide an overall description of specific requirements, a description of non-functional requirements, qualification provisions and an overall description.

The intent of this document is to provide a thorough understanding of how the project works from a design standpoint. It will also describe how the software will be used with the use cases. Lastly, it will show how the software will be tested with the test cases. The intended audience of this document is as follows:

The primary stakeholders- all people that are involved in outsourcing.

The secondary stakeholders – Main contractors

**Scope**

This document will show the design, use cases, test cases, functional, and non-functional requirements of the GUI (Price finder component) component for the PFOCE.

## Definitions, Acronyms, Abbreviations

The following is a list of commonly used definitions used throughout this document:

### Definitions

**Web –** the Internet.

**Java Virtual Machine-** the run time environment for java.

**Query –** the act of requesting information from a database.

**Servlet –** an individual process spawned by the Java virtual machine. It would best be described as a thread of execution that executes in a web environment.

**RMI –** A three tier client server architecture that adheres to the thin client model. Methods are accessed by a stub on the client side, invoked by the skeleton on the server side and executed on the server side.

**Vendor-** The entity that provides products to sell on the web site.

### Customer- The person that is browsing the web site to browse or purchase products from a vendor.

### Acronyms

**VM -** The java virtual machine.

**RMI –** Remote Method Invocation

**CD-** Compact disk

**DVD** – Digital Versatile Disk (used to be Digital Video Disk)

**DB –** database.

**JDBC –** Java database client.

**SQL -** Structured Query Language the standard language for accessing relational databases such as Oracle, DB2, Informix, Sybase, and Teradata .

## References

Supporting documentation includes the risk management and user guide documents.

## Overview

On-Line Store is a web-based storefront that has the following functionality. It will provide an interface for the customer to make queries by category and description of products supplied by vendors. The application will query the databases that are located both in house and at vendor sites to retrieve the information that the customer requests. If a customer decides to make a purchase the system will provide a secure transaction. The system will also provide the means of conducting transactions with the vendors. It will bill the vendor, pay the vendor, and allow the vendor to update their data if it resides in house.

# Overall Description

## Product Perspective

“PFOCE” is an E-commerce portal for consumer electronics. It provides robust, highly scalable price search engine, and on-line transaction capability in a heterogeneous environment. It searches among different vendors’ database, and return the price with user specific format. User could do secure online transactions based on the latest pricing comparison result.

## Product Architecture

Because of the salability and maintainability are the top priority in the E-commerce environment nowadays, “PFOCE” is based on the three-tier architecture which separate data, business logic, and presentation to provide zero client management, and component re-use capability. “PFOCE” runs JSP, Java Servlets, and Java Beans on Unix server and use JDBC to communicate with mySQL or some other relational database.

## Product Functionality/Features

1. Let user browse the price with the view they want

2. Compare the price of specific product for the user

3. Secure on-line transaction using Apache SSL

## User Characteristics

We plan to implement the 3-tier architecture with JSP, Servlet, and mySQL database

## Constraints

The accuracy of pricing data may affect by update frequency.

## Assumptions and Dependencies

# Specific Requirements

## Functional Requirements

* Needs to be web-based distributed system.
* The system should allow the user to query the vendors through specified parameters such as price, brand, model, year etc....
* The system will inform the users, depending on the search, of the number of items available in a particular store.
* The system should allow the user to send an order as well as cancel an order.
* The shopping cart should allow the user to add and remove items from the shopping cart.
* The system should allow customers to contact vendors or the system administrator.

## External Interface Requirements

* There will be two interfaces to the project.
* The external interface should be user friendly for people to use
* The external interface should have familiar ways to browse like: back, forward, main, etc.
* The first interface will be an Internet browser.
* System shall have Java enabled Internet browser: Netscape Navigator 4.0 or greater, Internet Explorer 4.0.

It should be noted that in order to use these browsers the java functionality must be enabled in the browser settings.

* The second interface will be a java swing interface that allows for business use of the software behind the firewall.

## Internal Interface Requirements

1. System shall contain mapping information to record how data from web data servers is accessed.
2. System shall have a server application that performs all analysis, database retrieval, and Real time query.
3. System shall be capable of being accessed by multiple users.
4. The system should inform the system administrator is something goes wrong
5. System shall be available 24 hours per day, 7 days per week.
6. System shall be able to support up to 1000 transaction per hour with average response time 5sec to maximum response time 10sec.
7. The system integrator or administrators should have the flexibility to deploy the applications in many different ways.
8. The system shall have the encryption for security requirement
9. Secure Socket Servers will provide users with an encrypted form for security.

## Internal Data Requirements

There exists one main database in the PFOCE server for the use of the application. The database can be modified from the outside by a vendor. The database will consist of one large database table with a vendorId for an attribute. When the vendors solicit an update on their products, this system will retrieve the latest information from their database. This is done for purpose of security.

## Design and Implementation Constraints

The design and implementation of this system will require object-oriented design as well as an object-oriented language. The design is limit to a heterogeneous architectural style consisting of client-server style and object-oriented style. The implementation language use should be able to be run in different platforms, run Servlets, java beans, java applets, and java JSP pages. The language use will be java since this language supports all of the above features. In addition, the design and implementation need both to be consistent with each other. The minimum deviation of implementation from the design could affect the performance and security of the system

Since this is an Internet application it carries all the constraints of being on a large-scale network. Moreover, the tools use to create the designs involve Rational-rose and choosing java as the language for the design.

# Non-Functional Requirements

## Safety Requirements

Safety requirements of a software package include the freedom of the users and stakeholders from physical harm and financial loss. For this application the only potential risk is the potential financial loss. The safety attributes the system must provide in order to be safe are as follows:

* + Ability of the system to securely store Employee data.
  + Ability of the system to securely store Employee’s documents.
  + Provide secure transactions.

The aforementioned attributes must be met to make the system safe. Section 4.2 will elaborate the security aspects.

## Security and Privacy Requirements

We will house employee information and documents that must be kept private.

## Environmental Requirements

In order for the application that we build to operate properly, it must be housed in a cool location with optimal air circulation and refrigeration.

## Computer Resource Requirements

### Computer Hardware Requirements

1. Client
   1. Some way to access the Internet though a specified browser (Firefox 3.x or any Webkit browser, such as Safari 4.x or Google Chrome will be supported)
   2. A valid email address.
2. Web Server
   1. Network Adaptor and connection to network
   2. Largish hard-drive
3. Firewall (Optional but strongly recommended)
   1. This could be a machine running firewall software or a dedicated firewall (such as a Cisco PIX). We will use a Linux system running iptables
4. SQL Server (Small networks may use the same machine)
   1. Network Adaptor and connection to network
   2. Very large hard drive, preferably with redundancy. The more users we have the larger the storage is necessary.

### Computer Hardware Resource Utilization Requirements

1. Client
   1. We cannot possibly consider the hardware resource utilization for the clients.
2. Web Server
   1. Through the use of local directors we can make it appear as though many web servers (with a large load) are one web server all with the same content.
   2. Scalability is very high with this implementation.
3. Firewall
   1. This piece of hardware should be capable of handling maximum number of connections. Although this is a critical piece of hardware the load on it should be relatively light because of the nature of what this piece of hardware does.
4. SQL Server
   1. If the load becomes to high for only one database (extremely unlikely) we have the option of upgrading to a faster server with more ram or scaling across multiple databases.

### Computer Software Requirements

1. Client
   1. Clients must be running a compatible web browser (ie Firefox 3.x or any Webkit based browser, such as Safari 4.x or Google Chrome will be supported)
2. Web Server
   1. nginx web server as a load balancer using mongrel as the Ruby on Rails hosts
   2. SSL software to facilitate encrypted data transfer for security reasons.
   3. Some Unix OS (max scalability)
   4. Database Connectivity (PostgreSQL)
3. Firewall
   1. OS does not really make a difference as long as it does an efficient job of isolating the network.
4. DB Server
   1. Unix OS
   2. PostgreSQL

### Computer Communication Requirements

1. Client
   1. Clients need to be able to access the Internet with their computers. (This means TCP/IP connection to the internet)
2. Web Server
   1. Must be capable of serving pages to requests fielded on port 80 (default http port)
   2. Must be capable of transferring secure requests to a secure server behind the firewall.
   3. Must be capable of reaching and querying the database.
3. Firewall
   1. Must be able to keep unwanted traffic out (ACL) and provide additional security to items behind the firewall.
4. SQL Server
   1. Must be able to serve with web servers
   2. Must be able to serve secure web servers

## Software Quality Factors

The quality factors of most concern to this project are as follows:

Time to Market

Cost to Develop

Reliability

Maintainability

Scalability

All of these factors contribute to the business attributes of the final product. Due to the nature of the Internet, time to market is important so that the product has not been preceded by another product doing the same thing. Cost to develop is a concern of any software product. If it costs to much to develop then the time to recuperate the cost will not make good business sense. Maintainability of the software implies that if the company wants to implement a change or add a feature it should be relatively simple. Scalability, implies that as the traffic of the software increases the entire system should be able to be increased to fit the traffic.

## Packaging Requirements

The system will be provided through a network install. Most of the software required is open source and the actual Document Management System will be installed from a source code repository.

## Precedence and Criticality of Requirements

The following will try to interpret the importance of each of the aforementioned requirements. Each of the requirements will be listed in their order of importance. The number assigned to the requirement will correspond to the section number of this document. For example, the reference to the number 4.1 will be referring to the section 4.1 Safety Requirements.

The Criticality of the requirement is rated on a scale of one to four. A four means it is of the utmost importance. A level of one means that this a nice feature to have. Two and three are nearly self-explanatory. If it is a little more important than a feature then it would be a two, etc…

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Criticality** | **Precedence** |
| Security and Privacy Requirements | 4 | 1 |
| Functional Requirements | 4 | 1 |
| Computer Communication Requirements | 4 | 1 |
| Safety Requirements | 4 | 1 |
|  |  |  |
| Computer Software Requirements | 3 | 2 |
| Computer Hardware Resource Utilization Requirements | 3 | 2 |
| Environmental Requirements | 3 | 2 |
| Computer Hardware Requirements | 3 | 2 |
| Internal Data Requirements | 3 | 2 |
| Internal Interface Requirements | 3 | 2 |
|  |  |  |
| Software Quality Factors | 2 | 3 |
| External Interface Requirements | 2 | 3 |
|  |  |  |
| Packaging Requirements | 1 | 4 |

# Qualification Provisions

***TO BE DETERMINED***

This section defines a set of qualification methods. For each requirement in Section 3, specify the methods to be used to ensure that the requirement has been met. A table may be used to present this information, or each requirement may be annotated with the method(s) to be used. Qualification methods may include: Demonstration, Test, Analysis, Inspection, Special.

This section of the SRS deals with how we qualify the proper execution of the use cases. The qualification methods to guarantee the requirements have been achieved are:

Analysis (separation of a whole into its component parts)

Testing (Systems Testing, User Acceptance Testing)

Customer Beta Testing Prototype (Demo Version Release)

Demonstrations

Inspections (checking or testing against established understandings)

Prototype

|  |  |  |
| --- | --- | --- |
| From Shopper | **Qualification Method** |  |
| Search | Analysis, Testing, Demonstrations | As much as possible, analyze. Break down the process and review the end result of your queries. Show the users the program’s search functionality. |
| Shop | Demonstrations, Inspections | Show the users the functionality of the program. |
| Shopping Cart | Demonstrations, Inspections | Show the users the functionality of the program. |
| Secure Server | Analysis, Inspections | Break down the process and analyze. Systems Test, then analyze the results. |
| Browse | Demonstrations, Customer Beta Testing Prototype | Could include Inspections, as we click different links, we can make sure of the expected. |
|  |  |  |
| From Vendor | **Qualification Method** |  |
| Login | Analysis, Testing | Break down the process and analyze. Systems Test, then analyze the results. |
| Change Information | Analysis, Testing | Break down the process and analyze. Systems Test, then analyze the results. |
| Automatically Update Information | Analysis, Testing, Inspections | Break down the process and analyze. Systems Test, then analyze the results. |
| Manual Product Information Change | Analysis, Testing, Inspections | Break down the process and analyze. Systems Test, then analyze the results. |
| Search Inventory | Analysis, Testing | Break down the process and analyze. |
| Update Inventory | Analysis, Testing | Break down the process and analyze. |
| From System | **Qualification Method** |  |
| Charge Vendor | Analysis, Testing, Inspections | Break down the process and analyze. Systems Test, then analyze the results. Generate accounting reports. |
| Update Vendor Information | Analysis, Testing, Inspections | Break down the process and analyze. Systems Test, then analyze the results. Set up test methods to ensure consistent information. |
| Update Shopper Information | Analysis, Testing, Inspections | Break down the process and analyze. Systems Test, then analyze the results. Set up test methods to ensure consistent information. |
| Process Credit Card | Analysis, Testing, Inspections, Demonstrations | Break down the process and analyze. Systems Test, then analyze the results. |
|  |  |  |
| From Admin | **Qualification Method** |  |
| Manage Vendor | Analysis, Testing, Inspections | Break down the process and analyze. Systems Test, then analyze the results. Detailed test methods will be in place to ensure information integrity. |
| Manage Shopper | Analysis, Testing, Inspections | Break down the process and analyze. Systems Test, then analyze the results. Detailed test methods will be in place to ensure information integrity. |
| Manage Database | Analysis, Testing, Inspections | Break down the process and analyze. Systems Test, then analyze the results. Detailed test methods will be in place to ensure all data integrity. |
|  |  |  |

# Requirements Traceability

## *TO BE DETERMINED* Upward Traceability and Downward Traceability

Upward:

Traceability from each requirement in this specification to the system or subsystem requirements it addresses.

Downward:

Traceability from each system or subsystem requirement to the requirements contained in this specification

The upward and downward traceability will be accomplished by utilizing a tracibily tree. The following tree illustrates the traceability of the major components used on this system:

Client

Server

System Administrator

Ordering a product

Paying for a product

Searching for a product

Selecting a product

Browsing on website

Updating products

Adding database data

Removing database data